BOOST: A BOOtSTrapping Protocol for 802.11-based Self-Organizing Hierarchical Wireless Ad-Hoc Networks

L. Raju, S. Ganu, B. Anepu, I. Seskar and D. Raychaudhuri
WINLAB, Rutgers University
Piscataway, New Jersey 08854
{lalit,sachin,bhaskar,seskar,ray}@winlab.rutgers.edu

***Abstract*— An important aspect of the deployment of wireless LANs lies in careful planning of the locations and the channel assignments of the Access Points (APs). The APs need to be placed optimally to provide maximum coverage and minimize interference with other APs. Also, in the case of ad-hoc networks that require no fixed network infrastructure, there is a need to allocate channels carefully so as to minimize interference and improve the system performance. In this paper, we propose a distributed bootstrapping mechanism (BOOST) for ad-hoc networks that allocates channels so as to improve system performance in terms of throughput and average delay. We present experimental results that motivate the design of BOOST protocol which is validated through ns-2 simulations. We propose two different approaches for bootstrapping using received power and number of beacons. Simulation results for both these approaches show a significant increase in the system performance compared to a random channel assignment.**

Key words: Ad-hoc network, Self-organization, Experimental work, Hierarchical networks

I. INTRODUCTION AND PRIOR WORK

In current WLAN deployments for corporate as well as public environments, placement of APs and assignment of frequency channels both play key roles in determining the performance of the network. The importance of channel allocation in wireless system design has been discussed in [1][2] and the problem of channel allocation and AP placement has been formulated and solved using linear programming techniques in [2]. The latter proposes a solution to the problem of AP placements and channel assignment to maximize coverage and minimize interference. Both the above approaches have been proposed for deployments using the IEEE 802.11b infrastructure mode [3], where the terminals use APs for external connectivity as well as to communicate amongst themselves. However, our approach addresses the issue of channel selection in a distributed manner for a given topology of nodes operating in ad-hoc mode.

It is observed that in addition to infrastructure mode wireless LANs, ad-hoc networks based on 802.11 radios have also been gaining popularity as they do not need any fixed network infrastructure and can be rapidly deployed and configured with minimal effort. In [4][5], the authors show that the performance and scalability of such *flat* ad-hoc networks can be further improved by introducing hierarchy. In [6], we describe the design aspects and prototype implementation of a three-tier hierarchical ad-hoc network that 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 tier, and wired access points (AP) at the third and highest tier as shown in Figure 1. We also describe the protocols used for node initialization, neighbor discovery and multi-hop routing.

The basic mechanism for self-organization follows four steps after the devices are powered on: Bootstrapping, Discovery, Routing and Data Transmission. The discovery mechanism and the performance of the routing protocols depend upon the initial configuration of the nodes with respect to channel assignment. Poor channel selection may result in increased co-channel interference at the FNs. In order to mitigate this effect, each FN needs to use an appropriate channel on its beaconing interface (interface towards the MNs). This should be done in a distributed manner to accommodate the fact that the FNs may not be powered on simultaneously and some of them may join the network at a later stage.

In this paper, we focus on an efficient bootstrapping mechanism (BOOST) for the above network. This mechanism is based on active scanning by the FN to collect beacons from the terminals in its vicinity and then using this obtained information to allocate a channel for its interface towards the MNs. We show that by proper selection of the channel, a significant improvement in the throughput can be achieved as compared to the case where the channels are selected randomly. We also show that the received power is a useful metric for channel allocation.

In section II, we describe the system architecture of the hierarchical ad-hoc network in order to better understand the bootstrapping mechanism that applies to the system. In section III, we describe the experimental observations that motivate the design of BOOST protocol and the protocol itself. Section IV discusses the simulation results to compare the performance of the different proposed algorithms for channel allocation. Section V concludes the paper.

[1]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, 2003

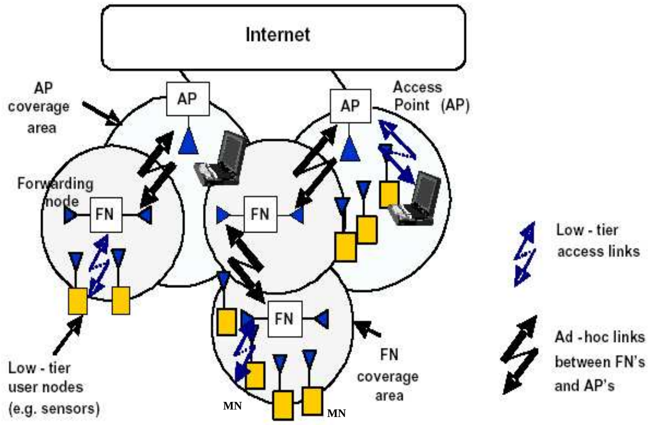

Fig. 1. Three tier hierarchical ad-hoc networks

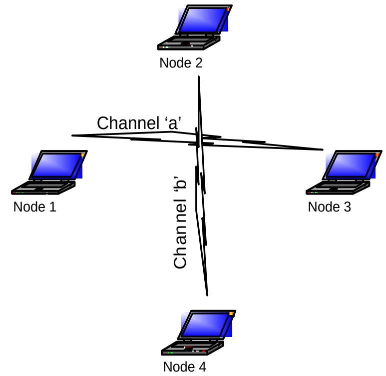

Fig. 2. Experimental setup

II. System Architecture

The self-organizing hierarchical ad-hoc network mentioned in the previous section is three-tiered and consists of the following nodes.

- *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 nodes at the higher tiers of the network in order to obtain service using a discovery protocol. The MN uses a single 802.11b radio operating in ad-hoc mode to communicate with the point(s) of attachment
- *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. An FN has two radio cards, one for traffic from MN to FN and another for inter FN and FN-AP traffic flows (typically carried on a different frequency)
- *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, unlike typical 802.11 WLAN deployments, operates in ad-hoc mode

III. BOOST Protocol for the Forwarding Nodes

As described in the previous section, the first important step in the self-organization process is node initialization. The APs can be configured to be on non-overlapping channels whereas the MNs are initalized to start scanning on the wireless interface in order to discover their neighbors and associate with them. The FNs however have two radio interfaces that have a different functionality. On one interface (known as the *scanning interface*), the FNs are initialized to scan channels in order to discover neighbors and associate with one (or more) neighbor(s). The second interface (known as the *beaconing interface*) is typically operated on a separate channel on which the FNs send beacons for the MNs to associate with them. The bootstrapping protocol helps the FN assign a proper channel for the second interface in order to minimize interference with the neighbors and maximize system throughput.

A. Experimental Observations

The bootstrapping protocol is based on the experimental results conducted in order to assess the effect of channel separation on the throughput of two simultaneous traffic flows. Figure 2 shows the experimental setup with four wireless devices, all in one-hop reach from the other. Four laptops operating on Linux with PCMCIA-based wireless LAN cards were used as wireless devices and configured to operate in ad-hoc mode. We used *netperf*[7], an open source network performance measurement utility to measure throughput over a UDP transport mechanism with 1472 byte packets.

Throughput of each pair (1-3 and 2-4) was measured individually, for reference. Subsequent experiments were conducted with both flows ON simultaneously. Channel 'a' was fixed at '1' while channel 'b' was varied from '1' to '11'. During each run, the pairs of nodes were made to communicate and their corresponding throughputs were recorded. The experimental results are shown in Figure 3.

The maximum throughput that could be obtained (when only one flow was active) was approximately 6 Mbps. We observed that when all the four nodes were on the same channel, physical carrier sensing and IEEE 802.11 MAC ensured that both the pairs contend fairly for access to the channel and hence the throughput was almost the same (approximately 3.5Mbps). In IEEE 802.11b [3], the adjacent channels are overlapping and there are a maximum of three non-overlapping channels. We see that when the channel separation between the two flows was 1 or 2, both the flows suffered considerably in terms of performance. This can be attributed to the fact that since the two flows were not on the same channel but on adjacent overlapping channels, there was an higher possibility of failure in carrier sensing resulting in more collisions and hence lower throughput. As the channel separation was increased to beyond 4, the flows experienced very little interference from each other and hence the throughput improved. Beyond a channel separation of 5, the channels were orthogonal

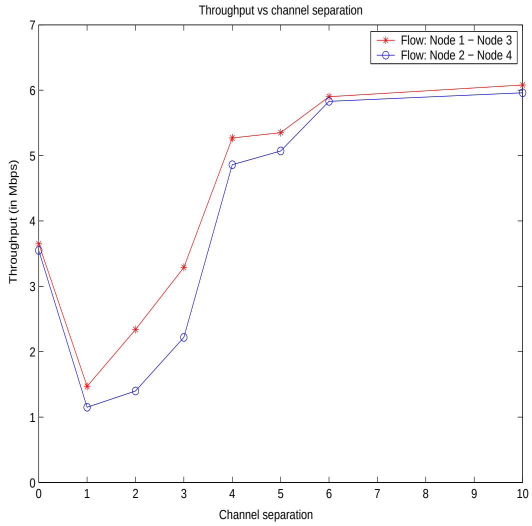

Fig. 3. Throughput with increasing channel separation

and hence the throughput saturated to its best case value (approximately 6 Mbps).

These results indicate that whenever the packets from the second flow are received at a power which is close to the receiver sensitivity threshold, there is a higher chance of failure in detecting a packet. This causes an increase in the number of collisions and hence a reduced throughput per flow. Whenever the channels are far apart so that the received power from the second flow is well below the receiver sensitivity threshold, the second flow does not cause interference and hence will not affect the throughput.

B. BOOST Protocol

The bootstrapping algorithm (BOOST-A) at the forwarding nodes involves the following steps:

- Each FN and AP sends out beacons periodically; we simply use the existing 802.11 beacons
- When an FN is powered on, the scanning interface sweeps through the channels in order to discover its neighbors
- During this sweep, it receives beacons from APs or other FNs in its neighborhood and records the received power and the number of terminals heard on each channel. If there is a channel on which no beacons are detected, then it is chosen over those on which beacons were received
- Out of the received power per packet on each channel, the FN then records the minimum value of the received power. By doing this, we are taking into account all the possible neighbors whose packets are received at power levels close to the receiver sensitivity threshold and hence might potentially interfere and reduce throughput by causing collisions
- Of these minimum values, we choose the highest across all the channels. This gives us the best possible channel to use assuming that there is no channel on which no beacons were detected
- Once the channels are assigned weights based on the above procedure, the beaconing interface of the FN is set to that particular channel and it starts sending beacons on that channel

In an alternative approach (BOOST-B), each FN records the number of beacons received on each channel during the process of scanning and chooses the channel on which minimum number of beacons were received. In case of a tie for the minimum number of beacons, a channel is selected randomly. The advantage of BOOST-B is its lower complexity. Also, while the received power varies from packet to packet, the number of beacons received on each channel is less likely to change as often.

IV. Simulation Results and Discussion

A. Simulation Environment and Performance Metrics

Our algorithms were simulated using the Monarch extensions to the ns-2 simulator [9][10]. The current capabilities of the simulator do not provide overlapping IEEE 802.11 channels and routing support for multiple interfaces. Therefore, flows even on two adjacent channels would experience no interference from each other. We have modified the ns-2 simulator to both reflect the overlap in channels and support the hierarchical architecture with multiple interfaces. In particular, we used modified AODV to simulate our algorithms. We used experimental results from [8] to enhance the ns-2 model to accomodate channel overlap.

Our simulations were conducted for a topology comprising 1 AP, 6 FNs and 10 SNs. The flow of traffic was from the SNs to the AP using intermediate FNs. The SNs scan all 11 channels to identify a suitable FN to forward their traffic. Table 1 shows the simulation parameters used to validate our algorithms. The efficiency of the boot-strapping algorithm was compared in terms of the system throughput and average delay.

TABLE I
Simulation Parameters

Simulation area	400m x 400m
Number of nodes: APs:FNs:MNs	1:6:10
Radio PHY	802.11 PHY (with channel overlap)
Radio PHY Rate: Radio Range	1Mbps:250m
MAC protocol	802.11 CSMA (adapted for hierarchy)
Traffi c (Packets/sec)	10,20,30,40,50
Number of communicating pairs	10
Packet size	512 bytes
Traffi c model	CBR

We simulated three algorithms for the bootstrapping at the FNs. BOOST-A and BOOST-B were implemented as discussed earlier. In order to determine the significance of our algorithms, we also simulated a network in which each FN would randomly choose a channel to send its beacons. For the random channel selection, we averaged the results over several different random channel allocations. We then compared the performance of all three algorithms in terms of throughput and delays experienced by the different flows.

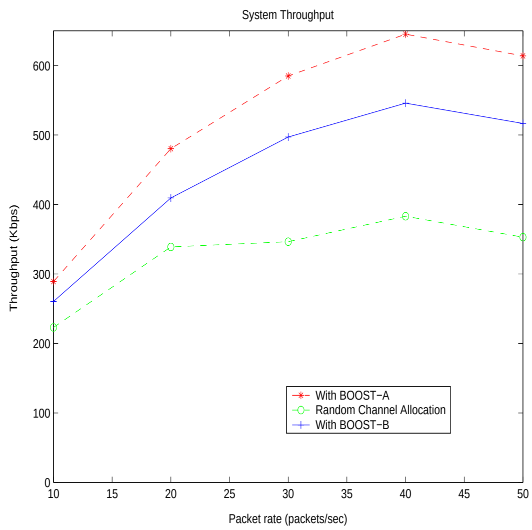

Fig. 4. System Throughput for different packet rates

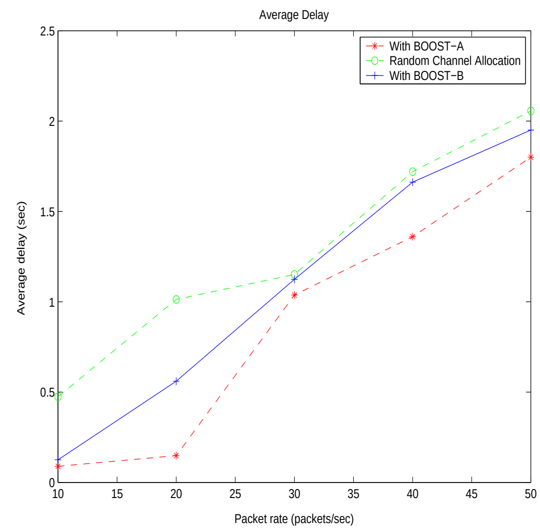

Fig. 5. Average delay for different packet rates

- System Throughput - measured as the number of bits per second delivered at the AP from each SN, averaged over the duration of the entire simulation
- Average End-to-end Delay - captures all delays including route discovery, queuing delays, propagation delays, retransmission delays at the MAC and transfer time

B. Results and Discussion

We plot the throughput and delay results from a series of simulation runs for the different algorithms for increasing offered load (packets/second). The results in Figures 4 and 5 indicate that channel selection has a significant impact on the performance of the network, both in terms of throughput and average delay. Both BOOST-A and BOOST-B outperform the random channel allocation. The reason for the poor performance of the random allocation may be attributed to the co-channel interference phenomenon that we observed in our experiments. In BOOST-B, this situation is improved as we reduce the number of interfering nodes and priority is given to channels on which no other node exists. This results in a performance gain of approximately 34%. However, this still does not ensure that co-channel interference is minimized. As the number of beaconing nodes (1 AP and 6 FNs) is more than the number of orthogonal channels (3), there is interference which BOOST-B does not consider. This is improved in BOOST-A which provides approximately 56% higher throughput than the random case. BOOST-A minimizes the likelihood of co-channel interference by avoiding channels in which reception is close to the receive threshold. The average end-to-end delays are reduced by using the BOOST algorithms. These results validate our experimental observations.

V. Conclusions and Future Work

We have presented BOOST, a bootstrapping mechanism for hierarchical self-organizing wireless ad-hoc network. This distributed mechanism can be used to automate the channel allocation on the forwarding nodes whenever they are powered on. We have used experimental observations to design the BOOST protocol and validated the performance of the algorithms using ns-2 simulations. Future work planned includes implementation of this mechanism on the experimental testbed and to verify the results using experimentation. Also, it would be of practical importance to augment BOOST protocol by including initial transmit power control along with channel allocation.

References

[1] A. Hills, "Large-scale wireless LAN design", *IEEE Communications Magazine*, vol 39, issue 11, Nov 2001, pp. 98-107.

[2] Youngseok Lee, Kyoungae Kim and Yanghee Choi, "Optimization of AP placement and channel assignment in wireless LANs", *Proceedings of 27th Annual IEEE Conference on Local Computer Networks*, November 2002, pp 831-836.

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

[4] S. Zhao, K. Tepe, I. Seskar, D. Raychaudhuri, "Routing Protocols for Self-Organizing Hierarchical Ad-hoc Wireless Networks", *Proceedings of the IEEE Sarnoff Symposium*, Trenton, NJ, March 2003.

[5] S.Zhao, I.Seskar, D. Raychaudhuri, "Performance and Scalability of Self-Organizing Hierarchical Ad Hoc Wireless Networks", *to appear in the IEEE Wireless Communications and Networking Conference (WCNC 2004)*, March 2004, Atlanta, GA.

[6] S. Ganu, L. Raju, B. Anepu, I. Seskar, D. Raychaudhuri, "Architecture and Prototyping of an 802.11-based Self-Organizing Hierarchical Ad-Hoc Wireless Network (SOHAN)", *submitted to the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC 2004)*, Tokyo, Japan, March 2004.

[7] Netperf Home Page, *http://www.netperf.org/netperf/NetperfPage.html*.

[8] Cirond Four Point Channel Allocation Technology, *http://www.cirond.com/site/products/whitepapers.htm*.

[9] Kevin Fall and Kannan Varadhan, *ns Notes and Documentation*, Technical Report, UC Berkeley, LBL , USC/ISI, and Xerox PARC, November 1997.

[10] CMU Monarch Group. CMU Monarch extensions to ns. *http://www.monarch.cs.cmu.edu/*.