

# Evaluation of A Wireless Enterprise Backbone Network Architecture

Ashish Raniwala Tzi-cker Chiueh  
Department of Computer Science  
Stony Brook University  
Stony Brook, NY-11794  
Email: {raniwala,chiueh}@cs.sunysb.edu

**Abstract**—IEEE 802.11 wireless LAN technology is mainly used as an access network within corporate enterprises. All the WLAN access points are eventually connected to a wired backbone to reach the Internet or enterprise computing resources. In this project, we aim to expand WLAN into an enterprise-scale backbone network technology by developing a multi-channel wireless mesh network architecture called *Hyacinth*. *Hyacinth* equips each node with multiple IEEE 802.11a/b NICs and supports distributed channel assignment/routing to increase the overall network throughput. In this paper, we present the results of a detailed performance evaluation study on the multi-channel mesh networking aspect of *Hyacinth*, based on both NS-2 simulations and empirical measurements collected from a 9-node *Hyacinth* prototype testbed. A key result of this study is that equipping each node of a *Hyacinth* network with just 3 NICs can increase the total network bandwidth by a factor of 6 to 7, as compared with single-channel wireless mesh network architecture.

## I. INTRODUCTION

Today the main application of IEEE 802.11 wireless LAN (WLAN) technology in the enterprises is to provide last-mile connectivity between end users' desktop/laptop machines and the enterprise backbone network, which is predominantly based on Ethernet. In the foreseeable future, there will always be a gap in physical link speed between wired and wireless networking technology, and therefore it is unlikely that WLAN will be able to compete with Ethernet for very-high-speed campus backbones in large enterprises. However, as WLAN's signal coverage and radio link speed improve, it is conceivable that WLAN deployment will inch toward some enterprise's core backbone, eventually turning the enterprise into a truly wireless computing environment. We envision two possible application scenarios for this wireless enterprise backbone network technology. First, for small to medium enterprises or public hot-spots, the bandwidth requirement is relatively modest, and the simplicity and agility of WLAN technology thus may significantly out-weigh the concerns on performance and security. In other words, when WLAN technology is "good enough," the link speed gap between wired and wireless networks may not matter for certain types of users. Second, even for large enterprises, where wired backbones are unlikely to be replaced, a wireless backbone can provide a backup to the main wired backbone for improved availability and throughput. The rapid deployability and low cost advantages of WLAN technology greatly reinforce this possibility.

Motivated by these applications, we have developed a multi-channel wireless mesh network architecture called *Hyacinth*, which is specifically designed to serve as a wireless enterprise backbone, and features several unique innovations:

- *Multi-Channel Mesh Networking*: Most of today's 802.11-based WLANs working in the ad hoc mode tune all their nodes in to the same channel. Using the same channel is necessary to keep the network connected as each wireless node is equipped with only one interface. This architectural choice greatly reduces the bandwidth available to each path because of interference between transmissions from adjacent nodes on the same path as well as neighboring paths [10]. In contrast, *Hyacinth* supports multiple WLAN interfaces per node, each operating at a different radio channel. Using a load-aware channel assignment, *Hyacinth* is able to make much more efficient utilization of the radio spectrum defined in the 802.11 standard and support much higher aggregate throughput than conventional ad hoc network architectures.
- *Device-Transparent Horizontal Handoff*: As *Hyacinth* is meant to cover an entire enterprise, network applications running on a user's mobile device should remain undisturbed as the user roams across the coverage areas of different wireless mesh network nodes. Not only can *Hyacinth* support transparent horizontal handoff to minimize application-level disruption, it can also provide this functionality *without modifying software/hardware on the mobile devices*.
- *Seamless Vertical Handoff*: To be an effective backbone technology, the coverage of *Hyacinth* has to be as comprehensive as possible. However, there are bound to be dead zones in a wireless network due to either deployment lapses or transient radio interferences. *Hyacinth* supports seamless vertical handoff between GPRS/3G and wireless LAN in such a way that when a host's WLAN link goes down, it automatically switches to the GPRS/3G link, and its network applications continue to run without interruptions during the transition.
- *Crosscheck-Based Routing Security*: We recognize that the major concern of enterprise users about WLAN technology is its security. In the case of *Hyacinth*, the security requirement is even more stringent, because if any individual *Hyacinth* node is compromised, the entire network can become unavailable. *Hyacinth* features a distributed cooperative route packet crosschecking mechanism that can quickly detect the inconsistency among route packets, and trace the roots of these inconsistencies to specific nodes. This mechanism significantly enhances the availability of a *Hyacinth* network when a wireless mesh network node is compromised, mis-configured, or broken.

We have designed and implemented an initial *Hyacinth* prototype that consists of a network testbed of nine nodes, each with two IEEE 802.11a WLAN interfaces and one IEEE 802.11b WLAN interface. In this paper, we will focus on the evaluation of the multi-channel mesh networking aspect of the *Hyacinth* architecture, using both simulation results and empirical measurements on the prototype. In Section 2, we briefly review previous research on using multiple radio channels simultaneously in wireless networks as well as load-balancing routing. Section 3 describes the key algorithms used in *Hyacinth* to exploit multiple radio channels in a wireless mesh network. Section 4 presents a detailed evaluation of the effectiveness of *Hyacinth*'s channel assignment and packet routing algorithms. Section 5 concludes the paper with a summary of main research results and an outline of on-going research.

## II. RELATED WORK

Several proposals [1] [2] have been made to *modify the MAC layer to support multi-channel networks*. In all these approaches, an optimal channel is found for every single packet transmission, so as to avoid interference and enable multiple parallel transmissions in a neighborhood. In contrast, our architecture does not switch channels on a packet-by-packet basis; our channel assignment lasts for a longer duration, such as several minutes or hours, and hence does not require re-synchronization of communicating network cards on a different channel for every packet. This property makes it feasible to implement our architecture using commodity 802.11 hardware.

The *multi-NIC approach* has also been discussed in some previous research [5] [6]; its true performance potential has however not been discovered earlier. Specifically in [5], authors use multiple 802.11 NICs per node in an ad hoc network setting, by assuming a simple a priori channel assignment to the NICs. For each node, NIC-1 is assigned channel-1, NIC-2 is assigned channel-2, and so on. This *identical channel assignment* technique can only yield a factor of 3 improvement using 3 NICs, as compared to a factor of 7 improvement possible with our channel assignment scheme. The key to this performance potential lies in the channel assignment technique that decides which channel to use for which NIC in the network and in turn how much bandwidth is made available to each NIC in the network. We bring out this fact, and present a centralized channel assignment/routing algorithm in [10]. In the current paper, we evaluate a distributed channel assignment/routing algorithm implemented in a real 9-node *Hyacinth* testbed. BelAir200 [12] is a commercial mesh networking product that uses *multiple radios in conjunction with directional antennas*. Their approach requires every node to use a separate radio to communicate with each one of its neighbors; this design increases the hardware requirement on each node. Transit Access Points [11] also proposes use of beamforming antennas on mesh nodes. Beamforming however requires modifications to 802.11 MAC [11].

A vast amount of research has been conducted in single-channel *multi-hop routing in ad hoc networks* [4]. In [6], authors propose an algorithm for load-balancing wireless links of a gateway node that connects a wireless access network

to the wired network. In their algorithm, the gateway node co-ordinates the movement of all the nodes across the tree to achieve load-balancing. In contrast, our load-balancing routing algorithm does not require such coordination or centralized computation. Various techniques have been proposed to measure an individual “link-load” in a wireless network – using the number of packets [8] or number of paths [7] going through a particular node and its interference zone. We similarly measure the traffic going through each node and compute the available bandwidth on each link of a path as well as on the gateway node to find a load-balancing route. As different links in a neighborhood can operate on different channels, we separate the load on each channel and use it to compute the residual bandwidth available to any link.

Lastly, Firetide Inc [14] and Strix Systems [13] feature use of mesh networks in the backbone of enterprise. Unlike *Hyacinth*, neither of them discusses use of multiple channels to solve the limited bandwidth issue of a single-channel mesh network.

## III. SYSTEM ARCHITECTURE

As shown in Figure 1, a *Hyacinth* network consists of fixed traffic aggregation nodes similar to wireless LAN access points. Each aggregation node provides connectivity to mobile wireless devices within its coverage area. The aggregation nodes in turn form a wireless mesh network among themselves to relay data to/from end-user devices and the wired network. Not all nodes need to have user-traffic aggregation capability. Some nodes could work as pure routers, while some other nodes, termed gateway nodes, connect the wireless mesh network to the enterprise wired network. All infrastructure resources such as file servers, Internet gateways (connecting to the ISP), and application servers, reside on the enterprise wired network, and can be accessed through any of the gateway nodes.

Each *Hyacinth* node is equipped with multiple 802.11-compliant NICs, each of which is tuned to a particular radio channel for a relatively long duration of time, such as several minutes or hours. For example, in Figure 1, each node is equipped with 2 NICs. The “virtual links” shown between nodes depict direct communication between them over the channel number associated with the link. Each node in this example network can only communicate over 2 frequency channels, but the network as a whole uses 5 different frequency channels. Note that a mobile node has only a single NIC, and the interaction between the mobile nodes and a traffic aggregation node is similar to the infrastructure mode operation in 802.11 standard.

The multi-channel mesh networking aspect of *Hyacinth* involves two fundamental design questions.

- 1) Which radio channel should be assigned to each network interface ? The goal of channel assignment in a multi-channel wireless mesh network is to bind each network interface to a radio channel in such a way that the available bandwidth on each virtual link is proportional to its load. As adjacent *Hyacinth* nodes connect via wireless links, they need to share a common channel between them. On the other hand, to effectively utilize multiple channels, each collision domain should be broken into

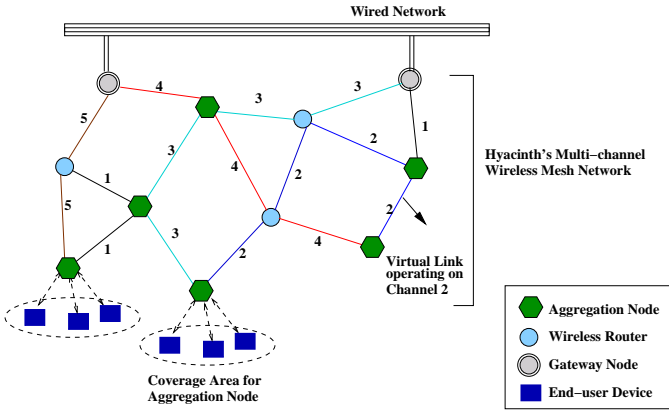


Fig. 1. System Architecture of Hyacinth's Multi-channel Wireless Backbone Mesh Network. End users' mobile devices connect to the network through access point-like traffic aggregation nodes, which form a multi-channel wireless mesh network among themselves to relay data traffic to/from end user devices. The links between nodes denote direct communication over the channel indicated by the number on the link.

as many channels as possible. A key constraint to the channel assignment problem is the small number of interfaces available on each node that limits the number of frequency channels the node can communicate over simultaneously.

- 2) How packets should be routed through this multi-interface mesh network? Each wireless mesh network node needs to establish a path to the wired network, in order to access the enterprise computing resources as well as the wired Internet. We provision for peer-to-peer traffic between aggregation nodes, but minimize the amount of interface and frequency resources dedicated to such traffic. Therefore, the goal of the routing algorithm is to determine a route between each traffic aggregation node and the wired network in such a way that balances the load on the mesh network, including the links to the wired network [6]. Load balancing helps avoid bottleneck links, and increases the network resource utilization efficiency.

The ultimate goal of the channel assignment and routing algorithms is to maximize the overall network goodput, or the number of bytes it can transport between the aggregation nodes and the gateway nodes within a unit time. To formalize this goal, we define the *cross-section goodput* of a network as

$$X = \sum_a \min\left(\sum_i C(a, g_i), B(a)\right) \quad (1)$$

Here,  $C(a, g_i)$  is the *useful* network bandwidth available between an aggregation node  $a$  and a gateway node  $g_i$ . If the bandwidth requirement between an aggregation node  $a$  and the wired network is  $B(a)$ , then only up to  $B(a)$  of the bandwidth between node  $a$  and all the gateway nodes is considered useful, and added to the cross-section throughput. The goal of the channel assignment and routing algorithms is to maximize  $X$ .

#### A. Centralized Channel Assignment/Routing

Even with a complete knowledge of network topology and traffic matrix, the channel assignment problem is *NP-hard*.

We prove this hardness by reducing *multiple subset sum problem* to the channel assignment problem in [10]. In the same reference, we also provide a greedy centralized channel assignment and routing algorithm. The algorithm works by first estimating the load exerted on each virtual link by each of the aggregated traffic flows, and thus the *total expected load* on each virtual link. The channel assignment algorithm then visits all the virtual links in decreasing order of their expected loads. Upon visiting a particular link, the algorithm greedily assigns it a least-loaded channel while maintaining the channels already assigned to the link's incident nodes as the constraint. The choice of the channel is based on the estimated usage of all the channels within the interference zone of the virtual link.

The overall algorithm goes through several iterations of channel assignment followed by routing. The channel assignment uses the expected link loads to decide the channels, and in turn the capacities of the virtual links. The routing algorithm takes these link capacities as input to come up with routes for different flows. These routes decide the expected link loads that are fed back to the channel assignment in the next iteration. The iterative process continues until the algorithm converges.

#### B. Distributed Load-balancing Routing

As most of the traffic is directed to/from the wired network, each node in the wireless mesh network needs to discover a path to reach one or multiple wired gateway nodes. When each node discovers a single path, the paths together form a forest of trees, where each wired gateway node is the root of one of the trees. These trees are therefore connected to each other through the wired network. In a more general case, each node can join multiple gateway nodes resulting in a meshed topology formation. While the latter approach can lead to better load balancing as well as higher resilience against node failures, it comes at the expense of additional wireless network interfaces to join multiple trees. In this paper, we focus on the former approach where each node actively associates with only one of the trees and uses the other paths only to recover from failures.

*Hyacinth's* basic routing mechanism is similar in operation to 802.1D Spanning-Tree-Protocol [3], although the route metric is modified to achieve load-balancing. In essence, a node that has already discovered a path to one of the wired gateways broadcasts this reachability information along with the cost associated with using that path. Other nodes that hear this advertisement can decide to join the former node. The resulting paths form a forest of trees, where each gateway node is the root of one of the trees.

The "cost" metric carried in the advertisements determines the final tree/forest structure. One possible cost metric is the *hop count* from a node to its corresponding gateway node, which allows a node to reach the wired network using the minimum number of hops, but does nothing to balance the network load. Another cost metric is the *gateway link capacity*, which indicates the residual capacity of the wired link that connects the root gateway of the node that sends out the advertisement to the wired network. The third cost metric is the *path capacity*, which represents the minimum residual

bandwidth of the path that connects the advertising node to the wired network. *Path capacity* is more general than *gateway link capacity* because the former assumes that the bottleneck of a path can be any constituent link on the path, rather than always the gateway link.

### C. Distributed Load-aware Channel Assignment

Along with discovering path to the wired network, a *Hyacinth* node needs to assign channels to its interfaces, and in turn the capacities of its virtual links. To enable distributed assignment of channels, each node divides its interfaces into three subsets - UP-NIC, DOWN-NIC, and P2P-NIC. An *UP-NIC* is used for communication with a parent node, while a *DOWN-NIC* for communication with the child nodes. The number of UP-NICs and DOWN-NICs is based on the proportion of traffic on two sides. A gateway node, for instance, dedicates all its interfaces as DOWN-NICs, while a non-gateway node typically distributes its interfaces equally across the two subsets.

Each node is responsible for the assignment of channels to its DOWN-NICs. An UP-NIC of a node, on the other hand, is associated with a unique DOWN-NIC of the parent node, and gets the same channel as the parent’s DOWN-NIC. The mapping of child nodes to DOWN-NICs is also done by the parent node, so as to balance the loads across all DOWN-NICs. To assign channel(s) to its DOWN-NICs, a node needs to determine the total usage of each of the channels within its neighborhood. Each node, therefore, exchanges its individual channel-usage information with all the neighbors within its interference range, and aggregates the received information to estimate the overall usage of each of the channels. The node then finds the least-interfering channel(s) in its neighborhood, and assigns them to its DOWN-NICs. Traffic patterns can evolve over time. All nodes periodically adjust their channel assignments and child-node to interface mappings, to suit the link capacities to the latest traffic patterns.

Each node also uses one of its NICs as *P2P-NIC*. P2P-NICs of all nodes in the network are assigned to the same channel; this helps maintain connectivity for infrequent peer-to-peer traffic as well as control traffic irrespective of the channel assignments done to the UP-NIC and DOWN-NIC. In an alternate design, one can also time-multiplex the control and peer-to-peer traffic over the UP-NICs and DOWN-NICs.

## IV. PERFORMANCE EVALUATION

We evaluated the effectiveness of *Hyacinth*’s channel assignment and routing algorithms through both NS-2 simulations and experimentation with the 9-node *Hyacinth* prototype. To conduct the simulations, we modified NS-2 to support multiple WLAN cards on each wireless node and to support dynamic channel assignment. The evaluation metric for most of these experiments is *cross-section goodput*, which is defined as the sum of all useful bandwidth between traffic aggregation nodes in a wireless mesh network and their corresponding gateway nodes. In each of the following experiments, 20 of the mesh network nodes were chosen as aggregation nodes that generate constant traffic. The amount of traffic generated by each node is randomly chosen between 0 and 3 Mbps. The ratio between interference and communication range was fixed

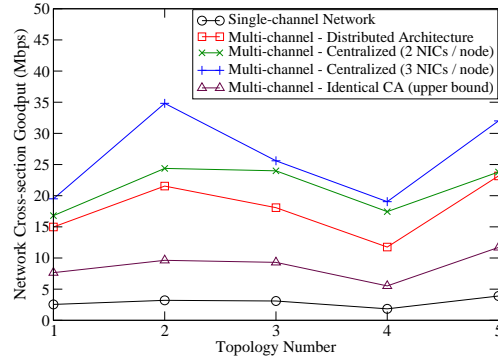


Fig. 2. The network cross-sectional goodput for 20 randomly chosen flows directed towards 4 gateway nodes uniformly placed in a 60-node network randomly sampled on a 9x9 grid. Five different 60-node network topologies are tested.

at 2. Depending on the topology and the node position, each node could communicate with up to 4 neighbors. All experiments were conducted with RTS/CTS mechanism enabled. Unless otherwise specified, each node in the mesh network is equipped with 3 NICs, and the total number of physical channels is assumed to be 12.

### A. Throughput Improvement of Multi-Channel Mesh Networking

We evaluated the performance improvements due to the ability to deploy multiple NICs per wireless network node by constructing five different 60-node mesh networks, each of which is randomly extracted from a 9x9 grid. We uniformly placed 4 gateway nodes within each extracted mesh network. The traffic was generated from 20 randomly chosen aggregation nodes. The results in Figure 2 show that even with a simple identical channel assignment scheme (section 2), deploying multiple NICs on each node improves the network goodput by a factor of up to 3 when compared with conventional single-channel architecture. *Hyacinth*’s distributed channel assignment algorithm boosts the throughput improvement over the single-channel architecture to a factor of between 6 to 7. Intuitively, equipping each wireless mesh network node with multiple interfaces helps to increase concurrency by breaking a collision domain into multiple collision domains each operating in a different frequency range. This division of collision domain is the key reason for the nonlinear goodput improvement (6-7 times) with respect to the increase in the number of NICs (from 1 to 3).

In the 3-NIC *Hyacinth* network, one NIC is dedicated to the peer-to-peer and control traffic, and therefore only two NICs are left on each node for backbone traffic. The fact that a 3-NIC wireless mesh network using *Hyacinth*’s distributed channel assignment/routing algorithm performs roughly the same as a 2-NIC wireless mesh network using the centralized channel assignment/routing algorithm [10] suggests that the throughput difference between centralized and distributed channel assignment/routing algorithms is very small. Adding one more NIC per node to the 2-NIC wireless mesh network

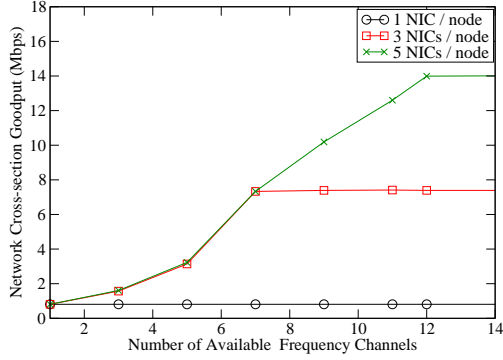


Fig. 3. Impact of varying the number of non-overlapped channels and NICs per node on the network cross-section goodput of the proposed wireless mesh network architecture using distributed channel assignment and routing.

using the centralized channel assignment/routing algorithm results in a throughput gain of upto 40%.

All these measurements exclude the *control traffic* that goes over the P2P-NIC. Our measurements show that backbone maintenance traffic consumes a very small portion, approximately 1%, of this P2P-channel bandwidth. Moreover, the data channels and interfaces are unaffected by this control traffic.

### B. Impact of Available Resources

1) *Interfaces per Node and Frequency Channels*: The number of available channels in 802.11b/g is 3, and is 12 in 802.11a. The results in Figure 3 show the effects of varying the number of non-overlapped channels on the network goodput of the multi-channel wireless mesh network architecture. The experiments were conducted on a 49-node 7x7 grid network with one gateway node at the center. We also vary the number of NICs per node to understand how increasing the number of NICs can improve the utilization efficiency of the non-overlapped channels. As expected, the network goodput increases monotonically with the number of non-overlapped channels when the the number of NICs per node is kept constant, because a collision domain can be broken into more non-interfering collision domains. When each node has 3 NICs, the network goodput saturates when the number of non-overlapped channels is 7. At this point, the gateway nodes become the bottleneck. Increasing the number of NICs on the gateway nodes further improves the goodput, as shown in the 5-NIC-per-node curve. When each node has 5 NICs, the network can use up to 12 non-overlapped channels before its performance starts to saturate.

2) *Number of Gateway Nodes*: Earlier experiments show that the gateway nodes are the main bottleneck. Figure 4 shows how the network goodput improves as more gateway nodes are deployed. In these experiments, conducted on an 81-node grid network, gateway nodes were added one by one. In the final configuration, 9 gateway nodes were uniformly distributed across the network. Each additional gateway node improves the network goodput because it adds more capacity to relay traffic generated by aggregation nodes to the wired network.

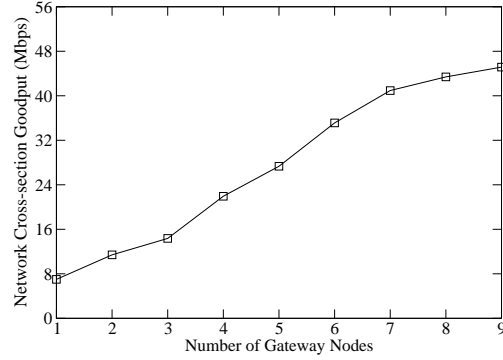


Fig. 4. Network goodput increases as more gateway nodes are deployed.

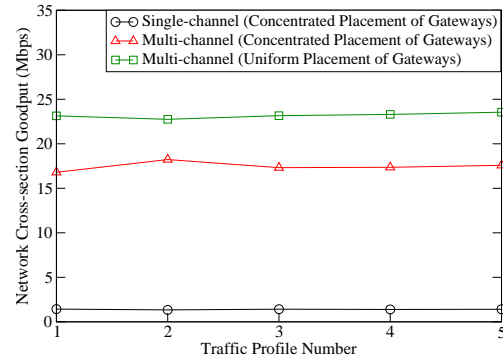


Fig. 5. Impact of locations of gateway nodes. Two different placements of four gateway nodes are tested: In one case four gateway nodes are uniformly distributed across a 64-node wireless mesh network, and in the other all four gateway nodes are concentrated at the center of the 64-node wireless mesh network. Each traffic profile corresponds to a different offered load.

3) *Placement of Gateway Nodes*: Figure 5 shows the effect of the gateway nodes' location on the network goodput. In these experiments, 4 gateway nodes were placed in a 64-node grid network. The network goodput is better when the gateway nodes are uniformly placed across the network. However, even concentrating all four gateway nodes at the center of the mesh network provides most of the performance gains. Placing multiple gateway nodes in close physical proximity is desirable because it reduces the wire installation cost and maintenance cost. *Hyacinth's* distributed channel assignment algorithm automatically adjusts the channels assigned to the gateway nodes and their neighboring nodes according to their physical locations, in order to maintain sufficient bandwidth to the gateway nodes.

### C. Design Decisions

1) *Load-Balancing Routing*: Figure 6 compares the effectiveness of different load-balancing routing algorithms. These experiments were carried out on a 64-node 8x8 grid network, with 4 gateway nodes one on each corner of the network. 20 traffic aggregation nodes were chosen in a skewed manner, so that they lie closer to 2 of the gateway nodes. In gateway load

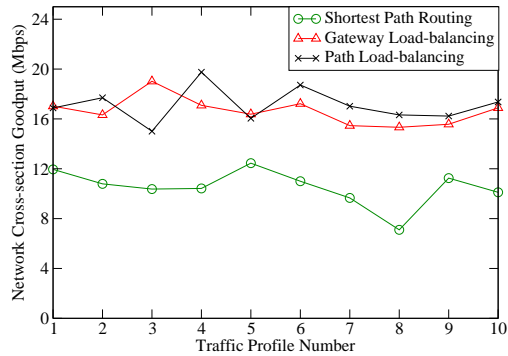


Fig. 6. Effectiveness of different load-balancing routing schemes. In gateway load balancing, only the available bandwidth on the gateways is considered when choosing a path. In path load balancing, the end-to-end available bandwidth on the path between a node and a gateway is used as the criterion.

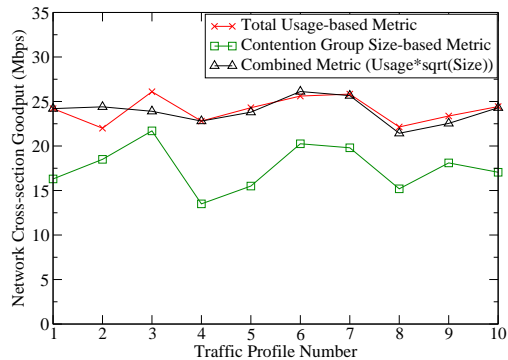


Fig. 7. Impact of different channel load metrics used in child-node-to-interface mapping and interface-channel mapping.

balancing, only the available bandwidth on the gateways is considered when choosing a path. In path load balancing, the end-to-end available bandwidth on the path between a node and a gateway is used as the criterion. With shortest path routing algorithm, two of the gateway nodes are overloaded. The load-balancing routing algorithms divert part of the traffic to the remaining gateway nodes, and thus increase the overall network goodput. Performance of *path load balancing* is only somewhat better than that of *gateway load balancing*, which suggests that gateway nodes are the main bottlenecks. In real networks, intermediate links could also form bottlenecks as well, and path load balancing is expected to produce routes that can achieve better network-wide load balance.

2) *Interface-Channel Assignment Scheme*: When assigning channels to interfaces, a node needs to compare the loads of radio channels in its vicinity. Channel load can be defined in terms of the number of interfaces using the same channel, or the summation of numbers of bytes per second transmitted on the channel by interfaces in the neighborhood, or a combination of the two. Figure 7 shows the impact of using different channel load metrics in the channel assignment algorithm on the network goodput. Summation of per-interface transmission

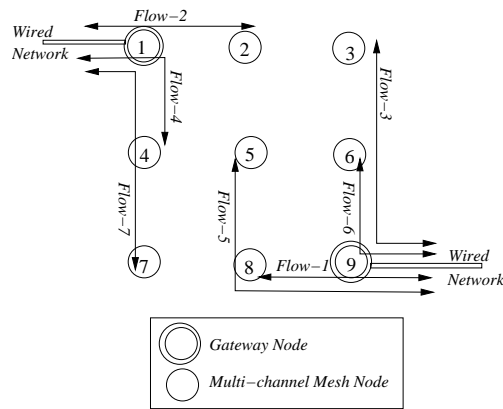


Fig. 8. The topology of the 9-node Hyacinth prototype. Nodes 1 and 9 are the gateway nodes connected to the wired network. Each node has 3 wireless interfaces - two 802.11a PCI cards for backbone traffic, and one 802.11b PCI card for peer-to-peer and control traffic.

rates on a channel is a more accurate measure of channel load than the number of interfaces sharing same channel. However, sum of per-interface transmission rates does not capture channel contention, which depends on the number of nodes sharing a common channel. The fact that the combined metric produces better performance for some traffic profiles suggests the need to take this contention effect into channel load estimation.

#### D. Prototype Performance Measurements

Figure 8 shows the 9-node *Hyacinth* prototype we have built. Each node is equipped with two 802.11a PCI NICs for data communication, and one 802.11b PCI NIC used for peer-to-peer and control channel. Each NIC is equipped with a separate external antenna. Use of external antennas minimizes the inter-channel interference among different NICs placed on the same node [10]. Two of the nodes, Node 1 and 9, are used as the gateway nodes and are connected to the wired network. All the other nodes in the testbed can access the department network as well as the Internet by going through the gateway nodes. For evaluation purposes, we operate the prototype in two different modes - the multi-channel mode, and the single-channel mode. In the single-channel mode, only one of the two 802.11a NICs is used for communication. Each node runs a user-level daemon that communicates with similar daemons running on other nodes, and participates in the routing and channel assignment protocol. The daemon sets the channels and IP addresses of each interfaces, as well as the routing tables. The actual forwarding of packets is tackled by the kernel IP layer itself.

1) *FTP Upload/Download Bandwidth*: We measured the performance of FTP flows on the non-gateway nodes that download data simultaneously. Figure 9 shows the aggregate bandwidth achieved by these flows in the multi-channel case is 55.58 Mbps, which is about 5 times the aggregate throughput in the single-channel mode (11.32 Mbps). The performance of upload FTP flows also showed similar performance gains. These results match closely the NS-2 simulation results for a 9-node multi-channel network, and thus validate our previous simulations results. The throughput improvement over single-

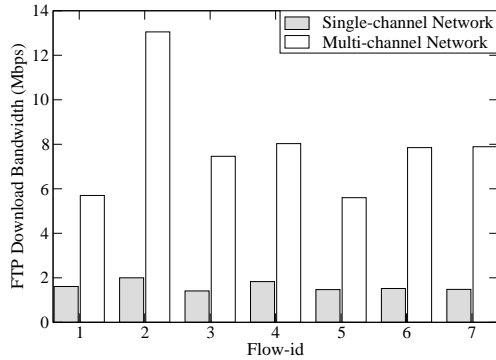


Fig. 9. Download FTP throughput comparison between the single-channel mode and the multi-channel mode of the Hyacinth prototype.

channel architecture is limited to 5 times as opposed to 7 times because the prototype testbed is smaller. For larger networks, the throughput gain should be higher.

2) *Failover Time*: Figure 10 shows how the bandwidth of network flows evolves over time when a node in the *Hyacinth* testbed fails. The period of time during which the network flows' bandwidth drops to zero represents the failure recovery time. In all the experiments, Node 6 was made to fail, and Node 3 failed over to Node 2 as its new parent. In the 2-card case, Node 3 pre-established a backup connection with Node 2 using an extra interface, and hence only route changes needed to be propagated up to the gateways. In the 1-card case, Node 3 has to establish a new link-layer connection with Node 2. Due to driver implementation problem, the channel switching latency on the Windows platform is extremely high. To remove this overhead from the measurement, we kept an interface of Node 3 on the same channel as Node 2. With an appropriate driver implementation, channel switching should only add about 50-100 msec [9].

In all the cases, the failure recovery time is between 600 to 700 msec. Out of these 150 msec is the failure detection time, which was done by exchanging HELLO packets between Node 3 and Node 6. The propagation time for the route-change request is about 1 msec. Most of the remaining time goes into changing the routing tables.

## V. CONCLUSION

The *Hyacinth* project attempts to expand the role of IEEE 802.11 WLANs beyond providing access network to mobile devices. Specifically, it focuses on the development of a wireless mesh network architecture that can eliminate most, if not all, of the wiring overhead associated with access point deployment. The fundamental research question that the *Hyacinth* project aims to answer is whether it is possible to build a wireless enterprise backbone using standard 802.11 hardware. To increase the aggregate capacity of a wireless mesh network so that it can serve as an effective backbone, *Hyacinth* features a multi-channel mesh network architecture that uses multiple commodity IEEE 802.11 NICs per mesh network node, each operating at a different channel. We briefly describe *Hyacinth's* distributed channel assignment and

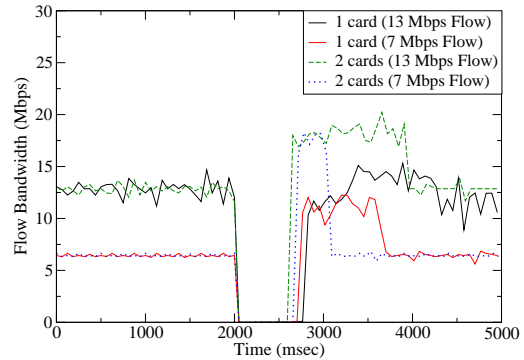


Fig. 10. Measured failure recovery time for Node 3 (Fig 8) when Node 6 fails. Failure occurs at time 2000 msec, and network recovers within 700 msec.

routing algorithms for exploiting multiple radio channels in a mesh network and present a comprehensive evaluation of their effectiveness. We show that with load-aware channel assignment and routing algorithms, equipping each node with just 3 NICs can increase the network bandwidth by a factor of 6 to 7, as compared with single-channel mesh network architecture.

The *Hyacinth* project is in its early stage. Once we stabilize *Hyacinth's* multi-channel mesh networking feature, we will integrate the device-transparent horizontal handoff and seamless vertical handoff features into the *Hyacinth* prototype. Next, we will design and implement the crosscheck-based secure routing mechanism to protect a wireless mesh network from compromised nodes. Finally, we will incorporate QoS mechanisms specifically designed for wireless mesh networks to support multimedia applications such as VoIP and to avoid transport-layer unfairness among best-effort connections.

## REFERENCES

- [1] A. Tzamaloukas and J. Aceves; "A Receiver-Initiated Collision-Avoidance Protocol for Multi-channel Networks"; Proc of Infocom, 2001.
- [2] Asis Nasipuri and Samir R. Das; "A Multichannel CSMA MAC Protocol for Mobile Multihop Networks"; Proc. of IEEE WCNC, 1999.
- [3] "IEEE 802.1D - MAC Bridges"; <http://standards.ieee.org/getieee802/download/802.1D-1998.pdf>
- [4] E.M. Royer and C-K Toh; "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks"; IEEE Personal Comm, April 1999.
- [5] V. Bahl, A. Adya, J. Padhye, A. Wolman. "Reconsidering the Wireless LAN Platform with Multiple Radios"; FDNA workshop in SIGCOMM 2003.
- [6] P. H. Hsiao, A. Hwang, H. T. Kung, and D. Vlah; "Load-Balancing Routing for Wireless Access Networks "; Proc. of Infocom 2001.
- [7] Hossam Hassanein, Audrey Zhou; "Routing with load balancing in wireless Ad hoc networks"; Proc. of ACM intl workshop on Modeling, analysis and simulation of wireless and mobile systems, 2001.
- [8] S.J. Lee and M. Gerla; "Dynamic Load-Aware Routing in Ad hoc Networks"; Proceedings of ICC 2001.
- [9] S. Sharma, N. Zhu, T. Chiueh; "Low-Latency Handoffs for Infrastructure mode Wireless LANs"; To appear in IEEE JSAC, Sp issue on All IP Wireless Networks, 2004.
- [10] A. Raniwala, K. Gopalan, T. Chiueh; "Centralized Algorithms for Multi-channel Wireless Mesh Networks"; ACM SIGMOBILE MC2R, Vol 8, No 2.
- [11] R. Karrer, A. Sabharwal, and E. Knightly; "Enabling Large-scale Wireless Broadband: The Case for TAPs"; Proc. of HotNets 2003.
- [12] BelAir Networks; <http://www.belairnetworks.com>
- [13] Strix Systems; <http://www.strixsystems.com/>
- [14] Firetide Inc; <http://www.firetide.com/>