Experimental Study of Measurement-based Admission Control for Wireless Mesh Networks

Dhruv Gupta† Daniel Wu† Chao C. Chen† Chen-Nee Chuah† Prasant Mohapatra† Sanjay Rungta‡
†Graduate Group in Computer Science, University of California Davis
‡Intel Corp.
†{dhgupta, danwu, cchchen, chuah, pmohapatra}@ucdavis.edu, ‡{sanjay.rungta}@intel.com

***Abstract*—The increased deployment of wireless mesh networks (WMNs) should be complemented by a robust resource management scheme that can provide performance guarantees to mission-critical applications. Several admission control schemes have been presented for wireless LANs and wireless ad-hoc networks. However, wireless mesh networks, with static wireless back-bone and multi-hop communication, pose new design challenges. Evaluation of existing admission control schemes has been done primarily via simulations, which often do not have accurate models for capturing interference between adjacent wireless links and nodes. In this paper, we develop light-weight monitoring modules to measure current network/traffic conditions and estimate end-to-end path delay, which is then incorporated in our admission control decision. We utilize a novel layer-2 packet forwarding mechanism, based on the Wireless Distribution System (WDS) for WMNs. We evaluate our scheme via experiments conducted on a test-bed consisting of IEEE 802.11a-based nodes that form a wireless mesh. Results show that our proposed scheme can provide performance assurance without incurring too much control overhead.**

I. Introduction

Wireless mesh networks (WMNs) have witnessed a tremendous growth over the last few years, both in terms of commercial installations, and as a topic of academic research [1]. Numerous wireless mesh network test-beds have sprung across universities and research groups [2] [3] [4]. A lot of work has been done in terms of understanding the behavior of mesh networks, analyzing the impact of various factors such as interference and number of hops on its performance and utilizing multiple radios and multiple channels to improve the performance of mesh networks [5] [6] [7].

Recently, the focus has been more on providing quality-of-service (QoS) in wireless mesh networks. Unlike wired networks, it is not an easy task to provide guaranteed end-to-end services in WMNs due to their highly dynamic nature. Characteristics such as interference, shared transmission medium and the broadcast nature of wireless communications, add to the complexity of designing QoS schemes for WMNs.

One key QoS mechanism is *admission control*, which helps regulate the amount of traffic in the network to meet the performance guarantees for end users. Authors in [8] note that measurement-based admission control schemes are more appropriate for the present day soft real-time applications, as compared to parameter-based schemes that are based on worst case bounds of the traffic behavior. Since the parameters in a wireless network vary greatly with time, such a measurement-based scheme can continuously keep track of the network/traffic statistics and adapt QoS provisioning based on more accurate data.

Therefore, we propose to adopt a measurement-based approach to admission control (MBAC) for WMNs. Our goal is to control the amount of traffic load such that the admitted flows will experience tolerable end-to-end delay under load, which is similar in spirit to the Assured Forwarding service in Diff-Serv architecture [9]. This is important since round-trip-time (RTT) more than 1000ms will affect the performance of most web applications [10]. Note that we do NOT intend to provide hard delay bounds (such as those required by real-time applications). We consider an enterprise or community wireless mesh network where the nodes are static access points (APs) that have two interfaces: one to serve the clients, and one to communicate with adjacent APs to form the wireless back-haul. One of these APs that has connection to the wired Internet serves as a gateway and centralized controller (CC).

We do not claim novelty in the admission control algorithm itself. Instead our main contributions lie in the implementation and experimental study of a pure measurement-based approach to this problem:

- We develop and implement a measurement-based admission control scheme for WMNs. End-to-end path delay is used as a metric to make admission control and path selection decisions. The centralized

1-4244-1455-5/07/$25.00 © 2007 IEEE

controller keeps track of passively monitored per-hop delays and uses these to calculate end-to-end path delay in the mesh. The measured delay includes waiting time in the output queue of a node and delay due to retransmission. Hence, it captures the notion of congestion, channel contention, and wireless link quality.

- We also implement a novel layer-2 packet forwarding substrate based on WDS in our WMN test bed. This allows us to set up both control and data paths easily across the wireless mesh, providing the flexibility of path selection at higher layers. For example, different path selection algorithms can be supported at network or application layers to optimize different performance metrics, independent of the underlying packet forwarding mechanism. For control packets, routes are selected via shortest path scheme while routes for data packets are selected by our admission control algorithm based on the delay requirements.
- We evaluate the effectiveness of our delay estimation technique and admission control scheme through experiments. An experimental approach enables us to capture the impact of important factors such as interference on our measurements and decision making process. Our test bed contains working prototypes of topology discovery protocol, layer-2 packet forwarding, per-hop monitoring service, path delay inferencing technique, and admission control module.
- Through experiments, we quantify the performance of a simple measurement-based admission control approach in terms of its effectiveness to achieve tolerable delays for admitted flows. We also characterize the associated cost such as admission delay and communication overhead.

The rest of the paper is organized as follows. Section II outlines some of the related work and also gives the motivation behind our work. The proposed methodology is explained in section III. Section IV contains the experimental evaluation of the proposed scheme. Section V concludes the paper and outlines some of the future work.

II. Related Work and Motivation

Most of the existing work on providing QoS in wireless networks focus on single-hop wireless LANs for mobile ad-hoc network environment. In [11], the authors propose a QoS-aware routing scheme that incorporates admission control based on approximate bandwidth estimation. They estimate the residual bandwidth at each node to support new flows. Authors in [12] propose a modification to the existing AODV protocol to incorporate admission control and bandwidth reservation. Another approach proposed in [13], Contention Aware Admission Control protocol (CACP), utilizes the knowledge of available resources not only at the local node but also at all nodes in the contention neighborhood of the node. They introduce the concept of c-neighborhood available bandwidth, which takes into account the node's neighbors in its carrier sensing range. Another similar scheme was proposed in [14], where the carrier sensing range of each node is modified to enable it to measure the available bandwidth in the surrounding region. They test the scheme for single hop ad-hoc networks and show how it can be extended to multi-hop networks. Another closely related approach is presented in [15]. They propose an admission control scheme for ad-hoc networks, integrated with a hop-by-hop ad-hoc routing protocol. They use a passive monitoring technique to estimate available channel bandwidth.

Motivation for our study: We notice that all of these approaches have been validated using simulations, which unfortunately do not accurately capture important characteristics of a wireless network, such as interference. Existing simulation models often make erroneous assumptions, such as interference range being twice the transmission range, interference range is circular, or only two-hop neighbors interfere with each other. In addition, most of the schemes proposed above are for ad-hoc networks, which have different characteristics compared to planned wireless mesh networks. It is even more challenging to model the intra- and inter-flow interference in a multi-hop environment of wireless mesh networks. The study in [16] points out that interference is not a binary concept as assumed in most works. An interference model should be able to capture the amount of interference between two links, and not just a binary characterization of whether they interfere or not.

The most accurate way of characterizing and taking into account the impact of interference in a wireless mesh network is to actually measure these quantities. Therefore, we propose to adopt a measurement-based admission control scheme, where the decisions are based on measured data that reflect the dynamic traffic and network conditions. We verify our approach via experiments, using a ten node test-bed, to effectively capture the impact of interference and transmission overhead on our approach.

Design decision to use delay as a metric: Most of the admission control schemes proposed so far use channel utilization as the metric. Once the channel utilization is estimated, the available bandwidth is calculated using the equation:

$$Av\_Bw = (1 - Channel\_Util) * Capacity \qquad (1)$$

The problem with this approach is that measuring the capacity bound of a wireless network may not be feasible in practice, while the theoretical bounds are often not very realistic. Most schemes proposed so far assume the network capacity to be a fixed value, which is inaccurate due to time-varying nature of wireless channels. In addition, measuring channel utilization is not an easy task in practice. Existing bandwidth estimation techniques usually employ some form of active measurements, thereby adding undesirable overheads.

For these reasons, we decided to use delay as the metric for admission control instead of previously proposed metric of channel utilization. Per-hop delay is easier to measure and can be obtained through passive monitoring techniques that do not involve high overheads in the measurement process.

III. Proposed Methodology

We consider a wireless mesh network, represented by a graph $G(V, E)$. V denotes the set of nodes in the mesh network, while E denotes the set of links between nodes that can communicate with each other. One of the nodes serves as a gateway node and has a wired connection to the wide-area Internet. The gateway node also serves as a central controller and runs the centralized admission control algorithm. The monitoring infrastructure deployed in the network will provide the gateway with the current network statistics, in terms of per hop delays, which will also be used as input to our admission control algorithm.

We target a small wireless mesh network with a maximum of 25 - 30 nodes, with each node being within a maximum of three to four wireless hops from the gateway node. Centralized schemes are apt for such small size networks. A distributed approach may have the advantage of not having a single point of failure, but suffers from large messaging over-heads among the nodes. Note that several of such WMNs can collectively form a larger mesh network, where a hybrid approach to resource management is feasible, i.e., the gateways will make centralized resource control decisions for the local clusters, while coordinating in a distributed manner with other gateways, for coarser grained load-balancing. How to scale the admission control and QoS provisioning to larger WMNs will be part of our future work.

Our goal is to maintain tolerable end-to-end delays for each client under heavy load conditions. In [10], it has been noted that for most web applications, the ideal round trip delay should be less than 800 ms for good performance. Also, it mentions that a RTT of more than a 1000 ms will affect the performance of the application. Subsequently, we decided to use 1000 ms as the maximum tolerable delay for each client that associates with the network. Again, it should be noted that we do not aim to provide hard-delay guarentees needed for real-time applications like VoIP, but only soft delay assurance.

Given a client request, with D_{user} being the maximum tolerable delay (= 1000 ms), our objective is to make a Yes/No admission control decision for that client and determine the best path to carry traffic from the client to the Internet gateway. Let RTT_i be the round trip delay on hop i of the network. Then the admission control decision will be yes if the algorithm can find the shortest path P from the source to the destination such that for all i hops on path P:

$$\sum_i RTT_i + \beta < D_{user} \tag{2}$$

where β is a hysteresis parameter. If the above condition is not satisfied, it means that the network does not have enough resources to support the new client and the request will be rejected. β is introduced to make our scheme more conservative and minimize the potential impact of the new flows on the delay of pre-existing flows in the network.

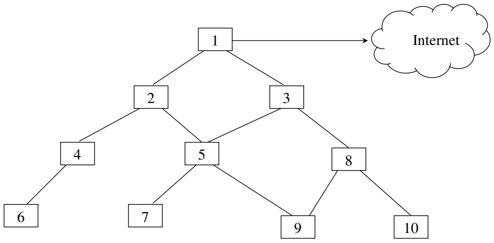
There are three key components in our proposed approach:

1) **Topology discovery:** To decide what routes to allocate to the clients, the central controller should be aware of the network topology and any changes that occur due to addition or deletion of nodes and link failures.
2) **Measurement of required traffic/network statistics:** The monitoring service at each WMN node or access points (APs) constantly measure the required parameters and forward them to the central controller (CC). We need to balance the tradeoffs between accuracy (for CC to estimate current network conditions) and overhead (additional control messages).
3) **Admission control algorithm:** The central controller uses the measurement reports from APs and checks if a new client can be allocated a path that satisfies the requirement.

A. Topology Discovery Protocol

The central controller needs to have a global view of the network topology. We thus need a protocol for neighbor discovery and setting up of control routes from each node to the gateway. We use a HELLO message protocol to achieve this. Every node periodically transmits 'Keep Alive' messages, which informs the node's neighbors about its existence. The nodes use 'Add' and 'Delete' messages to propagate any changes in their routing tables caused by node addition or deletion or due to link failure. These messages are broadcast throughout the network to ensure that each node has a global view of the network topology. Figure 1 shows the topology of our test-bed and the corresponding topology table built at Node 1 using our protocol.

The central controller marks a special flag in its broadcast messages to inform the nodes of its special status. Each node starts up by broadcasting messages that specify a list of their neighbors. This information is used by the nodes to build their routing tables. Each node then sets up a static control route to the gateway node



Node ID	Neighbor Flag	Neighbor List
2	Yes	4,5
3	Yes	5,8
4	No	6
5	No	7,9
6	No	NULL
7	No	NULL
8	No	9,10
9	No	NULL
10	No	NULL

Fig. 1. Test bed lay out and topology table at node 1

(through shortest-path computations). The routing tables are in the form of MAC addresses specifying the next hop to be taken for a given source destination pair. These control routes are used for sending control messages, such as route association request and replies, between each node and the central controller. Each node also reports monitoring information to the central controller via the control route. Another alternative is to flood the measured data and it would eventually reach the central controller. However, this would consume too much network resources.

If a new node is added to the network, it will start sending out 'Keep Alive' messages. The first 'Keep Alive' message of this node will have a flag indicating that it is a new node. The neighboring nodes will broadcast their entire routing table to this node so that the new node has an updated view of the topology. After this, the protocol will behave in the normal fashion. If three consecutive 'Keep Alive' messages are not received from a node, it is assumed that the node or the link is down. The neighboring node that is unable to hear from the failed node will update its routing table and send out an update message to inform the other nodes. The packet formats are not included here due to lack of space.

B. Measurement of per-hop delay

Our admission control scheme is based on periodic measurement updates received by the central controller from all nodes. The quantity that we intend to measure is the per-hop delay on each link in the network, due to reasons mentioned in Section II. By measuring this parameter, we can get an estimate of how much end-to-end delay would a particular client experience in the mesh network. We can then decide whether to accept the new user or not by comparing this data with D_{user} (Eq. 2).

We plan to utilize the information coming from the IEEE 802.11 MAC layer. The MAC layer provides some quality metrics such as number of successful and failed transmissions, modulation rate of the transmitted packet, time taken to transmit a frame and so on. This cross-layer information can be utilized to quantify the current network state and make informed decisions about admission control.

In a wireless network, a node that has data to send will first sense the channel for a fixed amount of time. If the channel is sensed busy due to other transmissions, then the node will back off for a random amount of time. The back off interval is increased exponentially every time the node senses the channel to be busy. After the back-off period expires, the node again waits for a constant time sensing the channel and transmits its data when the channel is idle. The receiving station will wait for a very short time period before sending out the MAC layer acknowledgement for the received packet. If the packet transmission fails, then no ACK will be sent out. A time out will occur at the transmitting station and the packet will be re-transmitted.

For each packet, we mark the time at which the packet was put in the output buffer queue. We then wait for the MAC-layer ACK for the packet to be received and record that time. The difference between these two times gives us the total delay involved in transmitting a packet on that link. This time difference accounts for the following delay components involved in the packet transmission:

- *Time spent by the packet in the queue.* The queuing delay can be due to other packets in front of the queue that are waiting to be transmitted. In addition, if the node senses the channel to be busy, the packet will remain in the queue for a longer time. Our measurement captures this queuing delay that accounts for interference between and across flows. If one node is using the channel, then the other nodes would have to back off and the delay involved would be captured in our measurement.
- *Transmission and propagation delay.* Once the packet is at the head of the transmit queue, it will be transmitted on to the channel and reach the destination. The corresponding delays involved in transmitting the packet and its propagation to the receiver are captured by our measurements.
- *Retransmission delay.* If the packet reaches the receiver and is successfully decoded, an ACK is sent back and the receive time for the ACK is noted. However, if the packet is lost or corrupted, then the

packet will be re-transmitted at the MAC layer. The time involved in the successful delivery and decoding of the packet, including any re-transmissions, is captured in our measurements.

Each node tracks the per hop delay involved in transmitting a packet on a link and sends it periodically to the central controller. The central controller maintains a data base of the per-hop delays in the network and estimates the end-to-end delay for a given path in the mesh network.

In order to measure the per-hop delay, we modified an open-source driver for wireless radios. We are using the madwifi driver [17] for our test bed. Necessary modifications were made to the driver so that each node can measure its round trip delay and report it to the central controller on a periodic basis. Each node maintains an exponential moving average of the delay and sends the data to the central controller, where it is used to calculate the end-to-end path delays.

C. Centralized Admission Control Algorithm

The central controller receives periodic measurement data from various nodes in the network. This data is sent via unicast along the control routes from each node to the central controller. The admission control module at the central controller uses the measurement data and discovered topology to estimate end-to-end path delays in the WMN. It uses the estimated path delay to make admission control decision and assigns routes to clients. The topology graph is constantly updated. If a new node is added in the network, the change of routing tables at some of the nodes will be propagated through the network and eventually reach the centralized controller. Similarly, when a node fails, it will stop broadcasting the 'Keep Alive' messages. The node's neighbor will detect the failure and send an update to the central controller. The central controller will update its tables and re-route the traffic that was going through the failed node.

The measurement data is also transmitted periodically from each node to the central controller. The central controller matches the data to the links in the topology graph. The RTT for each link is the weight of the link. We use a simple modified uniform-cost search algorithm to determine whether a path exists from the source to the destination. A uniform-cost search algorithm is a modified breadth-first-search algorithm for weighted graphs. At each node, a breadth-first-search is performed to search for the link that satisfies the requirement. If the links at the first node itself cannot satisfy the user requirement, then the algorithm can stop as no path exists in the network that can satisfy the user's delay requirement. If any of the links at the first node satisfy the delay requirement, then that node is added to the path and the next hop is considered. The above procedure is repeated till the destination node is reached. If in the end a path is found from the source to the destination, it means that the network has sufficient resources to handle the new traffic and the new client can be accepted. A reply is sent back to the client and the corresponding route is setup. As mentioned earlier, the admission of a new client will adversely affect the RTT of the existing clients in the network. In order to minimize this impact, we have introduced the hysteresis parameter, β, in Equation 2. By controlling the value of β, we can minimize the impact of over-admitting flows and the temporal traffic fluctuation.

D. Client Association, Route Setup and Packet Forwarding

Suppose a client wants to join the network and sends an association request to a particular AP. This AP needs to contact the central controller, which will decide whether enough resources are available in the network to satisfy this new request without adversely affecting the existing flows in the network. For this purpose, the AP sends a client association request to the controller. This request contains the MAC address of the incoming client. When the controller receives a client association request, it checks the MAC address of the AP from which the request has come. This AP will be the destination and the controller will be the source for the path that has to be decided. The controller will perform the steps explained in the previous section in order to find a path that satisfies the delay requirement of the user. It will compare the end-to-end delay of the path, calculated from the per-hop delays reported by the network nodes, with the delay threshold, D_{user}. If the controller is able to find a path satisfying Equation 2, then the new client request can be accepted. In this case, it sends an association reply message to the AP saying that the client can be accepted. Upon receiving this request, the AP will send an association frame to the client.

If the client is accepted, then the controller needs to setup a route for this client. It does this by sending route creation messages to each node on the path that has been chosen. This message contains the MAC address of the source (i.e. the client) and the destination. For each hop, it will set an entry in the routing table specifying the next hop to be taken for the particular source-destination pair. The forward and the reverse paths may be different depending upon the bandwidth availability (as reflected by the estimated path delays). So for each client, there will be one entry with client MAC as source MAC and gateway MAC as destination MAC for outgoing traffic and one entry with client MAC as destination MAC and gateway MAC as source MAC for incoming traffic.

Our routing scheme is based on the Wireless Distribution System (WDS). Under the WDS mode, an IEEE frame consists of 4 address fields, corresponding to the original sender, original receiver, the current transmitter and the current receiver stations. We build client-based MAC address tables at each node. For each packet, the

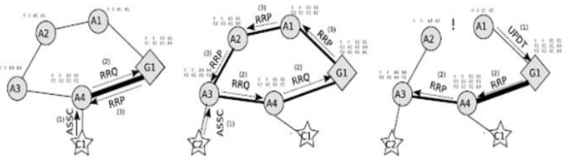


Fig. 2. WDS based Layer 2 routing

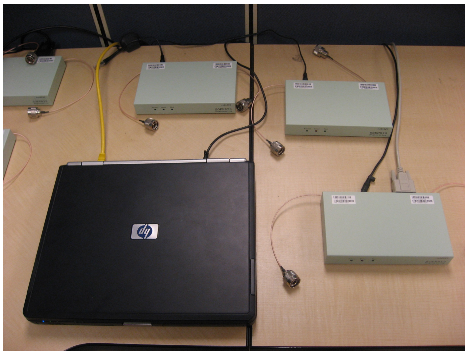


Fig. 3. Equipment used for experimental evaluation

AP will match the packet against the appropriate entry in the table and forward the packet to the next hop. The MAC tables are populated by the central controller as explained above. Figure 2 shows the details. The first figure shows the steps involved in basic client connection. The second figure shows a more complex case where the central controller assigns the path A1 A2 A3 C2 to the second client. In the third figure, the link between A2 and A1 fails. A1 updates the central controller, which re-computes the route and sends out the necessary update messages. The reader can refer to [18] for further details on the route setup and forwarding.

IV. Performance Evaluation

In order to test the scheme, we use a 10 node test bed in our laboratory (topology shown in Figure 1). Our test bed consists of nine Soekris boards and 1 HP nc6000 laptop (Figure 3). The laptop serves as the gateway node and the central controller. The Soekris boards serve as the mesh APs. Each of the Soekris boards in our test bed consists of 2 radios. One radio is configured to be in the access point mode and serves the clients that want to access the network. The other radio is configured to enable inter-AP communication. This radio connects with the wireless back haul. Each of the radios in the AP mode is configured on separate channels while the backbone radios are all configured on the same channel in order to minimize interference between the two. At present, we evaluate our scheme for a single channel back haul network. However, having multiple channels will not effect the implementation of our scheme.

We currently consider a centralized admission control scheme We assume that the central controller co-exists with the gateway node. The gateway node is the one that provides wired connectivity from the mesh network to the Internet. The nodes are configured to work in 802.11a mode. This is done to prevent interference between our test bed and other existing 802.11 b/g networks in the building.

The test bed was built using the small Soekris [19] net4826 embedded devices. This device runs on a 266 Mhz 586 processor with 128MB SD-RAM main memory and 64MB compact flash for the OS and other storage. They are optimized for wireless communications with dual Mini-PCI Type III sockets. We selected the Ubiquiti Networks SuperRange2 802.11b/g 200mW Atheros Wireless mini-PCI card as the wireless radios for our devices. These boards are driven by a custom built Linux distribution using a 2.6 Linux Kernel. The kernel and filesystem are optimized for running on the embedded systems without sacrificing speed. We use the madwifi-ng driver from Madwifi.org [17] on our APs due to their level of programmability. Each Soekris board acts as a single node in the network. Linux Bridging is used to bind the multiple network interfaces to one IP address.

First, we verify that our network discovery protocol deals effectively with link failures and addition of new nodes. In order to evaluate the effectiveness of our proposed scheme, we perform the following set of experiments:

- **Test the correctness of the delay measurement scheme.** We compare the delay measurements obtained from the driver against those obtained from a network monitoring tool.
- **Check for message overheads.** The network discovery protocol and the relaying of measurements data to the central controller involves certain messaging overheads. We evaluate the amount of over heads that we are generating.
- **Evaluate the admission control scheme.** The effectiveness of our scheme is evaluated in terms of improvement in throughput and delay for the clients.

A. Delay Measurement Accuracy

In order to confirm the accuracy of our delay measurement scheme, we compared the measurements against those taken using the network performance tool thrulay [20]. Thrulay (stands for throughput and delay) is a tool that can be used to measure the throughput and round-trip time between two end-points. The measurements were carried out with no traffic (Table I), as well as with some data traffic in the back ground (Table II), to make sure that the measurements are accurate in both the cases. In the second case, three clients were attached to the network, each generating UDP traffic at a constant rate of 5 Mbps. As can be seen , the measurements obtained are fairly accurate for the purpose of admission control. The results shown are for 1 wireless hop and 2 wireless hops. For the case of 2 wireless hops, we compared the RTT returned by thrulay against the sum of the delays measured at the two hops.

	Measured from Thrulay	**Measured from driver**
1 hop	20.5 ms	17.6 ms
2 hop	45.7 ms	41 ms

TABLE I
COMPARISON OF DELAY MEASUREMENTS WITH NO TRAFFIC

	Measured from Thrulay	**Measured from driver**
1 hop	41.2 ms	38.7 ms
2 hop	76 ms	66.3 ms

TABLE II
COMPARISON OF DELAY MEASUREMENTS WITH 15 MBPS TRAFFIC LOAD

It should be noted that the measurements obtained from the driver represent the delay the the Layer 2, which should ideally be less than the delay obtained from Thrulay (delay at the transport layer).

B. Delay in admission control decision

One of the first measurements that we conducted was to test the delay introduced by the centralized admission control scheme. As explained earlier, when an AP receives a client association request, it sends a message to the central controller where the admission control decision is made. This decision is relayed back to the AP, which will then accept or reject the client. We measured the delay involved in this process. Varying the amount of background traffic introduced into the network, we measured the delay for different number of wireless hops. The traffic introduced in each case was UDP traffic at a constant rate. The results are shown in Figure 4. Each measurement was performed over a time period of 30 seconds. It was found that our scheme did not introduce any significant delay over and above the normal delay in the network.

C. Improvement in throughput and delay

In order to evaluate the effectiveness of our admission control scheme, we measured the improvement in the throughput and delay of the clients. Figure 5 shows the variation in the throughput of three clients attached to our mesh network test bed. The first client was attached one wireless hop away from the gateway node (at node 2 in Figure 1) while the other two clients were attached at two wireless hops from the gateway node (nodes 4 and 5 in Figure 1). The two clients were introduced in the network a few seconds after the first client. As can be seen, the first client achieves a respectable throughput before the other clients become active. From the point when the other two clients start transmitting, the throughput for the first client drops drastically and also shows a lot of variation. The throughput achieved by the other two clients is also very low.

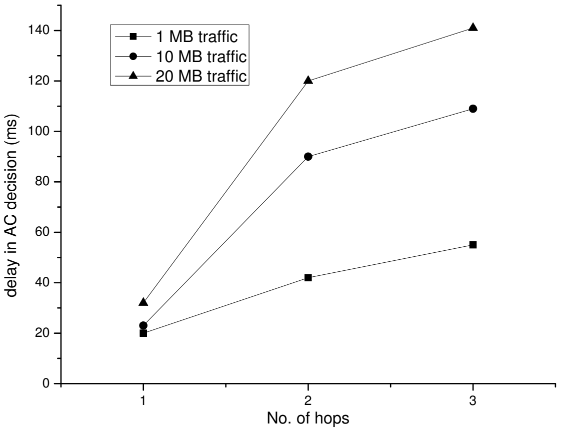


Fig. 4. Delay in centralized AC decision

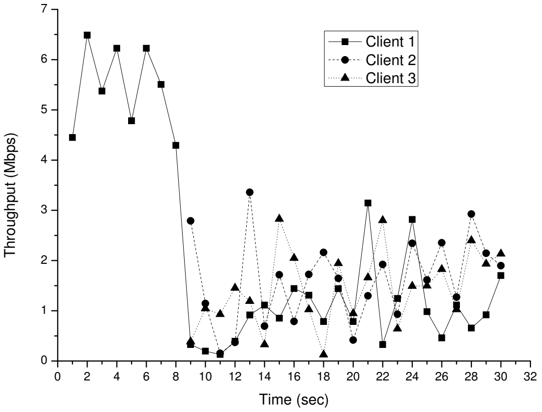


Fig. 5. Throughput variation without admission control

Figure 6 shows the throughput variation when admission control is implemented. The end-to-end delay threshold within the mesh network, D_{user} was set to 1 second (for reasons explained in section III). The value of the hystersis parameter β was set to 100 ms. Then the first client was introduced, followed by the second client after a few seconds. Since the central controller perceived that it can support these two clients, it accepted both the requests. However, the third client was rejected as no path existed in the network which could satisfy the delay requirement. As can be seen, the throughput available to the two clients is more than the case where there was no admission control in place. Also, the throughput exhibits fewer variations. This result implies that the end-to-end path delay is a good performance indicator to base the admission control decisions on, given that we do not measure throughput directly.

A similar comparison for the round trip delay is shown in Figures 7 and 8. Without the admission control, the delay for the first client increases and also shows a lot of

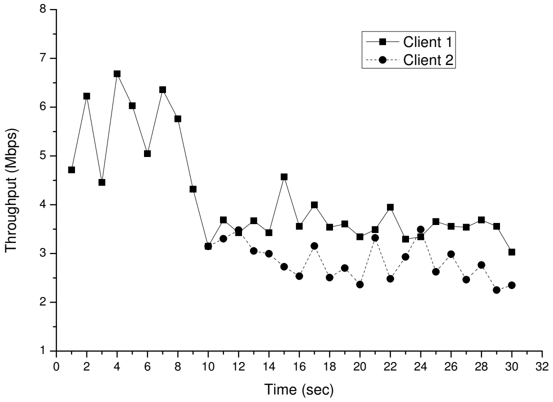


Fig. 6. Throughput variation with admission control

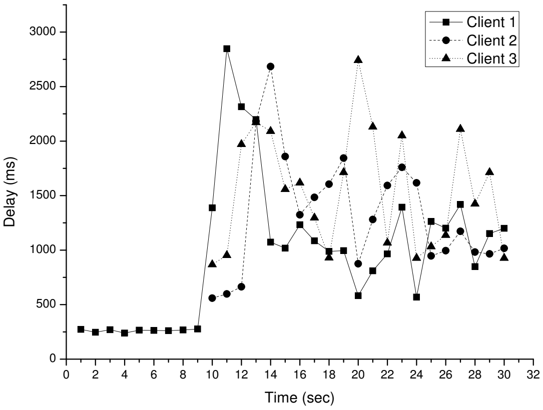


Fig. 7. Delay variation without admission control

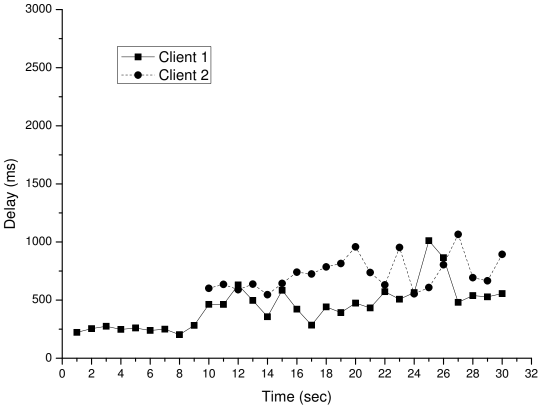


Fig. 8. Delay variation with admission control

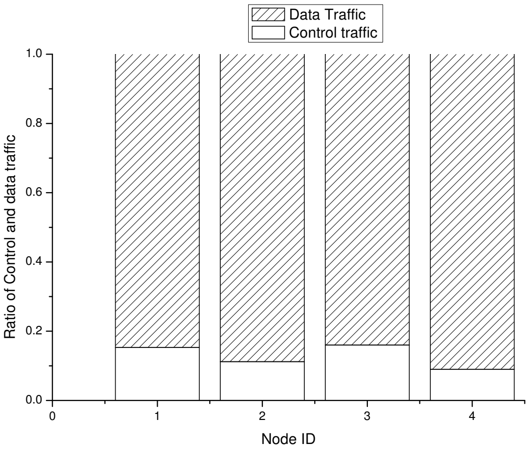


Fig. 9. Overhead vs. data traffic

variation. The delay faced by the other two clients is also very large. With the admission control scheme in place, the third client request is rejected, thereby providing the first two clients with lower delay and smaller delay variations. Note that admission of the second flow does not cause the delay of the first flow to exceed D_{user} (1000 ms). However, the end-to-end path delays do vary over time. There are transient spikes above 1000ms between t=26 and t=28 seconds due to fluctuations in wireless link quality. This uncertainty makes it extremely difficult to provide hard delay bounds.

D. Measurement of communication over heads

We use passive measurements in order to measure the per-hop delays at each node in the mesh network. As a result, the measurement process itself does not introduce any overheads in the network. However, since our scheme is centralized, it involves periodic reporting of measurement data from the mesh nodes to the central controller. In addition, each node periodically sends out 'Keep Alive' messages. Also, when a client tries to associate with the network, client association request and reply messages are exchanged between the mesh APs and the central controller.

All these messages contribute to communication overheads. It is important to make sure that the amount of over heads is kept at a minimum. Figure 9 shows the ratio of the control traffic to the data traffic. The measurements were carried out at four random nodes in the test bed. Clients were associated and disassociated at random. As it can be seen, the amount of control traffic is very low as compared to the amount of data traffic in the network.

V. Conclusion and Future Work

In this paper, we present a measurement-based admission control scheme for wireless mesh networks. The problem of provisioning QoS in wireless networks is not a trivial task, owing to their highly dynamic nature. A measurement-based scheme that constantly monitors the network, will incorporate the current network state in the decision making process. The key contribution of our work is that we evaluate our scheme experimentally using a ten node test-bed. Unlike the existing proposals that have been evaluated via simulations only, our scheme does not suffer from any inaccurate interference models and is not based on any assumptions in terms of channel capacity for mesh networks. Some of the important lessons learned as part of our work are:

- Utilizing cross-layer information for resource management in wireless networks is an effective approach. We measure per-hop delays at the MAC level and use it for network layer admission control and path selection. Several such parameters can be measured at the lower layers and used for different performance optimizations at the higher layers.
- Our Layer-2 scheme gives us an advantage in terms of implementing different higher layer optimizations using different performance parameters, without changing the underlying packet forwarding mechanism. For example, control packets are routed along shortest paths from each node to the gateway while the data packets are routed along routes chosen by our admission control algorithm, based on delay requirements. Both types of packets use our Layer 2 scheme for packet forwarding.
- Our experiments show that a centralized, measurement-based admission control scheme is effective in controlling traffic load in WMNs, while incurring very little communication overheads. Delay in making the AC decision is of the order of one round-trip-time.
- Our measurement-based approach allows us to provide soft QoS guarantees, i.e., tolerable delay under heavy traffic load (similar to Assured Service in Diff-Serv). Setting a more conservative delay threshold in the admission algorithm can help prevent performance degradation of admitted flows due to fluctuations in flow rates, channel quality and interference levels. However, it is not intended for providing hard delay bounds.

The problem of resource management in wireless networks needs to be addressed with their increasing popularity. Our scheme shows how to utilize measurement information for providing better service to end users. In the future, we plan to extend this scheme to larger mesh networks with multiple gateway nodes and also perform comparisons with a distributed scheme. We also plan to evaluate other metrics for admission control such as channel utilization and loss rates.

References

[1] I. Akyildiz and X. Wang, "A survey on wireless mesh networks," *Communications Magazine, IEEE*, vol. 43, pp. S23–S30, 2005.

[2] P. De, A. Raniwala, S. Sharma, and T. Chiueh, "Design considerations for a multihop wireless network testbed," *Communications Magazine, IEEE*, vol. 43, 2005.

[3] D. Wu, D. Gupta, S. Liese, and P. Mohapatra, "Qurinet: quail ridge natural reserve wireless mesh network," in *Proceedings of the 1st international Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH)*, 2006.

[4] J. Bicket, D. Aguayo, S. Biswas, and R. Morris, "Architecture and evaluation of an unplanned 802.11b mesh network," in *Proceedings of the 11th Annual international Conference on Mobile Computing and Networking (MobiCom)*, 2005.

[5] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom)*, 2004.

[6] A. Raniwala and T. cker Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," in *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM)*, 2005.

[7] M. Kodialam and T. Nandagopal, "Characterizing the capacity region in multi-radio multi-channel wireless mesh networks," in *Proceedings of the 11th annual international conference on Mobile computing and networking (MobiCom)*, 2005.

[8] L. Breslau, S. Jamin, and S. Shenker, "Comments on the performance of measurement-based admission control algorithms," in *Proceedings of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM)*, 2000.

[9] W. W. J. Heinanen, F. Baker and J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, June 1999.

[10] ITU, 2003. [Online]. Available: http://www1.cs.columbia.edu/∼andreaf/new/documents/other/T-REC-G.114-200305.pdf

[11] L. Chen and W. B.Henzelman, "Qos-aware routing based on bandwidth estimation for mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 23, pp. 561–572, 2005.

[12] H. Zhu and I. Chlamtac, "Admission control and bandwidth reservation in multi-hop ad hoc networks," in *Computer Networks*, 2005.

[13] Y. Yang and R. Kravets, "Contention aware admission control for ad hoc networks," *Transactions on Mobile Computing, IEEE*, vol. 4, pp. 363–377, 2005.

[14] I. D. Chakeres and E. M. Belding-Royer, "Pac:perceptive admission control for mobile wireless networks," in *First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QSHINE)*, 2004.

[15] L. Luo, M. Gruteser, H. Liu, D. Raychaudhuri, K. Huang, and S. Chen, "A qos routing and admission control scheme for 802.11 ad hoc networks," in *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks (DIWANS)*, 2006.

[16] J. Padhye, S. Agarwal, V. N. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *ACM/USENIX Internet Measurement Conference (IMC)*, 2005.

[17] "Madwifi:multiband atheros driver for wifi." [Online]. Available: www.madwifi.org

[18] D. Gupta, J. LeBrun, P. Mohapatra, and C. Chuah, "Wds-based layer 2 routing for wireless mesh networks," in *Proceedings of the 1st international Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH)*, 2006.

[19] "Soekris engineering." [Online]. Available: www.soekris.com

[20] "Thrulay: throughput and delay measurement tool." [Online]. Available: http://shlang.com/thrulay/