Toward the use of local monitoring and network-wide correction to achieve QoS guarantees in mobile ad hoc networks

Harpreet Arora
Department of Computer Science
Drexel University
Philadelphia, PA-19104
Email: harpreet@drexel.edu

Lloyd Greenwald
Department of Computer Science
Drexel University
Philadelphia, PA-19104
Email: lgreenwa@cs.drexel.edu

***Abstract*— This paper presents the exploration of novel mechanisms towards providing Quality of Service (QoS) guarantees in mobile ad hoc networks. We study mechanisms that provide differentiated services to packets of varying priority traffic flows. These mechanisms do not require any central coordination and do not depend on any specific protocols at the physical, MAC, or network layers. Nodes independently monitor the rates of the highest priority flows and signal corrective mechanisms when these rates fall outside of specified local bounds. Triggering conditions for network-wide corrective mechanisms are designed to trade-off rapid reactive response to local QoS violations with control packet overhead. A range of corrective mechanisms are explored that attempt to maintain reactive response while improving total network utilization, including resources consumed by lower priority traffic. We provide simulation results that demonstrate the effectiveness of monitoring, reactive triggering, and basic and advanced corrective mechanisms. We discuss the extension of these novel mechanisms to a complete QoS solution for mobile ad hoc networks.**

I. INTRODUCTION

Multimedia applications, such as voice and video for military and disaster relief, drive mobile ad hoc network (MANET) development. Quality of Service (QoS) guarantees are crucial for successful deployment of multimedia applications on MANETs, yet research activity in this area has lagged far behind research topics such as routing protocol design. While robust routing protocols are necessary to ensure connectivity, the connected network may not be useful if the traffic flows of applications can't receive the bandwidth and timing guarantees they require. This is especially true for the highest priority traffic flows, which must be differentiated and whose requirements must be differentially treated compared to other lower-priority flows.

Differentiating and guaranteeing resources for high-priority traffic flows is an especially challenging task in the bandwidth-constrained MANET environment. In contrast to wired networks, decentralized QoS protocols are strongly preferred over centralized control and protocols must carefully limit bandwidth-consuming control packets. Furthermore, due to the research nature of many MANET protocols, including routing layer protocols and even PHY and MAC layer solutions, QoS solutions must either be independent of or adaptable to differing cross-layer solutions.

Existing QoS models for MANETs satisfy a strict subset of these requirements. Some QoS solutions are built directly into existing routing or link layer protocols, making them completely dependent on the adoption of those protocols in MANETs. Some QoS solutions provide guarantees that are too soft for the highest priority, application-critical traffic flows. Some QoS solutions lack adequate differentiation across traffic flows. The SWAN model [1], for instance, differentiates real time UDP traffic from TCP traffic. However, it does not differentiate across real time flows. Further, the assurances provided to these flows are soft, and could be void if the network conditions change.

This paper presents the exploration of novel mechanisms towards providing QoS guarantees in mobile ad hoc networks. Beginning from a modified version of the differentiated services architecture (DiffServ) [2], these mechanisms differentiate flows based on their resource requirements and prioritize them such that flows can be dropped or re-routed selectively during times of congestion. A DiffServ-based solution includes standard queues and schedulers that can be configured to provide distinct Per-Hop Behavior (PHB) to traffic of different classes.

We present and empirically evaluate novel mechanisms to provide improved QoS to the highest priority traffic flows. These mechanisms do not require any central coordination and do not depend on any specific protocols at the physical, MAC, or network layers. Nodes independently monitor the rates of the highest priority flows and signal corrective mechanisms when these rates fall outside of specified local bounds. Triggering conditions for network-wide corrective mechanisms are designed to trade-off rapid reactive response to local QoS violations with control packet overhead. A range of corrective mechanisms are explored that attempt to maintain reactive response while improving total network utilization, including resources consumed by lower priority traffic.

The performance of the model is compared with that of SWAN, as well as with a direct implementation of DiffServ

without additional mechanisms. Simulation results empirically demonstrate the effectiveness of monitoring, reactive triggering, and basic and advanced corrective mechanisms.

The rest of the paper is organized as follows: Section II explains the choice of DiffServ as a base model for traffic differentiation. Section III discusses the monitoring and corrective mechanisms built on top of DiffServ for maintaining the rate of high-priority traffic. In Section IV, we present some optimization schemes for improving total network utilization. Section V presents simulation results that demonstrate the effectiveness of our mechanisms and an improvement in performance over DiffServ and SWAN. Section VI categorizes the existing QoS schemes for MANETs and briefly describes the schemes in one of the categories. Finally we conclude with discussion and future work in Section VII.

II. Differentiating MANET traffic (DiffServ)

In the bandwidth-constrained MANET environment shared resources must be carefully allocated across traffic flows. Differing traffic flows have differentiated resource requirements and differentiated costs for not receiving desired resources. A first step in providing optimal resource allocation across all traffic flows is then to differentiate traffic flows according to requirements and priority.

As mentioned earlier, decentralized QoS protocols are strongly preferred over centralized control in MANETs, suggesting that the differentiation of traffic be done independently at each node. Further, protocols must limit bandwidth-consuming control packets. The DiffServ [2] model satisfies both of these requirements.

DiffServ was developed for the wired Internet as a *fully distributed*, *stateless*, and *scalable* QoS model. The model differentiates traffic into a fixed number of classes. The network is divided into *edge* and *core* nodes. Complexity is pushed to the edge of the network so that the core can be simple and fast. The nodes at the edge of the network are responsible for *classification* of flows and for *policing* them to ensure that the traffic complies with agreements (SLAs) made between users (flow originators) and service providers. Edge nodes also *mark* packets so that they can be differentiated by nodes in the core of the network. Nodes in the core provide *Per-Hop-Behavior* (PHB) depending upon the class of the packet, indicated by the *DiffServ code point* (DSCP) in the header of the packet. Nodes use RIO queue management and schedulers such as weighted round-robin to provide differentiation.

Applicability to MANETs: The DiffServ model, as defined for wired networks, cannot be directly applied to MANETs. There are several issues that need to be resolved, such as distinction between the edge and the core nodes, the definition of SLAs, and the number and type of classes to be supported. Intuitively, the source nodes play the role of edge routers and the relaying nodes act as core nodes. But, in a MANET any node can be a source and/or a relay for traffic. Therefore, each node must have the capability to act as an edge node and a core node, resulting in an increased complexity at each node. Further, DiffServ is a static model suited to wired networking needs. It does not define any mechanisms to dynamically adapt to the varying conditions of a MANET. In a wireless medium, the bandwidth along a link depends not just on the traffic along that link, it also depends on the traffic on all links which can potentially interfere with this link. Further, the resources vary at the mercy of environmental conditions. A complete QoS model for MANETs, therefore, requires mechanisms that can respond to these varying conditions. Despite these issues, the traffic differentiation mechanisms of DiffServ provide a good basis for a QoS model.

As such, we implemented a DiffServ-based queue for the differentiation of traffic into multiple priority classes. The queue consists of a classifier for differentiating flows into separate classes, a meter to monitor the rate of the incoming traffic, a policer to distinguish out-of-profile flows from the in-profile flows, a marker to mark the header of the packets with the DSCP, a separate physical queue for each class of traffic, and a scheduler to schedule the packets out of the queues. Figure 1 shows the position of these components in a MANET stack. In [3], the authors show that this infrastructure is well capable of traffic differentiation in a MANET environment. Given the DiffServ traffic differentiation mechanisms extended to MANETs, we can then focus on decentralized mechanisms that dynamically adapt network-wide resource allocation in response to changing network conditions. These mechanisms are presented in the following sections.

III. Monitoring and Reactive Correction

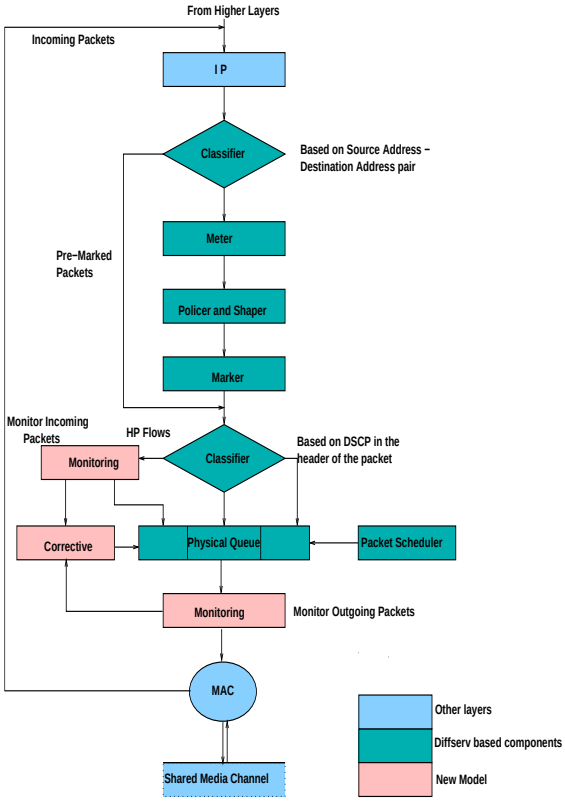

Fig. 1. Block Diagram of the New Model within a MANET node

Our primary focus is on providing improved QoS to flows of highest priority class, a feature absent in existing QoS models for MANETs. We assume that the flows of this class have clearly defined traffic characteristics such as packet rate and packet size. The model must ensure that each of the high priority flows is able to maintain these characteristics for the lifetime of connection. All such flows must therefore be **monitored** at each node through which they pass. Any divergence from the designated packet rate, for example, must trigger a **corrective action** by the node detecting it. We need to define elaborate mechanisms to monitor the activity of each high priority flow. The model also requires mechanisms that define how a corrective action needs to be taken to adjust any interfering flows affecting the resources of a high priority flow. Each of these **monitoring** and **corrective** mechanisms are dealt with separately. Before delving into the details of the mechanisms, we define the following terms:

Reception Range: This is the maximum range within which two nodes can communicate with each other directly in one hop. Packets transmitted by one node can be received and processed by the second node.

Interference Range: This is the range within which transmissions from one node can interfere with transmissions/receptions at another node to cause collisions. This range is usually greater than the reception range. Thus, when two nodes are outside the reception range of each other but within the interference range, packets transmitted by one node 'cannot' be received by the second node, but such packet transmissions can interfere and corrupt the transmissions/receptions at the second node.

H-Node: This is the first node along the path of highest priority flow under consideration which can receive packets of this flow from the previous hop at the desired rate but cannot transmit at the same rate because of interference from transmissions of other nodes in the network.

H-Flow: The highest priority flow carried by the H-Node

I-Node: Node carrying Medium or Low Priority traffic that potentially interferes with the high priority traffic carried by H-Node thereby reducing the resource availability to that flow.

A. Monitoring Mechanisms

In times of congestion and bandwidth-constraints, network resources must be concentrated on flows with the highest priority. Our monitoring mechanisms require each node to monitor the activity of each high priority flow it carries. In an effort to achieve tight guarantees for the highest priority flows, we bound the number of high priority flows through any node. This not only limits the amount of per flow state information within a node but also provides a way to ensure that the network does not try to support more high priority flows than resources permit. The packets of a high priority flow are generated and transmitted by the source at a fixed rate of r packets/sec (for example, a streaming video flow). Each node carrying a high priority flow monitors the number of packets received and transmitted for each flow within a window of w seconds. For normal operation, the number of packets received at any H-Node within any window must be greater than the threshold r_{th} and the number of packets transmitted within the same window must be greater than t_{th}. Here, r_{th} is the receiving threshold and t_{th} is the transmitting threshold.
Obviously, $r_{th} \leq r \cdot w$, $t_{th} \leq r \cdot w$ and $t_{th} \leq r_{th}$
If the above condition is not satisfied, monitoring mechanisms at the H-node signal the presence of interference affecting either the transmissions by the current node or the receptions by the next node along the path. The conclusion about the next node being affected by interference comes from the observation that although the current node is able to gain hold of the medium for transmission, the next hop is not able to receive those packets because of interference. The presence of interference must trigger immediate corrective action to adjust the flows causing it.

For our simulations, we restrict the number of H-Flows to 2 per node, the packet rate r to 32 packets/second. With a packet size of 80 bytes, this corresponds to an audio stream at 20480 bps. The monitoring window w is initialized to 2 seconds, the receiving threshold r_{th} is equal to the transmitting threshold t_{th}, both set to 55. At a rate of 32 packets per seconds, the maximum number of packets that can be received and transmitted within any window of size 2 seconds is 64. A threshold of 55 packets, therefore, corresponds approximately to a tolerance of 8.6%. Thus, a node signals the presence of interference, if within a window of 2 seconds, it receives at-least 55 packets but is not able to transmit more than 55 packets. A lower tolerance can be set by having the threshold value closer to 64. Although this would lead to a quicker triggering of corrective mechanisms, it may cause false triggering in case of small spikes in the congestion level, resulting in lower network utilization.

B. Corrective Mechanisms

Once the monitoring mechanisms signal the presence of interference obstructing the flow of highest priority traffic, the Corrective Mechanisms have the task of suppressing the interference to the level that the highest priority traffic is able to re-attain its traffic flow rate of r packets/sec. In doing so they must encounter three potential sources of interference.

Direct Range Interference Nodes: These nodes are within the reception range of the node carrying high priority traffic. Some of these nodes may be carrying medium priority or low priority traffic which reduces the bandwidth available to the high priority flow. Being in the direct reception range, these nodes can be informed relatively easily of the interference by broadcasting a message and corrective action can be taken quickly. These nodes are subsequently called as

DRI Nodes.

Nodes Outside Direct Transmission Range but within Interference Range: These nodes are within the interference range of the H-Node, but not in reception range. Thus, transmissions from these nodes interfere with the transmissions and receptions of H-Flow. In a random ad hoc network topology with random connections, it is likely that the prevention of interference from the DRI nodes is not sufficient to restore the rate of the H-Flow. It may be required to stop any interfering flows within these nodes. It is not straightforward to inform these nodes of the interference. Some of these nodes may be multiple hops away from the nodes carrying H-flow. Broadcasting a message to nodes two hops away may reach some of these interfering nodes, not necessarily all. However, as the control packet travels in an expanding ring, a 2 hop control packet may cause too many flows to stop, resulting in an exorbitant underutilization of the network resources. In our definition of corrective mechanisms, we assume that the network is sufficiently dense so that stopping interference caused by nodes in direct range is sufficient to restore the resource availability of the H-flow. However, we do not ignore this case completely, but leave this as a subject of our future studies.

Nodes Interfering with the Next Hop: Some of the nodes may lie within the interference range of the next hop. This implies that the next hop neighbor is not able to receive packets even though the transmitter is able to send them at the desired rate. In this case, the monitoring mechanisms of the next hop will not detect any interference, since the node does not receive at the desired rate. The interfering nodes may be several hops away from the node detecting interference. In this case, even broadcasting messages 2 hops away may not help. For our mechanisms, we ignore this case and assume that the case is too rare to be considered to add complexity to the protocol. We plan to explore this case in our future work.

The corrective mechanisms only suppress interference caused by the nodes within direct range of the H-nodes. Our simulations demonstrate that these corrective mechanisms are sufficient to provide improved performance of the H-flows while maintaining a high network utilization. The position of monitoring and corrective mechanisms within the MANET stack is shown in Fig. 1. Detailed working of monitoring and corrective mechanisms is presented in the next subsection.

C. Working of the model

When the monitoring mechanism signals the presence of interference, the H-Node broadcasts a *Squelch packet* with a TTL (time to live) of 1. All DRI nodes upon the reception of this packet stop the transmission of packets from any medium or low priority flows that they carry either as sources or relays. All packets that are in the queue of these nodes awaiting transmission, are dropped. In doing so, the transport layer at the source of any TCP flows carried by these nodes would cut down the rate of their flows, suspecting congestion in the network. However, the rate of these flows may build up soon, leading to interference. Also, the CBR sources would continue to generate and transmit packets that get relayed up-to these DRI nodes, where they get dropped. The corresponding source nodes of both the UDP and the TCP flows must therefore be informed of this link break, so that they can stop their transmissions. The DRI nodes explicitly send *Squelch* messages to all sources whose packets they carry for relaying. Note that the DRI nodes keep dropping any medium of or low priority packets that they receive for the next 5 seconds after they receive the $Squelch$ packet, to ensure that by the end of that period, the source nodes have received the $Squelch$ packets and taken the necessary corrective action. On receiving this message, the source nodes stop generating packets for a random time interval in the range of 0 to t_{sstop} seconds. Upon the completion of this interval, the source nodes re-initiate their flows, presenting their requests to the routing protocol. The sources stop for a duration of t_{sstop} seconds because we assume that a high priority flow lasts for an average of t_{sstop} seconds. For our simulations, we set this value to 100 seconds. However, this value can be changed based on the statistics of the duration of the high priority flow.

This method ensures a fast restoration of resources to the high priority flow. The restoration time is bounded by:

$$T_r = T_d + T_t + T_p \tag{1}$$

where,
T_d: Time it takes for the H-Node to *detect* interference
T_t: Time it takes for the H-Node to broadcast the Squelch packet to all the DRI neighbors. This includes the time it takes for the H-Node to create and *transmit* the packet to the MAC layer, time the packet spends in the queue before being taken by the MAC layer for transmission and time taken by the MAC layer to take control of the medium and broadcast the packet.
T_p: Time taken by the DRI node to *process* the squelch packet and take corrective action.

T_d depends on the time window used by the monitoring mechanism, of the order of a few seconds. T_t and T_p are negligible in comparison with T_d and can be ignored.

IV. Optimized Network-wide Correction

Although this mechanism results in a quick restoration of the flow rate, it may pre-empt more flows than necessary for the restoration, since the DRI nodes squelch all flows passing through them. This may cause a reduction in the network utilization. For example, if the H-Node has 5 DRI neighbors, 4 of which carry voice traffic at a rate of 20kbps and 1 carrying video traffic at 64kbps, it may be sufficient to stop the 1 flow carrying video traffic. However, the broadcast of the squelch packet to all 1-hop neighbors causes all of them to stop. Further, it may be possible to reroute the stopped flows around areas of congestion, without having to stop the flows at all.

A. P-Broadcast

This technique aims at improving network utilization by probabilistically selecting the nodes that must take corrective

action. A simple way to do this is to have a fixed value added to the $Squelch$ packet, called the $p-value$, whose value lies between 0 and 1. The I-nodes, upon receiving the $Squelch$ packet, compute a random number between 0 and 1. If the random number is greater than the p-value, the nodes take corrective action, otherwise, the $Squelch$ packet is discarded. Thus, if the p-value is set to 0.5, then only about 50% of the nodes will take corrective action. A p-value of 0 corresponds to the basic corrective mechanism where all nodes take corrective action.

An improvement of this technique would calculate the p-value based of the difference between the number of packets transmitted within the monitoring window and the maximum number of packets that could be transmitted within the same window. A large negative difference between the two values indicates the presence of high interference and thus a low p-value and vice-versa. This results is a larger number of nodes taking corrective action. The p-value is directly proportional to the difference or the *error*.

B. Selective Reject

Instead of broadcasting a control packet to all 1-hop neighbors, the H-Node enters promiscuous mode upon receiving a signal from the monitoring mechanism. In this mode, the signal taps all packets that are sent by the 1-hop neighbors and selectively asks a few nodes to stop sending. The victim nodes (nodes that have been asked to stop sending) may either be chosen randomly or based on their packet transmission rate. The restoration time with this method is bounded by:

$$T_r = T_d + T_{pm} + T_t + T_p \tag{2}$$

where T_d, T_t and T_p have the same meaning as defined for the basic broadcast mechanism. T_{pm} is the amount of time the H-node stays in *promiscuous mode* to detect the victim neighbors. Note that we can reduce T_{pm} by having the H-nodes be in the promiscuous mode at all time while they are carrying a H-Flow. However, this results in an increased processing at the node and hence an increase in the battery power consumption.

This scheme increases the network utilization by selectively asking the nodes to stop. However, the improvement in the utilization comes at the cost of increased time the node is required to spend in promiscuous mode to detect the victim neighbors. The accuracy of determining the victim nodes may depend upon the amount of time the node spends listening to the neighbor's transmissions. For example, if the victim nodes are chosen randomly based on the packets overheard from the ongoing transmissions in the past t seconds, the interference may not reduce substantially to bolster the rate of H-Flow to the required level. Or, it may result in cutting down more flows than necessary for the restoration. However, if the victim nodes are chosen based on their packet transmission rates, then the H-Node needs to stay in the promiscuous mode long enough to accurately determine the transmission rates of the neighboring nodes. This increased latency comes in addition to the increased complexity and power consumption of promiscuous mode mechanisms.

C. Re-Routing

Instead of stopping the lower priority flows completely, it may be possible to reroute these flows around the area of congestion. Then the overall network utilization will be less affected by the mechanisms that take corrective action. To do that, when the source nodes receive *Squelch* packets from the DRI nodes, they simulate link breaks at the routing layer. The routing layer then tries to look for an alternate path. The reactive protocols broadcast route request packets in the network, which are replied to by either an intermediate node that knows the path to the destination, or the destination node itself. Since the new routes must bypass the areas of congestion, where the corrective action was taken by the H-Node, it is important that the corresponding DRI nodes do not reply to or relay any of these route request packets. Hence, when the DRI nodes receive *Squelch* packets, they compute a random interval in the range of 0 and t_{stop} as computed by the source nodes upon receiving the *Squelch* packets. Within this interval, these nodes do not forward any Route Request packets that are broadcast by the source nodes. This ensures that the source nodes find alternate paths if any. Although this scheme seems to be potentially good, there are two concerns in the applicability of this approach:

1) The source nodes may be able to find an alternate route to the destinations, however, many of these new routes may start interfering with the H-Flow at different points of its path. Because of this, other H-Nodes may need to take corrective action. The H-flow may lose its packet rate more frequently than if using the earlier techniques.
2) In the case of mobile scenarios, the H-Flow may need to reroute. Since the DRI nodes do not forward any control packets within the randomly computed interval, they may drop control packets generated by the nodes for re-routing that flow. If one of the DRI nodes happen to follow on the new route, the H-Flow may not find a new route at all thus degrading its performance.

V. Advanced Mechanism Simulation Results

Simulation results presented in this section demonstrate the effectiveness of monitoring, reactive triggering and basic and advanced corrective mechanisms. Simulations were performed with a large variety of mobility and traffic scenarios. The results show that the model is able to achieve favorable bandwidth and delay performance. The performance of the new model is demonstrated with varying parameters. We also compare the performance of the new model with that of SWAN, DiffServ and MANET stack without any QoS framework.

A. Simulation Scenario

1) Mobility Model: The scenario consists of 50 nodes placed randomly in a 1500 m X 300 m rectangular area. We consider two kinds of scenarios: static and mobile. The mobility of the nodes within the mobile scenarios is based on the random-waypoint model. The nodes pause for an average of 10 seconds between consecutive movements. Once they

pause, they randomly choose a new direction, speed and distance they want to move.

2) QoS Model: The model as described in the previous section was implemented as a part of the CMU wireless extensions in NS-2. For improving the overall network utilization, we also implemented the P-Broadcast scheme with fixed p-values and Re-Routing Scheme mentioned in Section IV. The simulations evaluate the performance of the high priority flow and overall network utilization with basic model and with the optimization schemes. Within the graphs, the suffix $p0$ corresponds to the basic model, $p0.5$ corresponds to the model with P-Broadcast with fixed p-value of 0.5 and $p0.7$ corresponds to the model with P-Broadcast and a fixed p-value of 0.7. We also compare the performance of our model with that of SWAN and DiffServ. Although the Re-Routing technique is implemented, the results with that technique are not so favorable and hence not shown here. The issues with this technique are still under consideration.

3) Traffic Model and Traffic Differentiation: Although the model is capable of differentiating traffic into an arbitrary number of traffic classes, for the evaluation of the new model, we differentiate the traffic into two classes: high-priority class and low-priority class (or best-effort). Further, we restrict the number of high-priority flows to one. Essentially, it is a constant-bit-rate (CBR) flow generating 80 byte packets at a rate of 32 packets/sec (20kbps). The flows of low-priority class are CBR flows generating 800 byte packets at a rate of 20 packets/sec (128kbps).

4) Miscellaneous Considerations: We use AODV [4] as the routing protocol, since it is one of the most mature routing protocols available to the MANET community and it is the first protocol to be converted from an Internet Draft to an RFC. MAC 802.11 b is implemented at the MAC layer, which offers a maximum data rate of 11 Mbps.

5) Parameters Monitored: We measure the running rate of the high-priority flow as a function of time and compare it with other models. We also measure the percentage of packets that successfully reach the destination with an end-to-end latency of less than 100 ms and 200 ms. The overall network throughput is compared with different models and with different versions of our new model. Finally, we consider the overall control overhead with the various models and the total overhead of our new model.

B. Results

We considered both static and mobile scenarios for evaluating the performance of the model. The description of the scenarios and the corresponding results are presented below.

The *Static scenarios* consist of 50 nodes placed statically and randomly in a 1500m x 300m area as mentioned in Section V-A.1. One node is randomly chosen to act as a source of high-priority flow and a corresponding destination is chosen randomly. The number of low priority flows is varied from 9 to 13. This range is chosen because for the traffic characteristics mentioned in Section V-A.3, saturation sets in the network within this range. This allows us to evaluate the performance of our model under high-traffic conditions. The results are presented in Fig. 2, Fig. 3, Fig. 4, Fig. 5 and Fig. 6. The results presented in Fig. 4, Fig. 5 and Fig. 6 are produced by averaging over five random scenarios.

Within the mobile scenarios, 50 nodes are allowed to move randomly according to the random way-point model. The maximum speed of the nodes is varied from 5m/s to 25m/s. One node is randomly chosen to act as the source of high-priority flow. A random destination node is correspondingly chosen. The network also consists of 10 randomly chosen sources and destinations of low-priority flows. The results are presented in Fig. 7, Fig. 8 and Fig. 9 Each point within these graphs is produced by averaging results over five random scenarios.

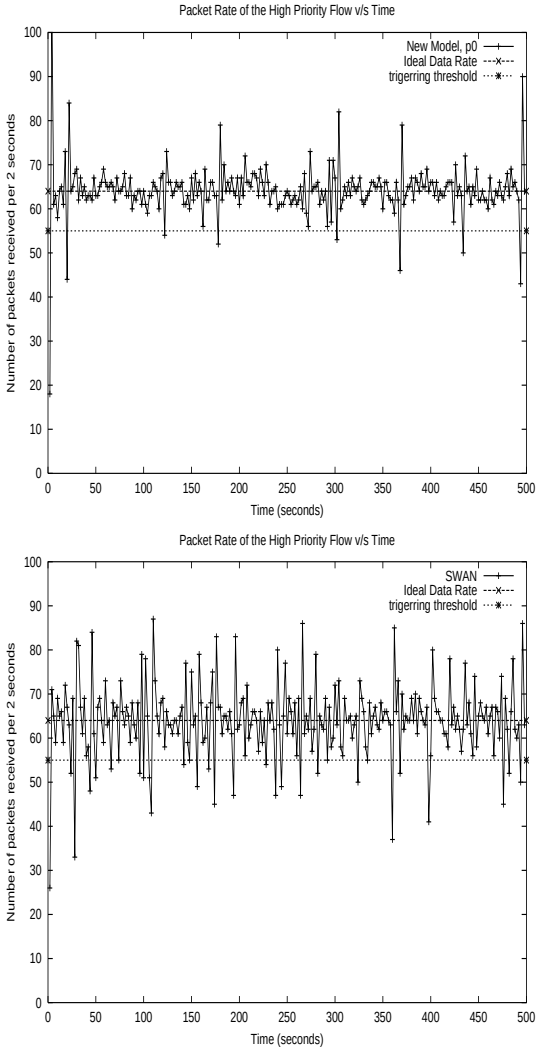

Fig. 2. Variations in the packet rate of the high priority flow with our new model (top) and with SWAN (bottom)

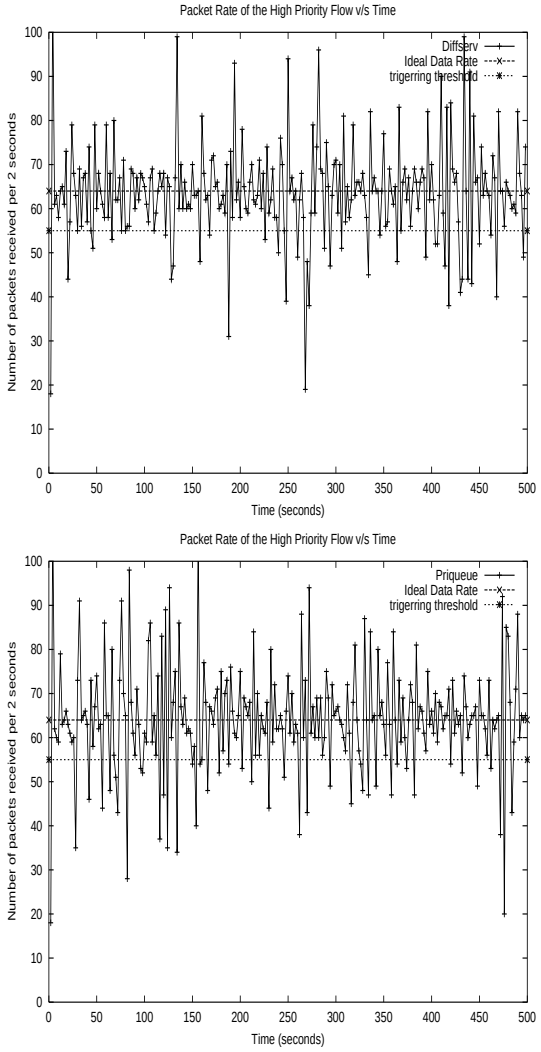

Fig. 3. Variations in the packet rate of the high priority flow with DiffServ (top) and without any QoS framework (bottom)

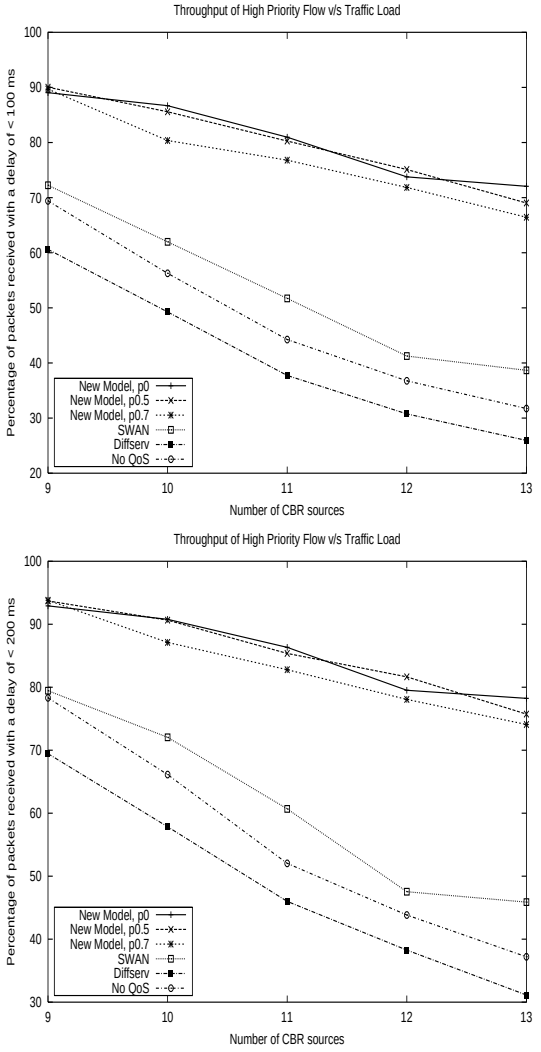

Fig. 4. Percentage of packets successfully received within delay bounds of 100 ms (top) and 200 ms (bottom)

C. Discussion

Figures 2 and 3 show the variations of the packet rate of the high priority flow as a function of time in a typical static random scenario. The rate of the flow with our new model is consistently higher than the threshold of 55 packets/(2 seconds) as compared to that with SWAN and DiffServ. The degree of variations can be easily controlled by adjusting the triggering threshold of the monitoring mechanisms.

Figure 4 shows the percentage of packets successfully received at the destination within delay bounds of 100 ms and 200 ms. Our new model outperforms SWAN by more than 15% for low loads and by a much larger margin at higher loads.

The improvement in the performance of the high priority flow comes at a price. Figure 5 shows the overall throughput of the network. This is measured as the ratio of the total number of packets received at all destinations to the total number of packets transmitted by all sources, taken as a percentage. The transmission of $Squelch$ packet results in some flows being stopped for some duration of time. This is implemented by dropping the packets being generated by the application agent at the new model at the source node itself. A reduction in the overall throughput is because of the corrective action taken by one-hop neighbors to maintain the rate of high-priority flow.

From Figure 6 it can be seen that the new model causes a reduction in the total control overhead. This is because, by broadcasting Squelch packets at points of heavy congestion, the model acts as a congestion control agent thereby reducing false link breaks. False link breaks are caused due to the mechanism of link layer feedback. If the link layer is not able to transmit a packet to the next hop even after a specific number of attempts, it assumes that the link to the next hop is broken and signals this to the network layer. The routing protocol then deletes the link from its table and initiates a fresh search for the destination that the broken link led to. During heavy congestion, the link layer may signal false link

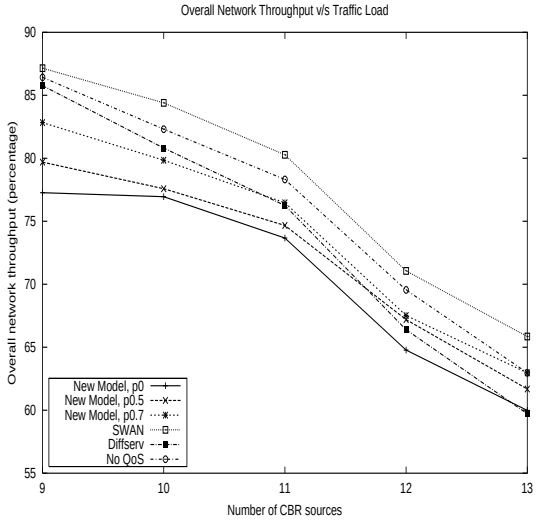

Fig. 5. Overall throughput of the network

breaks if it is not able to transmit the packets to the next node, even if the link still exists. The control overhead incurred by the new model itself is negligible. Figure 7 shows the number of packets successfully received at the destination with a delay of less than 100 ms and 200 ms. Even at high speeds, the performance of the model is consistent, keeping the throughput much higher as compared to DiffServ and SWAN. This indicates that the model performs as effectively with a mobile network as with a static network.

D. Improvement in Network Utilization using Network-wide Optimization

As seen from the results, the improvement in the performance of the high priority flow causes a reduction in the overall network utilization. A decrease in the total throughput indicates that some of the network resources are being underutilized. The throughput can be improved by increasing the p-value in the P-broadcast scheme. The P-Broadcast scheme causes only p percent of the one-hop neighbors to take corrective action. An incremental improvement in the network throughput with an increase in the p value can be seen from Fig. 5 and Fig. 8. However, an increase in the p value correspondingly degrades the performance of the high priority flow as evident from Fig. 4 and Fig. 7. An appropriate trade-off is an implementation issue and not studied in detail here.

VI. Related Work

Several QoS schemes for MANETs can be found in literature that are either an extension of the existing routing or link layer protocols or define independent mechanisms that lie between the routing and link layers. This section classifies the existing schemes into three categories and briefly describes schemes in one of the categories, relevant to the discussion in this paper.

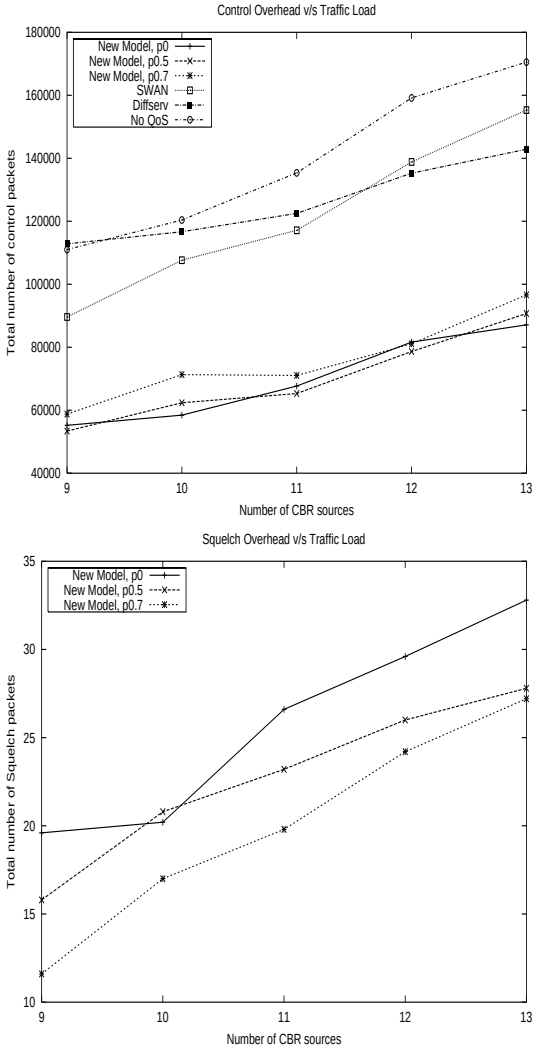

Fig. 6. Total control overhead (top) and Squelch overhead (bottom)

A. QoS aware routing protocols

Conventional routing algorithms aim at finding a minimum-hop route to the destination without considering factors such as congestion along the links and bandwidth availability along a path. Attempts have been made at devising routing algorithms that take QoS factors into account and find a path from the source to the destination that can satisfy the requirements. In [5] authors show that link interferences make the problem of finding a feasible path in a multi-hop MANET environment an NP-complete problem. This is because, to make a reservation along a path, it is necessary to reserve resources in all nodes that are in the interfering range of any node along the path from the source to destination. Nevertheless, there are many QoS aware routing protocols in literature that claim to provide a complete solution to the problem of finding a feasible path that satisfies the given requirements. Examples of such schemes are CEDAR [6] and QoS for AODV [7].

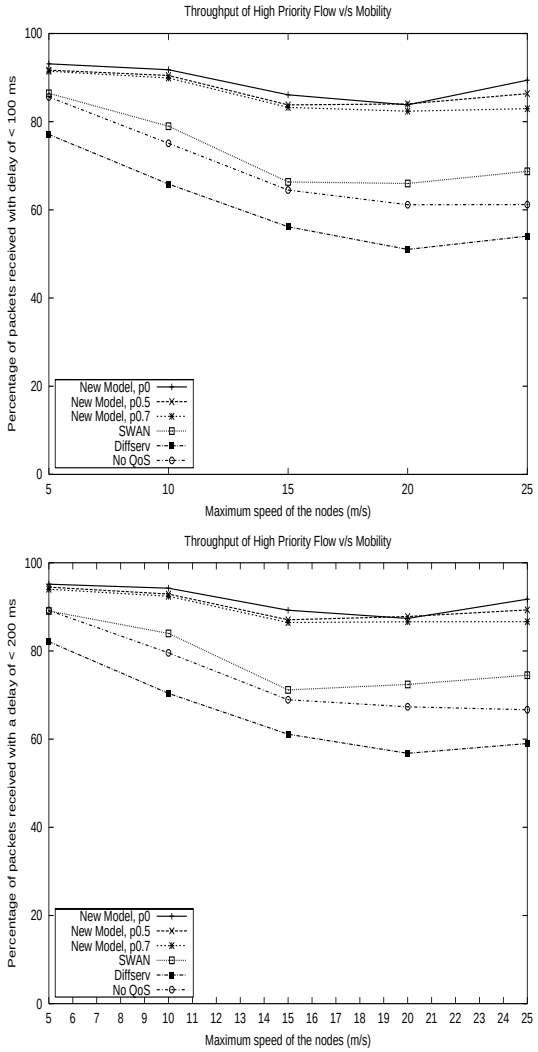

Fig. 7. Percentage of packets successfully received within delay bounds of 100 ms (top) and 200 ms (bottom)

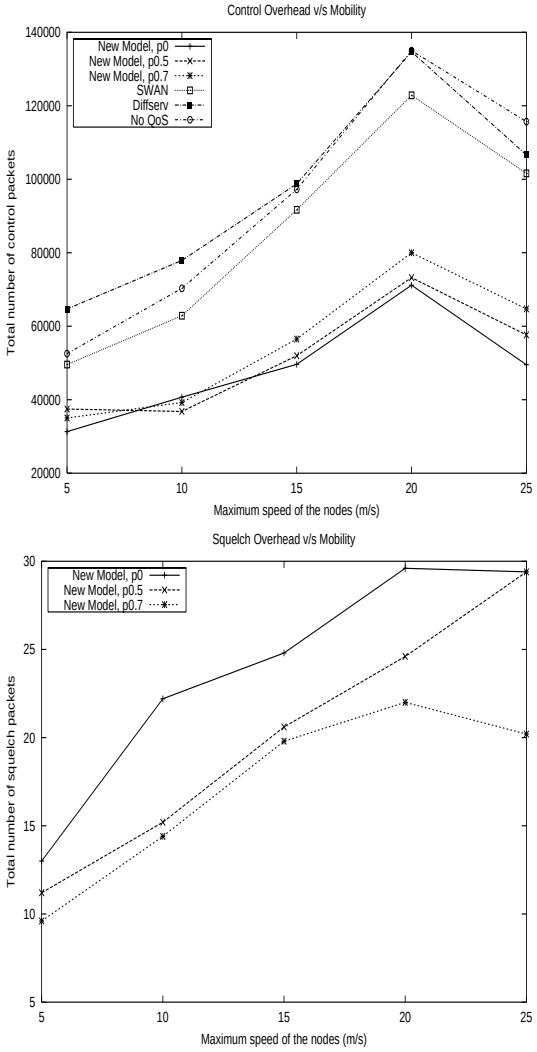

Fig. 9. Total control overhead (top) and Squelch overhead (bottom)

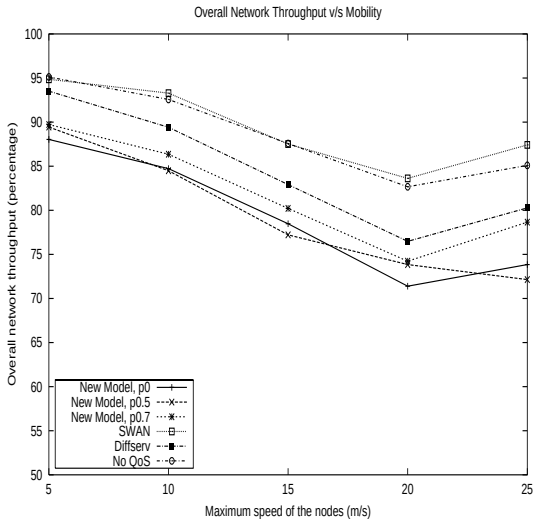

Fig. 8. Overall throughput of the network

B. Link layer based schemes

The *Distributed Coordination Function* (DCF) of 802.11a/b link layer protocols in their current form do not have support for QoS guarantees for real-time traffic. These algorithms, based on *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) protocol, are not capable of providing differential treatment to flows. They aim to distribute the available bandwidth equally and fairly among the competing nodes. Several schemes have been proposed which are a variant of the IEEE 802.11 based protocols. Many of these schemes propose modification of the *inter-frame spacing intervals* (DIFS, SIFS) to differentiate service provided to flows. Other schemes propose the use of slotted TDMA based protocols and suggest ways to reserve slots based on bandwidth requirements. Examples of such schemes are IEEE 802.11e [8] and Black Burst Contention Mechanism [9].

C. Independent schemes

Instead of adding QoS to the routing or link layer, these schemes act as an independent layer, defining separate QoS mechanisms like the Intserv and the DiffServ models. These mechanisms lie between the routing and link layers, however, some of them may rely on feedback from either or both of these layers. We present the details of two such schemes SWAN and INSIGNIA.

1) SWAN (Service Differentiation in Wireless Ad hoc Networks): The SWAN model was developed by the Comet team at Columbia University [1].It is a stateless and fully distributed model that provides *soft QoS assurances* to real-time traffic. The model differentiates traffic into real time UDP traffic and best effort TCP traffic. It uses *admission control* for real-time traffic, rate control of TCP traffic and *ECN congestion control mechanisms* to ensure that real-time packets meet QoS bounds. Each node comprises an admission controller that maintains information about the status of the outgoing link in terms of the available bandwidth and amount of congestion. It does this by promiscuously listening to all packet transmissions within its range. The admission controller located at the source node sends a *probe message* toward the destination when a new real-time flow requires servicing. The probe message returns carrying the value of the bottleneck bandwidth along the path. If this value is greater than the requirements plus a threshold value, the flow is admitted, otherwise it is rejected and marked as best-effort. All TCP flows are considered as best-effort. The best-effort traffic passes through a *rate-controller* that shapes the traffic according to the rate based on the feedback from the MAC layer. The admitted real-time traffic bypasses the rate controller and has a scheduling priority over best-effort traffic. The admitted real-time flows only have soft QoS assurances, so that some of the flows may be dropped or downgraded to best effort if network traffic conditions change due to re-routing of traffic.

2) INSIGNIA: INSIGNIA [10] is an *in-band signaling* system that supports adaptive reservation-based services in ad hoc networks. Thus all the control information is carried within the header of the data packet itself, without the need of a separate control channel. The INSIGNIA framework is also developed by the Comet group at Columbia University.

The signaling system supports a number of protocol commands that drive fast-reservation, fast restoration and end-to-end adaptation mechanism. These commands are carried in-band with the data and encoded using the IP option field in datagrams. This in-band information is $snooped$ as data packets traverse intermediate nodes/routers and used to maintain *soft-state reservations* in support flows/microflows. To establish reservation-based flows between source-destination pairs, source nodes initiate fast reservations by setting the appropriate fields in the INSIGNIA IP option field before forwarding packets. Reservation packets (i.e. data packet with the appropriate IP option set) traverse intermediate nodes, executing admission control modules, allocating resources and establishing soft-state reservation at all intermediate nodes between source-destination pairs. The reservations need to be periodically refreshed by the packets of the flows. In the event of a change in the path resulting from movement of the nodes, the first packet along the new path makes fresh reservations along this path thereby performing a fast restoration. Reservations made along the old path are removed on a timeout. Flows in the network are expected to be adaptive to bandwidth availability. A flow that was allocated a MAX amount of bandwidth initially could be downgraded to MIN amount or even to best-effort in the event of re-routing of a flow or if network conditions change.

The source node continues to send packets with the reservation bit set until the destination node completes the reservation setup phase by informing the source node of the reservation status using a QoS reporting mechanism.

VII. Conclusions and Future Work

This paper addresses the challenging problem of providing quality of service guarantees in mobile ad hoc networks. We present new mechanisms that focus on improving QoS to flows of the highest priority class. The model uses DiffServ as its base and is completely independent of the routing layer and the underlying link layer. Nodes independently monitor the rates of the highest priority flows and signal corrective mechanisms when these rates fall outside of specified local bounds. Triggering conditions for network-wide corrective mechanisms are designed to trade-off rapid reactive response to local QoS violations with control packet overhead. The basic corrective mechanisms may cause under-utilization of network resources by stopping more flows than necessary. Optimization schemes are presented that result in the improvement of overall network utilization. Simulation results are shown under a diverse set of mobility and traffic scenarios. These demonstrate the effectiveness of our mechanisms and indicate performance improvements for the highest priority flow using our new model over DiffServ and SWAN. It should be noted that although the mechanisms demonstrated here consider flows of a single highest priority class, they can be easily extended to monitor flows of multiple classes such that monitoring and corrective actions take place at different thresholds based on traffic characteristics of the class under consideration.

A. Future Directions

This paper reports initial success with novel monitoring and corrective mechanisms for achieving QoS in MANETs. We are currently pursuing the following improvements toward developing a complete QoS solution for mobile ad hoc networks.

1) The model introduces a number of new parameters that impact QoS. An analysis of the role of these parameters and their most effective values has not yet been attempted. These parameters include the monitoring

window size, the triggering threshold, and the setting of p-value in the P-broadcast scheme.

2) As mentioned, our initial monitoring and corrective mechanisms explicitly address only nodes within direct reception range of nodes carrying high priority traffic. We are continuing to study extensions of our mechanisms to handle (i) nodes that are outside direct transmission range but within interference range, and (ii) nodes that interfere with the receiving node.
3) In order to provide tight QoS guarantees we must assure the highest achievable throughput (given network conditions) for the highest priority flow. While our mechanisms demonstrate empirical improvement in throughput we cannot yet claim this guarantee. To further improve throughput we must look more closely at the impact on performance of the overlying routing layer. In our simulations, we only consider AODV as the routing protocol. We also need to study the performance of the model with other reactive protocols such as DSR and proactive protocols such as OLSR.
4) Our re-routing optimization scheme did not produce the desired improvement in network utilization. Our initial investigation indicates that the source nodes may be able to find an alternate route to the destinations, however, many of these new routes may start interfering with the H-Flow at different points of its path. Because of this, other H-Nodes may need to take corrective action. The H-flow may then lose its packet rate more frequently. In the case of mobile scenarios, the H-Flow may need to re-route during the lifetime of the connection. Since the DRI nodes do not forward any control packets within the randomly computed interval, they may drop control packets generated by the nodes for re-routing that flow. If one of the DRI nodes happens to follow on the new route, the H-Flow may not find a new route at all thus degrading its performance. We are continuing to investigate this mechanism.
5) MANET protocols are undergoing rapid change and improvement. While our current mechanisms are independent of protocols at other layers, to achieve a complete QoS solution we would like to be able to automatically adapt our mechanisms to changing PHY and MAC protocols, as well as novel cross-layer solutions.

ACKNOWLEDGEMENT

We thank Harish Sethu for his contributions to our preliminary work in building a DiffServ model for mobile ad hoc networks. We thank John Novatnack for his contributions to building the NS-2 infrastructure used in this work. This research is sponsored in part by a National Science Foundation (NSF) Instrumentation Award under grant CISE-9986105 and in part by a grant from Lockheed Martin Corporation.

REFERENCES

[1] Gahng-Seop Ahn, A. Campbell, A. Veres, and Li-Hsiang Sun, "Supporting service differentiation for real-time and best-effort traffi c in stateless wireless ad hoc networks (swan)," *IEEE Transactions on Mobile Computing*, vol. 1, no. 3, pp. 192–207, July-Sept. 2002.

[2] K. Nichols, V. Jacobson, and L. Zhang, "A two-bit differentiated services architecture for the internet," *RFC 2638, ftp://ftp.rfc-editor.org/in-notes/rfc2638.txt*, July 1999.

[3] H. Arora and H. Sethu, "A simulation study of the feasibility of differentiated services architecture for qos in mobile ad hoc networks," in *Proceedings of Applied Telecommunications Symposium*, San Diego, CA, April 2002.

[4] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on demand distance vector routing protocol," *http://www.ietf.org/rfc/rfc3561.txt*, July 2003.

[5] L. Georgiadis, P. Jacquet, and B. Mans, "Bandwidth reservation in multihop wireless networks: Complexity and mechanisms," *http://www.inria.fr/rrrt/rr-4876.html*.

[6] R. Sivakumar, P. Sinha, and V. Bharghavan, "Cedar: A core-extraction distributed ad hoc routing algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454–1465, August 1999.

[7] C. Perkins and E. Belding-Royer, "Quality of service for ad hoc on-demand distance vector routing," *http://www.ietf.org/html.charters/manet-charter.html*, October 2003.

[8] IEEE WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifi cations," *IEEE 802.11 Standard*, 1999.

[9] J. Sobrinho and A. Krishnakumar, "Quality-of-service in ad hoc carrier sense multiple access wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, August 1999.

[10] S.-B. Lee, G.-S. Ahn, and A. Campbell, "Improving udp and tcp performance in mobile ad hoc networks with insignia," *IEEE Communications Magazine*, vol. 39, no. 6, June 2001.