

A Wireless Overlay Network with QoS capabilities

Eduardo Magaña, Daniel Morató, H. Wilson So, Bill Hodge, Jean Walrand, Pravin Varaiya
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley, CA 94720

email: {emagana, dmorato, so, billh, wlr, varaiya}@eecs.berkeley.edu

Abstract

In this paper, we present a prototype of an overlay network that provides Quality of Service routing for flows in an ad-hoc wireless network. A distributed scheduler offers MAC layer reservation capabilities and a best effort traffic maximization algorithm completes the architecture. We use network measurements and simulation to improve the decisions. The design is tested through a working implementation.

Keywords

Overlay networks, Soft QoS, Wireless ad-hoc networks, QoS routing

I. INTRODUCTION

Wireless networks are becoming a cost effective solution in network deployment. They allow user mobility and provide network connectivity without the costs of wiring the building. This connectivity is provided through access points that link the wireless network to the wired one. However, a special type of these networks provide even more flexibility. They are called wireless ad-hoc networks and they provide user to user connectivity without going through an intermediate access point.

These ad-hoc networks do not need any network infrastructure, neither a wired network nor access points. They are convenient solutions for rapid deployment scenarios where reducing the time it takes to deploy the network is of maximum importance. Examples of scenarios where rapid deployment is needed are natural disasters, emergency situations or military environments. The wireless users become the routing nodes in the network and obtain connectivity to the outer world through wired points, cellular networks or satellite links.

In these scenarios it is very important to have some mechanisms that guarantee Quality of Service (QoS) parameters for some flows. For example, emergency voice calls or high quality video flows showing the

This research was supported by DARPA Contract N66001-00-C-8062

terrain in a rescue mission require prioritized treatment. However, most wireless networks deployed today only offer best effort service. 802.11b is the MAC layer proposed by IEEE for these wireless networks. It is a great success and a lot of commercial equipment has appeared following the standard. However, the wireless channel implemented by IEEE 802.11b is a shared medium without the reservation capabilities needed to guarantee the QoS requirements of important applications. The standard specifies an special access mode called PCF (Point Coordination Function) based on polling for services with QoS requirements. However, PCF has been found to perform badly [1] [2] and most network cards do not implement this access mode but only the basic one called DCF.

The objective of this work is to develop an architecture for providing quality of service in a real 802.11 wireless environment working in the ad-hoc mode. Each wireless device in our prototype acts as a router relaying packets for distant nodes that are not directly reachable. To add more flexibility, some nodes have more than one network interface. Each wireless interface in a node uses a different non-interfering channel to increase the capacity. In this paper we show an actual implementation of the system and reveal the problems that appear in practice.

The capabilities offered by the network are the result of several implemented features:

- Routing is carried out using an overlay network. This overlay is a very convenient solution for an ad-hoc scenario. It offers differentiated treatment of the QoS-sensitive traffic from the best effort services.
- Optimal paths are selected for those flows with specific QoS requirements. This per-flow routing algorithm considers the wireless shared medium and maximizes the amount of total (QoS and best effort) traffic carried by the network.
- A distributed scheduling algorithm was developed to compensate for the lack of MAC layer reservation capabilities in 802.11b. The algorithm is implemented on top of the MAC layer and hence common low cost 802.11b network interfaces can be used without any changes.
- Measurements from the network are used by the QoS routing algorithm to improve the routing decisions.
- A simulation is used to predict and check delay constraints for QoS traffic.
- Best effort traffic is shaped such that it is maximized but yet do not congest critical links.

The rest of the paper is organized as follows: in section II we present the network scenario and the differ-

ent classes of traffic flows considered. Section III gives an overview of the building blocks of the system. Sections IV and V detail the neighbor discovery mechanism and the network measurements respectively. Such information is used by the routing algorithm for choosing path for high priority flows described in section VI. In section VII the whole procedure for QoS flows configuration is described. Section VIII focuses on maximizing the best effort traffic. Section IX presents the distributed scheduling algorithm for adding reservation capabilities on top of the MAC layer. Section X offers an operational example of the entire system in a real scenario followed by the conclusions in section XI.

II. SCENARIO

A. *Traffic types*

Similar to a Differentiated Services scenario [3], we consider two types of traffic: the traffic with QoS requirements (Expedited Forwarding or EF traffic in DiffServ) and the traffic without requirements or best effort traffic (BE traffic).

The kind of applications that we envision for the QoS service are video broadcasting/unicasting, video-conference or audioconference. These applications are sensitive mainly to the available bandwidth in the path but are tolerant to some amount of losses and delay jitter thanks to buffering [4] [5]. Therefore providing soft-QoS [6] for the high priority flows satisfies our application requirements.

The available bandwidth left by the high priority flows is used to satisfy as much of the best effort traffic as possible in the overlay. We do not offer any guarantee to such flows but the routing algorithm for the QoS flows knows about the amount of best effort traffic and finds optimal routes which allow maximum BE traffic carried.

B. *Wireless ad-hoc network*

The scenario under study is a wireless ad-hoc network where nodes run end side applications and also act as routers. This means that the nodes can be source or destination of traffic, and at the same time they can forward traffic for others.

These nodes will have one or more wireless interfaces at different non-interfering frequencies. Nodes that can hear each other in the same frequency form a wireless cell. Those nodes with two or more wireless

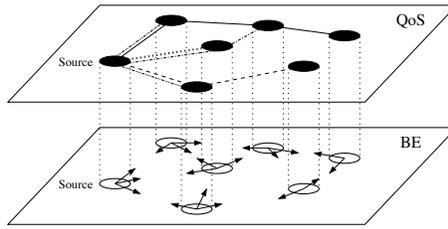


Fig. 1. Overlay network

interfaces can interconnect cells using different frequencies. Within a cell, a node can reach others using multiple hop ad-hoc routing.

To generalize the scenario as much as possible, we also consider *transient nodes* that work only as routers but not as end nodes. They model special repeater nodes deployed to improve network coverage or bridges to connect to other wired and wireless networks.

C. Overlay network

In this work we consider an overlay network that offers traffic classification and different forwarding mechanisms at two different layers (figure 1). The *top* layer routes the data flows with minimum bandwidth and maximum average delay requirements using its own set of routes. The *bottom* layer routes all best effort traffic. A flow is defined as the communication between any two nodes at a particular QoS level. Our architecture is general enough to support multiple QoS levels, but for simplicity of discussion, we will restrict to only two levels of service: high-priority flows with QoS requirements and those without.

The two types of traffic in the overlay network are classified using a different value of the TOS (Type of Service) field in the IP header. Each node redirects the packets to the right layer based on those bits (figure 2).

Flows with QoS requirements are routed on a per-flow basis at the high-priority forwarding layer. The routing tables for QoS traffic contain flow-specific entries that specify the next hop based on (*source node*, *destination node*) pairs. As we will show, this approach offers significant higher routing flexibility than traditional destination-only based routing.

On the contrary, best effort flows are routed in a different layer using traditional destination-based routing entries provided by an ad-hoc routing protocol that chooses shortest paths. Each wireless router runs

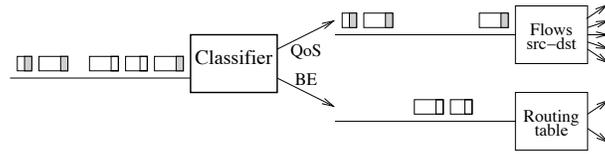


Fig. 2. Overlay routing in each network node

an ad-hoc routing protocol to maintain connectivity. The ad-hoc routing protocol provides an up-to-date routing table to route control messages and best-effort traffic. Our implementation uses DSDV (Destination-Sequenced Distance Vector [7]) which chooses shortest paths based on hop-counts.

In the QoS layer we provide our routing proposal called the WQRSP (Wireless QoS Routing and Scheduling Protocol) algorithm.

D. WQRSP: Wireless QoS Routing and Scheduling Protocol

WQRSP is the fusion of a path selection mechanism, a constraint verification procedure, and a scheduling algorithm.

The main path selection tasks in WQRSP run in a centralized control server. This control server gets information about the topology and measurements from the network (see Section IV and V). Such information is processed and used in the QoS path selection mechanism.

Distributed solutions similar to OSPF [8] floods the link state information to every node in the network. Usually, these wireless ad-hoc networks do not have a large number of nodes (on the order of dozens). Therefore, the centralization of the control server reduces the overhead of control messages compared to a distributed version with total knowledge of the network state in each node.

The main drawback of a centralized solution is if the control server goes down or loses connectivity, the network will stop offering new QoS flow configuration capability. However, this problem can be solved using a leader election algorithm to choose a new control server as soon as the original disappears [9]. When the new control server is elected, it can recover all topology and measurement information from the nodes. It can also learn about existing QoS flows by flooding a request for all routers to report any existing QoS flows.

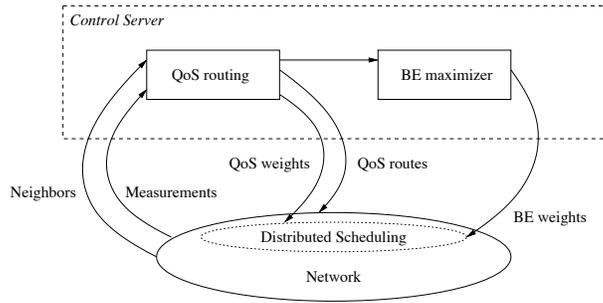


Fig. 3. WQRSP functional blocks

III. WQRSP BLOCKS

The WQRSP system is composed of the functional blocks shown in figure 3, some running in the elected control server, some distributed in all the nodes in the network. The *QoS routing* block is in charge of computing paths for all QoS flows. It outputs the routes for the QoS flows with the associated weights for the distributed scheduler. The *BE maximizer* block computes the optimal parameters needed by the distributed scheduling configuration at each node so the amount of total BE carried by the network is maximized. The outputs are the weights for the BE traffic. Both blocks require information about the network topology and traffic measurements.

The following subsections describe the motivation for each block inside the control server.

A. *QoS routing contribution*

Our primary QoS routing objective presents a delay and bandwidth constrained routing problem. The second objective is to achieve maximum global efficiency measured by the total amount of traffic that the network can handle. If we add a least cost constraint based on the distance to the QoS routing problem, we would get the shortest-feasible paths that will leave the maximum spare capacity.

We propose a QoS routing scheme that simplifies this hard problem with several heuristics that use measurements from the network. These measurements provide detailed knowledge about the demand of BE traffic. With this information, one can reduce the general least cost optimization problem to that of finding feasible (delay and bandwidth constrained) paths that can carry all the best effort demand. Although the solution to the simplified problem does not maximize the spare capacity of the network, it does guarantee that all offered traffic are carried. Consequently, the *real maximum utilization reachable* at that moment is

achieved. In other words, even if a better arrangement of the QoS flows could lower network utilization, it could not increase the goodput of the network as all the present traffic demands are satisfied.

Previous works on routing protocols for ad-hoc networks have focused mainly on keeping the network connected for best effort traffic delivery, with routes based generally on shortest paths [10] [11]. The papers that have studied routing for traffic with QoS requirements for wireless medium have solved media contention by assuming reservation capabilities in the MAC layer [12]. However, these functionalities do not affect the routing decisions, but only provide traffic differentiation in the network or convert the shared medium into dedicated links with the bandwidth divided among them with a contention avoidance protocol such as TDMA [13] [14]. Our QoS proposal focuses on an actual implementation of the widely deployed 802.11b networks based on the standard [15]. To implement QoS routing in a wireless LAN, our model must consider the interaction between different links due to the sharing of medium; the path selection mechanism must take into account which links belong to the same contention domain. To further increase the realism of our model, we also want to include the link quality as an input parameter. In 802.11b, different modulation schemes (and hence different links speeds) are selected based on the measured signal strength or the SNR (Signal to Noise Ratio). We collect such SNR information from the network nodes in real time and incorporate them into our QoS routing algorithm for smarter route selections.

Route computation in the literature follows two different approaches: they try to have precomputed paths for incoming flows with different possible QoS requirements [16] or they compute paths on demand [17]. In both cases, the paths chosen are based on the assumption of a new flow arrival and the objective of finding a feasible path for it. At the same time, they try to maximize the amount of spare capacity or reduce the probability of rejecting new flow requests due to congestion. We improve on existing QoS routing algorithms by adding more information from measurements. In addition, our proposal can re-optimize several existing flows together which allows us to find better solutions than in proposals that consider flows individually.

An important part of this work is the detailed implementation. We are presenting the work of a real implementation of QoS over 802.11 networks. The papers in the literature offer solutions to many problems but usually limit their studies to simulations [12] [18] [19] [14]. In this paper we reveal the problems

that arise when dealing with actual working systems based on standards. As an example, one common assumption in many works on routing with delay constraints is that delay is an additive metric and that measurements of its value in every hop are available [12]. However, measuring the one way delay is not an easy task in real WLANs. End to end one way delay measurements require end hosts to synchronize their clocks. Round trip time delay is not necessarily a good predictor of one way delay due to possible asymmetric paths. Queuing delays are highly variable; access delays also depend heavily on the load in the channel and the variable link quality. Furthermore, the delay experienced by a newly admitted flow depends on the traffic the flow introduces in the network in the near future. This means prediction is needed before the flow can be accepted. The solution we suggest tests the delay in a detailed packet level simulation before the flow is admitted. This approach has the advantage of giving detailed one way delay statistics while being unintrusive. It can also predict the delay under a network load different from the current one by simulating different future scenarios.

B. Contribution in best effort traffic maximization

The QoS routing algorithm takes the first steps towards best effort traffic maximization. The routing procedure always finds a feasible path if one exists. In case several paths exist, it chooses one that does not hamper the BE traffic. This avoids creating unnecessary congestion with QoS flows. After this optimization, we run a second one oriented to avoiding a similar effect but from the BE flows. We do not want a best effort flow to congest a link resulting in reduced global utilization.

We extend the ideas presented in [20] where our proposal for wired networks solves a similar problem. We have extended the formulation to cope with a shared medium. The results translate into weights for output schedulers in each node (for example Weighted Fair Queuing schedulers [21]) that we implement in the distributed scheduler.

IV. TOPOLOGY DISCOVERY

Topology or neighbor discovery is a crucial but deceptively simple component of an ad-hoc routing protocol. A simple neighbor discovery procedure comprises two simple steps. In step 1, an originator node broadcasts a beacon packet containing its node ID. In step 2, the receiver of such beacon records the

originator node as a neighbor. If links are required to be bi-directional, each node must also include a list of neighbors from which it can hear the beacons. Then in step 2, the receiver of a beacon considers the sender a neighbor only if the beacon packet indicates that the sender can also hear the receiver. Finally, to ensure that the neighbor information is updated constantly, each node repeats step 1 and step 2 at regular intervals. A neighbor is removed if it has not been heard for some timeout period. The beacon interval and the timeout is determined by the application requirements and the amount of tolerable transmission overhead.

Our implementation uses broadcast ICMPs. Each node broadcasts one ICMP echo request packet to the local IP subnet broadcast address via each of its wireless interface addresses. Upon receiving an ICMP request, a node replies with an unicast ICMP echo response to the sender of the request. The originator waits 1 second for responses, and then records the IP addresses of all nodes who have responded. All such nodes are considered neighbors. If a neighbor does not respond to 3 consecutive requests, it is assumed to be down and is removed from the neighbor list.

The neighbor discover protocol above has a flaw. If the network is heavily loaded, the beacon packets may be delayed significantly or even lost. To make the neighbor discovery more robust to the jitter and loss of the beacons, one can increase the timeout for the beacons and the tolerance for lost beacons before declaring a neighbor is gone. However, these adjustments, while reducing the probability of falsely declaring that a link is down, also increase the time to detect a genuinely defective link. Our experiments show that when one sender hogs the cell by sending data as fast as possible, the beacons from itself and others tend to be lost frequently. One solution would be to transmit these broadcast ICMPs with higher priority than other data packets.

V. MEASUREMENT

The WQRSP algorithm uses information about the amount of best effort traffic that is carried by the network to help decide the routes for QoS traffic. We use a Linux router testbed. The traffic measurement functions are implemented by specifying a set of firewall rules in the kernel to count packets matching the criteria.

Our approach is to measure the traffic at the ingress router where the traffic first enters the overlay network. One firewall rule is added at the ingress router for each possible egress router interface in the

wireless network for every traffic class. This firewall rule counts both the number of bytes and the number of packets of a particular traffic class (e.g. QoS or BE). This approach is known as the traffic matrix approach. Under this approach, one can calculate the amount of traffic of each class carried by each link using the measurements and routing tables collected at each of the routers for each of the traffic classes.

As mentioned above, a firewall rule is used to record the number of packets and bytes. The iptables facility in the Linux 2.4.x [22] allows the root user to insert firewall filter rules at various points in the kernel. Since we are interested in counting packets that have either originated from the local host or are destined for it, we place a set of firewall rules in the INPUT and OUTPUT chains. When a router first joins an ad-hoc network, it does not know all the remote addresses that require traffic measurements. Therefore, the set of firewall rules must be maintained dynamically. A daemon process is used to monitor the ad-hoc routing tables calculated by WQRSP. Whenever new destinations appear in the routing table, the daemon process automatically inserts a pair of firewall rules in the INPUT and OUTPUT chain.

VI. QoS ROUTING

In this section we describe the algorithm for QoS routing that we implemented and tested in WQRSP.

A. *Effect of sharing into QoS routing*

Ad-hoc networks present several characteristics that make routing decisions different from traditional wired or dedicated links. The first issue is that the wireless medium is shared among all the nodes using the same channel and in range, and yet it has no traffic differentiation capability. This creates several necessities:

- A reservation mechanism is needed. This mechanism must prevent the low priority traffic from hampering the high priority flows.
- Increased reachability is achieved at the expense of bandwidth. In an ad-hoc network, sometimes a node needs to send a packet to a destination, both are using the same wireless channel but are too far apart. In these cases an intermediate node can be used as a relay for the packet. Since the intermediate node sends out the packet using the same channel it came in, a packet uses twice the bandwidth in the channel. An example would be a flow in figure 4 from node 1 to node 2 to node 4. This effect must be considered by the

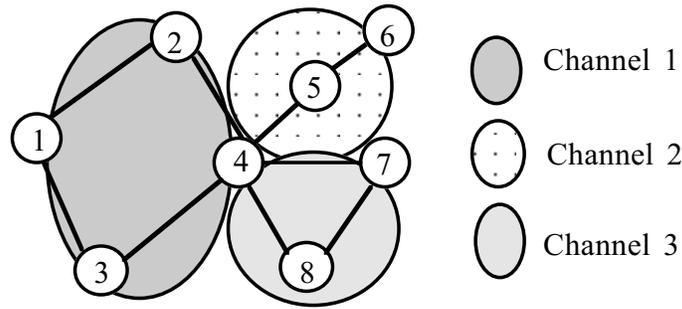


Fig. 4. Link/cell bandwidth constraints

QoS routing algorithm when selecting alternate paths.

- There is not a bandwidth metric per link. A bandwidth constraint is of the type called *link constraint* [23]. This means that each link in the chosen path must have the requested available bandwidth. The simple solution to this constraint routing starts by deleting from the topology graph the links that do not comply with this requirement and then looking for a feasible path based on the other constraints in the new graph, knowing that the bandwidth constraint is assured. However, with a shared medium there is not an *available bandwidth* value for individual links but only for a set of them. The value of the bandwidth for the domain cannot be used as the metric in every link, because if a path uses two links in one channel we could be exceeding the available bandwidth. We show an example using figure 4. If the maximum goodput in each channel is 6Mbps we could route a 6Mbps flow from node 2 to node 4 to node 7, using all the bandwidth in channels 1 and 3. However, a flow from node 1 to node 2 to node 4 to node 7 can only achieve a maximum of 3Mbps as the first two hops are in the same channel.

B. Effects of the forwarding implementation

For the specific forwarding paradigm we evaluate two different proposals. The first one offers maximum flexibility and is the one closest to a flow-based or MPLS-like paradigm. In this proposal, individual paths may be selected for each pair of source and destination nodes (section VI-D). The second one tries to reduce complexity in the routing nodes using traditional destination based routing tables instead of flow based routing. These routing tables will limit the possible paths to trees rooted at each destination node (section VI-E).

We have chosen to solve the routing computation problem for the QoS flows using an Integer Linear

Program (ILP [24]). This approach allows us to incorporate the information from the shared media and optimize the computation for a set of flow requests which provides better results than an optimization that works on flow requests one by one.

C. Definitions

Let N be the set of nodes. $L \subseteq NxN$ is the set of unidirectional links in the network provided by the neighboring discovering mechanism in section IV. Given a node $n \in N$, the set of outgoing links from that node is $Out_n \subseteq L$ and the set of incoming links to the same node is $In_n \subseteq L$. The set $J_n \subseteq N$ is the set of directly connected neighbors of node n , meaning that $m \in J_n \Leftrightarrow (n, m) \in Out_n$. The set $K_n \subseteq N$ is the set of nodes that can reach node n in one hop, which means $m \in K_n \Leftrightarrow (m, n) \in In_n$. For each $(i, j) \in L$ there is a $BW_{i,j} \in \mathcal{R}$ that represents the maximum data rate in that link due to the physical coding scheme. In some cases we will refer to $BW_{i,j}^{(class)}$ as the available bandwidth in the link when the bandwidth being used by flows of type $(class)$ has been subtracted.

In order to add the constraints from the shared medium, we define C as a set partition of L . This C is such as $\forall A \in C$, if $\epsilon_1, \epsilon_2 \in A \Rightarrow$ both links can not be used without interfering one to the other.

B is the set of QoS flow reservation requests. $B = \{(s, d, bw) : s, d \in N, s \neq d, bw \in \mathcal{R}\}$ where s is the source node for the flow, d is the destination and bw is the amount of bandwidth for this flow. For any set B taken as input the following must be true:

$$(s, d, b_1), (s, d, b_2) \in B \Rightarrow b_1 = b_2 \quad (1)$$

Then we can uniquely define $e_{s,d}$ such as $(s, d, e_{s,d}) \in B$. It means that we only consider one possible flow between a specific pair of $(source, destination)$ nodes. We represent the path for this flow by $Path_{s \rightarrow d}^{QoS}$.

D. Network paths based on per-flow routing

In this section we formulate the constraints for the optimization problem that provides independent paths for every flow. We do not restrict ourselves to using destination based routing. As we consider only one possible flow between each pair of nodes in the network it will be enough with routing based on the source and the destination addresses of the packets.

We define $x_{n,j}^{s,d} \in \{0, 1\}$ as an integer variable that takes the value 1 if in node n the packets belonging to the flow $(s, d, e_{s,d}) \in B$ should be forwarded to node j . In this first approach we will disallow the possibility of load balancing and so the flows will use only one path. We impose this in the following equation:

$$\sum_{j \in J_n} x_{n,j}^{s,d} = 1, \quad \forall n \in N, (s, d, e_{s,d}) \in B \quad (2)$$

We also have a flow conservation constraint. We define $f_{n,j}^{s,d}$ as the amount of traffic of the flow $(s, d, e_{s,d})$ using link $(n, j) \in L$. Then equation 3 must hold:

$$\sum_{j \in J_n} f_{n,j}^{s,d} - \sum_{j \in K_n} f_{j,n}^{s,d} = \begin{cases} e_{s,d} & , \quad n = s \\ -e_{s,d} & , \quad n = d \\ 0 & , \quad s \neq n \neq d \end{cases} \quad (3)$$

Now we have to relate variables $x_{n,j}^{s,d}$ with variables $f_{n,j}^{s,d}$:

$$f_{n,j}^{s,d} \leq e_{s,d} x_{n,j}^{s,d} \quad (4)$$

We have to limit the amount of traffic in each link. This is easy because we have information about the maximum traffic allowed in each link due to the modulation coding (equation 5). In section VII we will see that in different optimization steps, we will use different (*class*) superscripts.

$$\sum_{(s,d,e_{s,d}) \in B} f_{n,j}^{s,d} \leq BW_{n,j}^{(class)} \quad (5)$$

But we also have to take into account the sharing among different links using the same channel. This constraint is more complicated because each link could be using a different modulation scheme. Links using a high speed modulation (low error protection) transmit packets using less time than links using lower speed modulations. If all the links were using the same modulation we could limit the total amount of traffic in the collision domain. However, different distances and obstacles make different link speeds very common. In 802.11 the lower speeds mean more error resistant codes and longer packets. Hence, less traffic can be sent using the same amount of the spectrum. If for example in a collision domain we have an 11Mbps link and a 2Mbps link (figure 5), it is different to send 1Mbps using the high speed one versus the low speed

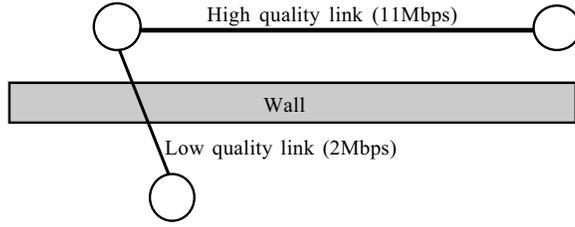


Fig. 5. Different quality links

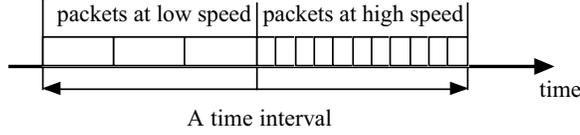


Fig. 6. Different temporal length of transmission

one. A 1Mbps flow in the 2Mbps link leaves half of the available bandwidth. That bandwidth is shared with the 11Mbps link. We make the approximation that we can compare *utilization factors* of the links. In the example that means that because of that 1Mbps traffic the channel is at 50% utilization and the 50% left available could be used with another 1Mbps flow using the 2Mbps link or 5.5Mbps traffic using the 11Mbps link.

We could see this effect in figure 6 where we assume an idealistic slotted transmission. In the first half of the slot interval, there is a 1Mbps transmission in the low speed link. In the second half we can fill the interval with another 1Mbps transmission in the same link or with a 5.5Mbps in the high speed link because packets are shorter in time when sent over the faster link.

What we want to limit is the total utilization of the wireless channel, and we can do this from the local utilization factors of each link ($\rho_{i,j}$), where the modulation dependence has been removed according to equation 6. In this model we take into account the DATA+ACK or RTS+CTS+DATA+ACK mechanism at the MAC reducing the total available bandwidth (section IX).

$$\forall (i, j) \in L, \quad \rho_{i,j} \equiv \frac{\sum_{(s,d,e_{s,d}) \in B} f_{i,j}^{s,d}}{BW_{i,j}} \quad (6)$$

$$\forall A \in C, \quad \sum_{(i,j) \in A} \rho_{i,j} \leq 1 \quad (7)$$

Finally, the objective function for the optimization problem tries to minimize the amount of bandwidth being used (equation 8).

$$\text{Objective} = \min \sum_{(i,j) \in L, (s,d,e_s,d) \in B} f_{i,j}^{s,d} \quad (8)$$

E. Network paths based on destination-based routing

Next, we formulate a very similar optimization problem but in this case, the solution must be destination based routing tables. This forwarding mechanism is simpler than source-and-destination (or per flow) forwarding so it is a faster algorithm at the expense of less flexibility in finding a feasible path (and so higher rejection probability).

Let $x_{n,j}^d \in \{0, 1\}$ be an integer variable that takes the value 1 if in node n the routing entry for destination node d has node j as the next hop. Now, we have reduced all the possible next hops for that destination to only one. The number of possible final destinations does not have to be the whole set of nodes but just those that are destination for some flows. We define the set of destinations in use as D :

$$n \in D \subseteq N \Leftrightarrow \exists (s, n, b) \in B \quad (9)$$

We want to restrict the routing entries with the same destination in the same node to exactly one. We impose this with the constraint in equation 10 (similar to equation 2).

$$\sum_{j \in J_n} x_{n,j}^k = 1 \quad \forall n \in N, k \in D \quad (10)$$

We need the flow conservation constraint in equation 3 and the relationship between $f_{i,j}^{s,d}$ is similar to equation 4 but using $x_{n,j}^d$ instead of $x_{n,j}^{s,d}$. Link and channel bandwidth constraints are again equations 5 and 7 and the objective function remains the same (equation 8).

F. Performance evaluation

We compare the performance of both routing solutions with a test similar to the one used in [25]. It is clear that the option using destination based routing tables is more restrictive than the one using independent

paths. Once a routing entry to a destination is specified in a node all the traffic to that destination that reaches that node will follow the same path.

We run several simulations with a topology of 18 nodes and 27 bidirectional links. There is always more than one path for every destination. The average node degree is 3.22, meaning, on the average, each node has at least three neighbors. The links have different speeds in a two orders of magnitude range with a maximum of 10Mbps. We use a Poisson arrival process and an exponential distribution for the duration of the flows with average 20secs. Source and destination of the flows are chosen randomly from the topology and the bandwidth requested for the flows is uniformly distributed from 1 to 10Kbps. We increase the arrival rate of the flows and compute the number of flows blocked in simulations with 20.000 flow requests.

In figure 7, we plot the percentage of increase in blocked flows from a solution using paths relative to a solution using destination based routing tables. It is different to block a high bandwidth flow than blocking a low bandwidth one, so we compare also the amount of bytes that could have been carried with each solution. As we increase the flow arrival rate there are more flows trying to be on the network at the same time (we keep the same average flow duration). The number of blocked flows increases with both solutions but the situation is worse if we only allow the trees from a destination based routing scheme. As the figure shows the percentage of blocked flows could be a 15% worse using routing tables.

Therefore, when we show the results from the implementation of WQRSP we will use a control server using the formulation for finding independent paths as this is the most flexible solution.

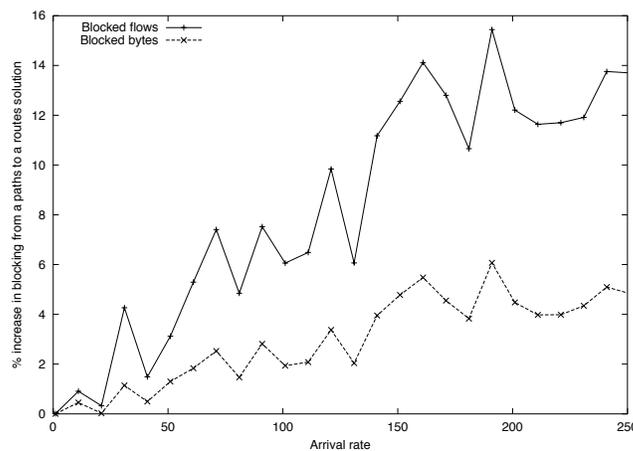


Fig. 7. Increase in blocking

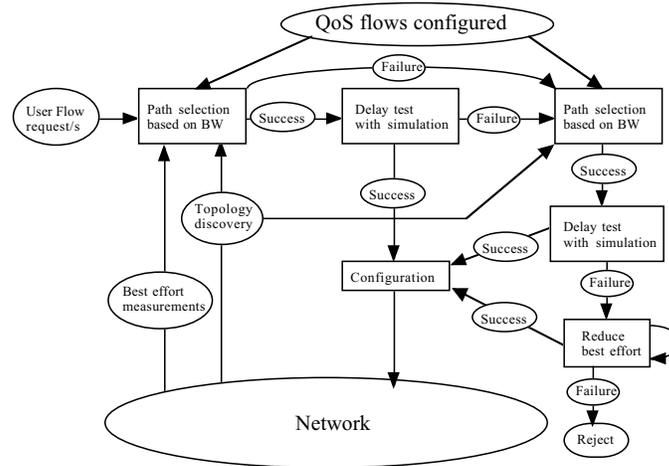


Fig. 8. Whole QoS routing procedure

VII. QoS ROUTING PROCEDURE

In this section we explain the steps taken by the control server to configure a new QoS flow.

The procedure starts with a new incoming flow request from a node to the control server. The request specifies the end nodes for the QoS flow, the minimum bandwidth, and the maximum delay for the flow. Normally the node sending the request will be one of the end nodes, but this is not required. The control server obtains the topology information by periodically polling the network. Topology information include nodes, links, and their modulation speeds. It also includes measurements of the best effort traffic.

The complete procedure can be summarized as follows (figure 8):

1. The server tries to find a suitable path for this new flow taking into account the current QoS and best effort flows in the network.
2. If found, the delay is tested using a simulation. If the delay for every QoS flow is low enough then the configuration commands are sent.
3. If no feasible path is found or the delay is too high for any QoS flow then it will look for another path by ignoring the best effort traffic. It means that there are more possibilities of finding a path but maybe this path will result in less best effort traffic being carried.
4. If a feasible path is obtained then the delay is again tested in a simulation and the configuration is done if no flow exceeds the requested delay.
5. If the delay is too high for any flow then the amount of best effort traffic in the most congested cells

is reduced. This is repeated until the delay is lower than requested delay, in which case the configuration commands are sent. Otherwise, the flow is rejected.

Next, we describe each step in detail.

A. Finding a path that does not interfere with best effort traffic

The best case for the existing QoS flows would be if their routes do not change and their delay does not grow above their requirements after fitting in the new flow. The best case for the best effort traffic would be if even with the new QoS flow added to the network, the network can still carry all the best effort demand. In the first step, the control server looks for a flow configuration that achieves both of the above mentioned objectives.

With the topology and the information from the flows already configured, the control server creates a graph with each link labeled with $BW_{i,j}^{(QoS+BE)}$, that means the available bandwidth once both the bandwidth used by QoS and best effort flows are subtracted. For this graph with the new flow request, we solve the optimization problem described in section VI-D.

The solution of the problem is a path that guarantees the available bandwidth for the actual network situation. But the formulation of the problem does not consider the delay suffered by any flow. Then, if a solution is found, we test the delay as described in section VII-B. If there is no feasible solution, the control server looks for a path in a less restricted (and sub-optimal) scenario. This second possibility is described in section VII-C.

B. Delay test

There are several approaches to verifying the delay that the new flow will suffer. One could rely on analytical models for the systems involved. However, it requires complex models for the video QoS flows (modeling the long range dependence and the codec influence), the generic best effort traffic (probably self-similar), the 802.11 MAC layer, and the queuing policy used. An analytical result would be a hard task. We could use simple empirical models for the delay in wireless networks [26] but their results are not very accurate in multihop scenarios. Another possibility is to start the flow using the selected path and test the delay in the real network. However, this is too slow and too network intrusive. Another possibility is to run

a short simulation to predict the delay the new flow will suffer, and to see how the delay of other QoS flows will change. In this simulation we can use models for the traffic, queuing systems and networks as detailed as the result requires. Fast simulation techniques are available [26]. In addition, such simulation only needs to run long enough to obtain a delay estimation.

In the WQRSP system, the control server predicts the delay for the QoS flows using the GloMoSim simulator [27]. GloMoSim offers 802.11 interfaces, priority queues and traffic sources. The traffic sources for the QoS flows are based on the maximum reservation requested for them (and so it will be a worst case result). The best effort traffic models are based on the actual measurements collected from the networks. This allows an interesting degree of feedback from the network.

From the simulation we extract results about the average delay that each QoS flow would suffer in this configuration. If the delays are below the requirements, the new flow may be admitted. If any delay is too high, several alternatives exist. First, if we have selected a path that avoids interfering with best effort traffic, we can try a path not so protective of the best effort traffic, but a shorter one (section VII-C). If there are no better paths, the network conditions can be improved by reducing the amount of best effort traffic (section VII-D). If the previous alternatives have failed, the flow should be rejected.

C. Finding a path not best-effort-cooperative

Trying to find a path that cooperates with the best effort traffic probably means using a longer path that avoids domains that are congested or highly utilized by the best effort traffic. If this is the case, we can reformulate the optimization problem. This time the labels in the graph will be the available bandwidth $BW_{i,j}^{(QoS)}$, and the solution will optimize the path ignoring the best effort traffic. Now we could be using highly loaded channels but the bandwidth will be guaranteed for the QoS flows by the distributed scheduling mechanism implemented in WQRSP (section IX). However, delay must be tested. Again, for the delay test (section VII-B) all the traffic in the network is modeled into the simulation.

If the results from the simulation are not satisfactory then the control server will do a last attempt changing the network traffic conditions (section VII-D).

D. Reducing best effort traffic

If the control server had to select a short path that does not avoid the best effort traffic and the delay in that solution was still too high, then the best approach to fit the new QoS flow is to reduce the amount of best effort traffic in the selected path. We keep using the path selected in section VII-C because being shorter it uses less wireless channels. The control server will select the channel with higher amount of traffic and reduce the best effort proportion.

This available bandwidth is computed based on the maximum bandwidth in the channel and the amount needed by the QoS flows. If we reduce the maximum bandwidth, as the QoS flows have higher priority, it means reducing the amount of bandwidth for the best effort flows. We keep reducing this maximum shared bandwidth until either the delay requested is reached or there is no bandwidth left for the best effort traffic. In the first case the new flow can be configured; in the second the flow should be rejected.

VIII. BEST EFFORT TRAFFIC MAXIMIZER

The routing tables for the best effort traffic are computed by the DSDV routing protocol. During times when the nodes do not move rapidly, the routes can be assumed to be stable and the network offers static paths for the best effort flows. We want to maximize the amount of best effort traffic that the network can carry with this routing configuration by just adapting parameters in the nodes. As explained in [20], with an implementation of a scheduling discipline like WFQ, the best effort traffic carried can be maximized by carefully selecting the weight for each flow. In that work a Linear Program is formulated that offers the best configuration for the queues that carry best effort traffic. However, the formulation is oriented towards dedicated links, and it does not support shared mediums. In WQRSP, we implement this optimal weight selection procedure but adapt the constraints to the wireless medium.

The objective is to find the amount of best effort traffic between each pair of nodes that maximizes the global carried traffic. We define $b_{s,d}$ as the amount of optimal best effort traffic from node s to node d . This flow uses the route $Path_{s \rightarrow d}^{BE} \subseteq L$. The $b_{s,d}$ values give the weights for the best effort traffic when the QoS flows use $e_{s,d}$ as their weights [20]. We also consider the case where a subset of nodes $N_t \subseteq N$ may be neither source nor destination of best effort flows because they are *transient* nodes. Then the formulation of

the linear problem is contained in the following equations: equation 11 limits the amount of bandwidth in each link, equation 12 introduces the transient nodes, equation 13 limits the total bandwidth in one channel among the possible interfering nodes and equation 14 is the objective of the problem.

$$\sum_{(s,d)/(i,j) \in Path_{s \rightarrow d}^{BE}} b_{s,d} \leq BW_{i,j}^{(QoS)} \quad (11)$$

$$s \in N_t, d \in N \Rightarrow b_{s,d} = b_{d,s} = 0 \quad (12)$$

$$\forall A \in C, \sum_{(i,j) \in A} \rho_{i,j}^* \leq 1 \quad (13)$$

$$Objective = max \left\{ \sum_{s,d \in N/s \neq d} b_{s,d} \right\} \quad (14)$$

$\rho_{i,j}^*$ is defined similar to equation 6 as the utilization of link (i, j) but now we consider the link carrying both QoS and best effort traffic (equation 15).

$$\rho_{i,j}^* = \left(\sum_{(s,d)/(i,j) \in Path_{s \rightarrow d}^{BE}} b_{s,d} + \sum_{(s,d)/(i,j) \in Path_{s \rightarrow d}^{QoS}} e_{s,d} \right) BW_{i,j}^{-1} \quad (15)$$

IX. DISTRIBUTED SCHEDULING

The MAC layer in 802.11b offers a shared medium and is based on a CSMA/CA and RTS/CTS/DATA/ACK mechanism. With this shared medium all the nodes compete for the same bandwidth. One cannot guarantee a minimum bandwidth for an QoS flow because we cannot control the rest of flows (QoS or BE). 802.11b provides a simple mechanism to guarantee resources for certain flows via PCF but its performance has been demonstrated to be too poor [2]. There are many proposals to incorporate bandwidth guarantee to 802.11b using schemes similar to techniques used in wired Ethernet like a token protocol [28]. Besides, there are other proposals designed specifically for 802.11b changing some part of the MAC protocol by adding a Multiclass Priority Fair Queue in the base station [29], changing the IFS (Interframe Space) time and CW (Contention Windows) size per class [30] and interchanging Virtual Clock data about backoff times between neighbor nodes [31]. IEEE's 802.11e is a committee working to include QoS characteristics within 802.11 wireless LANs [32] [33] and they change the MAC layer incorporating a EDCAF (Enhanced Distributed

Coordination Function) which is much more effective. However, for our implementation we need to use existent hardware. Therefore we provide a solution compatible with the 802.11 MAC layer. In [34] we have a similar proposal based only on priorities. We will complete this approach with limiters per interface to avoid congestion in the wireless cell.

There is a similar problem with guaranteeing the delay. We can include a simulator in this part as in the control loop (section VII) to obtain accurate delay measurements. This simulator must report the maximum amount of traffic per node to achieve certain end to end delay. However it would require us to run the simulation each time that traffic flow changes. Because of the burstiness of BE traffic this solution is not acceptable because of the high computational cost. However, one can use a simple model of the 802.11b MAC protocol as proposed in [26] that provides delay values close enough to reality [35] for this situation of one-hop cells. This model says that with a network utilization no bigger than 60%, the delays are kept low. The high delays appear when the network utilization exceed 60%.

To guarantee bandwidth and delay for QoS flows, we need a scheduling in the outgoing interface of the nodes. One can shape the traffic at the source, but it requires the source to know the state of the entire network. In our proposal, it is enough to know the state of nearby nodes inside the same cell. The existence of hidden nodes can be a potential, but for our proposal we have worked with the assumption of total connectivity between the nodes inside each cell as in [34] [30].

We propose to guarantee that maximum delay and minimum bandwidth by limiting the flows in each cell under 6 Mbps and giving QoS traffic priority. The rate limiting is achieved with a distributed scheduling algorithm working in each cell. This scheme consists of a traffic shaper in each node that limits total generated traffic in the cell and shares the available bandwidth using a WFQ scheduler with the weights from the best effort traffic maximizer (section VIII). All the nodes inside the cell exchange the amount of QoS/BE traffic they want to transmit in order to share the bandwidth fairly. However, because of the difficulty of combining a WFQ scheduler, the limiter and a priority scheduling in this order inside the Linux kernel (traffic control [36]), we have made an approximation shown in figure 9. In this scheme the traffic shaping has two parts. First, the QoS traffic is marked as either high priority (HP) or BE with low priority (LP). In the output queue, the QoS traffic is always going to be serviced first using a simple priority

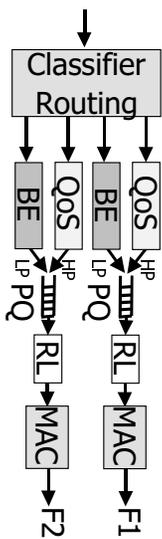


Fig. 9. Scheduling per interface in each node

scheduler (Priority Queuing, PQ). Second, the resulting traffic is limited (Rate Limiter, RL) to a certain value depending on the volume of QoS traffic through that node, the weights for BE traffic given by the best effort traffic maximizer (to optimize network utilization by BE traffic), and the amount of QoS traffic and BE weights in other nodes. Under this proposal, the total traffic in the cell never exceeds 6 Mbps, keeping a low delay for each flow. This scheme uses different queues per class of service together with a priority scheduler, similar to the proposed IEEE 802.11e, but working over 802.11 MAC layer without modifying the MAC layer as IEEE 802.11e does [32].

In the distributed scheduling, all nodes broadcast periodically the amount of QoS/BE they would want to transmit in the next interval. This transmissions are synchronized in such a way that if the nodes listen to a new broadcast information from some node, they transmit at that moment too with a short random delay to avoid collisions. The distributed scheduling guarantees that the network utilization never exceeds 6 Mbps and assures the transmission of QoS traffic. It also tries to use all remaining bandwidth for BE traffic. The limiter per node is chosen in such a way that it is no smaller than the amount of QoS traffic of that node. If there is remaining bandwidth, it is shared among the nodes in a proportional way depending on the BE weights calculated by the best effort maximizer block.

X. DEMONSTRATION OF CAPABILITIES

We provide a real implementation of the whole set of capabilities. It has been implemented in a real network for a simple demonstration as part of the DARPA NMS project [37]. This solution is based on notebook computers running GNU/Linux. The topology is presented in figure 10. It contains 6 laptops in 3 different channels and 3 laptops with 2 interfaces to connect the different cells. We have tested it under situations where high quality video transmissions compete with greedy best effort flows. We have shown that in this overlay network, optimal paths are selected to allow the video to play flawlessly, and that all

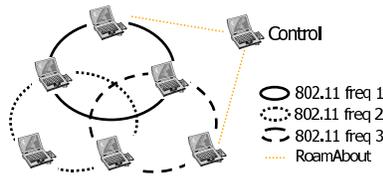


Fig. 10. Scenario implemented

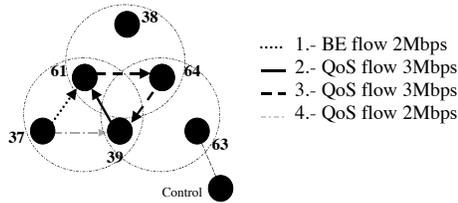


Fig. 11. Flows established in the scenario

available bandwidth remaining is used by the best effort traffic.

Figure 11 presents an example of configuration of flows to illustrate the procedure. First, one BE flow of 2Mbps is established between nodes 37 and 61, using DSDV routes. Then, we want to establish a 3Mbps video flow (QoS) between nodes 39 and 61. As the source, node 39 makes the request for this flow to the control server and WQRSP decides the path 39-61. Now, cell 37-39-61 is almost congested. After that, node 61 request a new 3Mbps video flow with node 39 as destination, and WQRSP decides the path 61-64-39 to reduce congestion in the cell composed by nodes 37, 39 and 61. Finally we put a 2Mbps QoS flow between nodes 37 and 39, and then the first BE flow is automatically limited to 1Mbps. With that configuration it is possible that those 3 flows coexist in that cell, keeping the guaranteed bandwidth and delay for the two QoS flows.

XI. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a simple, yet self-contained architecture which provide soft QoS guarantees to applications with bandwidth and delay constraints. We have shown a real implementation of QoS routing over an overlay ad-hoc network using only off the shelf 802.11b network interfaces and laptop computers.

Through the experience of implementing the algorithms and protocols, we have studied the problem of incorporating shared medium constraints into routing. We have proposed solutions that enable us to provide

QoS in rapid deployment scenarios. We have demonstrated that our implementation achieves our original goals of supporting high priority flows while accommodating as much best effort traffic as possible using an overlay network.

As future work, we are working in different aspects like including PDAs (Personal Digital Assistants) as wireless nodes, including a distributed protocol for topology discovery and measurement, and applying a hierarchical scheme instead of a centralized control server. This hierarchy will provide better scalability to large wireless ad-hoc environments.

XII. ACKNOWLEDGMENT

This project is a joint work with Eric Chi, Luan Nguyen, Duc Pham, and other members of the Smart Networks Project at UC Berkeley. For brevity reasons, the list of authors has not included their names.

REFERENCES

- [1] M. Barry, A.T. Campbell, and A. Veres. Distributed control algorithms for service differentiation in wireless packet networks. In *IEEE INFOCOM 2001*, 2001.
- [2] A. Kospel, J.P. Ebert, and A. Wolisz. A performance comparison of point and distributed coordination function of an IEEE802.11 WLAN in the presence of Real-Time requirements. In *7th International Workshop on Mobile Multimedia Communications (MoMuC2000)*, Waseda, Tokio, Japa, Oct 2000.
- [3] J. Liebeherr and N. Christin. Rate allocation and buffer management for Differentiated Services. *Computer Networks, Special Issue on the New Internet Architecture*, August 2002.
- [4] J., B. Li, and Y. Zhang. An End-to-End adaptation protocol for layered video multicast using optimal rate allocation. *IEEE Transactions on Multimedia*, 2002.
- [5] D. Wu, Y.T. Hou, and Y.Q. Zhang. Scalable video coding and transport over broadband wireless networks. *Proceedings of the IEEE*, 89(1):6–20, January 2001.
- [6] D. Reiniger, R. Izmailov, B. Rajagopalan, M. Ott, and D. Raychaudhuri. Soft QoS control in the WATMnet broadband wireless system. *IEEE Personal Communications*, pages 34–43, February 1999.
- [7] C.E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers. In *ACM SIGCOMM'94*, pages 234–244, London, UK, Sep 1994.
- [8] J. Moy. OSPF version 2. RFC 2328, IETF, April 1998.
- [9] N. Malpani, J.L. Welch, and N. Vaidya. Leader election algorithms for mobile ad hoc networks. In *Fourth International Workshop on Discrete Algorithms and Methods for Mobile*, Boston, MA, 2000.
- [10] X. Hong, K. Xu, and M. Gerla. Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, 16(4):11–21, July/August 2002.
- [11] E.M. Royer and C.-K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, pages 46–55, April 1999.
- [12] S. Chen and K. Nahrstedt. Distributed Quality-of-Service Routing in Ad Hoc Networks. *IEEE Journal on Selected Areas in Communications*, 17(8), 1999.
- [13] J. Tsai and M. Gerla. Multiclustor, mobile, multimedia radio network. *ACM-Baltzer J. Wireless Networks*, 1(3):255–265, 1995.
- [14] C.R. Lin and M. Gerla. Adaptive clustering for mobile wireless networks. *IEEE Journal on Selected Areas of Communications*, 15(7):1265–1275, 1997.
- [15] IEEE Computer Society LAN MAN Standards Committee. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11-1997*. The Institute of Electrical and Electronics Engineers, New York, 1997.
- [16] R.A. Guerin, A. Orda, and D. Williams. QoSrouting mechanisms and OSPF extensions. In *Proceedings of Globecom 1997*, volume 3, pages 1903–1908, 1997.
- [17] G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi. On Reducing the Processing Cost of On-Demand QoS Path Computation. In *International Conference on Networking Protocols (ICNP)*, Austin, TX, October 1998.
- [18] R. Ramanathan and M. Steenstrup. Hierarchically-organized, multihop mobile wireless networks for quality-of-service supprt. *ACM/Baltzer Mobile Networks and Applications*, 1998.
- [19] R. Leung, J. Liu, E. Poon, C. Chan, and B. Li. MP-DSR: A QoS-aware mutli-path dynamic source routing protocol for wireless ad-hoc networks. In *Proceedings of the 26th IEEE Annual Conference on Local Computer Networks*, pages 132–141, 2001.
- [20] E. Magaña, D. Morató, and P. Varaiya. Tuning the weights in WFQ schedulers for the maximization of carried best effort traffic. In *To appear in the Proceedings of ICT 2003*, 2003.
- [21] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Journal of Internetworking Research and Experience*, pages 3–26, October 1990.

- [22] Iptables/netfilter. <http://www.netfilter.org/>.
- [23] S. Chen and K. Nahrstedt. A overview of Quality of Service routing for next-generation high-speed networks: problems and solutions. *IEEE Network*, pages 64–79, November/December 1998.
- [24] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. Willey-Interscience in Discrete Mathematics and Optimization. John Wiley & Sons, 1988.
- [25] Q. Ma and P. Steenkiste. On path selection for traffic with bandwidth guarantees. In *Proceedings of IEEE International Conference on Network Protocols*, 1997.
- [26] J. Liu, D.M. Nicol, L. F. Perrone, and M. Liljenstam. Towards high performance modeling of the 802.11 wireless protocol. In *2001 Winter Simulation Conference (WSC 2001)*, Arlington, VA, Dec 2001.
- [27] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: a Library for the Parallel Simulation of Large-scale Wireless Networks. In *20th Workshop on Parallel and Distributed Simulation*, pages 154–161, Banff, Alberta, Canada, 1998.
- [28] S. Sharma, K. Gopalan, N. Zhu, G. Peng, P. De, and T. Chiueh. Implementation experiences of bandwidth guarantee on a wireless LAN. In *SPIE Multimedia Computing and Networking (MMCN2002)*, San Jose, CA, Jan 2002.
- [29] J.R. Moorman, J. Lockwood, and S. Kang. Wireless quality of service using multiclass priority fair queuing. *IEEE Journal on Selected Areas in Communications*, August 2002.
- [30] J. Deng and R.S. Chang. A Priority Scheme for IEEE 802.11 DCF Access Method. *IEICE Transactions on Communications*, E82-B(1), January 1999.
- [31] H. Luo, P. Medvedev, J. Cheng, and S. Lu. A Self-Coordinating Approach to Distributed Fair Queueing in Ad Hoc Wireless Networks. In *IEEE INFOCOM*, volume 3, pages 1370–1379, Anchorage, Alaska, April 2001.
- [32] S. Mangold, S. Choi, P. May, O. Klein, G. Hiertz, and L. Stibor. IEEE 802.11e Wireless LAN for Quality of Service. In *European Wireless'2002*, volume February, Florence, Italy, 2002.
- [33] A. Lindgren, A. Almquist, and O. Schelen. Evaluation of Quality of Service schemes for IEEE 802.11 Wireless LANs. In *26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*, 2001.
- [34] A. Koepsel and A. Wolisz. Voice transmission in an IEEE 802.11 WLAN Based Access Network. In *WoWMoM 2001*, pages 24–33, Rom, Italy, July 2001.
- [35] Y. C. Tay and K. C. Chua. A capacity analysis for the IEEE 802.11 MAC protocol. *Wireless Networks*, Kluwer Academic Publishers Hingham, MA, USA, 7(2):159–171, March/April 2001.
- [36] B. Hubert, G. Maxwell, R. Mook, M. Oosterhout, P.B. Schroeder, and J. Spaans. Linux Advanced Routing and Traffic Control. <http://www.lartc.org/>, 200-2002.
- [37] DARPA-IPTO project: Network Modeling and Simulation. <http://www.darpa.mil/ipto/research/nms/index.html>, 2000-2002.