Fair Scheduling with QoS Support in Ad Hoc Networks

Hsi-Lu Chao, Jia-Chun Kuo and Wanjiun Liao
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan
Email: wjliao@cc.ee.ntu.edu.tw

Abstract

This paper studies fair scheduling with Quality of Service (QoS) support in ad hoc networks. The goal is to satisfy the minimum bandwidth requirements of guaranteed flows and to provide fair share of residual bandwidth among all flows. We propose a new cluster-based mechanism called two-tier slot allocation protocol (2TSAP), which improves system throughput and ensures per-cluster fairness while satisfying the requirements of guaranteed flows. With 2TSAP, the network is logically partitioned into clusters, each with a scheduler. The scheduler assigns time slots to mobiles in the affiliated clusters based on the first tier algorithm. The mobiles scheduled to send at the next time slot then in turn assign the time slot to flows determined by the second tier algorithm. We compare the performance of 2TSAP with Q_EMLM-FQ, a virtual-time based fair scheduling mechanism with QoS support. The simulation results show that 2TSAP meets the minimum requirements of guaranteed flows, shares the residual bandwidth fairly among all flows, and improves overall system throughput.

Keywords: QoS, fair scheduling, ad hoc networks

1. Introduction

An ad hoc network is a self-organizing wireless network comprised only of mobile nodes interconnected by a multihop path and without the support of any pre-existing wired infrastructure. To date, research in ad hoc networks has mostly focused on the provision of best-effort service [1,2]. With the increasing popularity of multimedia traffic, one must address how to support QoS in ad hoc networks. Providing QoS in ad hoc networks is challenging, given their peculiar characteristics compared to conventional wired or single hop wireless networks. In wired or cellular networks, only routers or base stations are involved in making scheduling decisions. In ad hoc networks, all nodes may be involved in making decisions. The major problem of a fully distributed scheduling mechanism is that it may cause serious collisions and significantly degrade network throughput. Besides, the spatial channel reuse is another characteristic of wireless ad hoc networks. Due to the broadcast behavior, each node within the sender's transmission range will receive data packets. That is, simultaneous transmissions of the nodes far enough to be out of the sender's transmission range are permitted.

According to the decision metrics, scheduling disciplines in wireless ad hoc networks can be classified into two categories: *virtual-time based* [1,2] and *credit based* [3]. [1,2] work similarly. Using [2] as an example, each newly arriving packet is locally assigned two timestamps: a start tag and a finish tag [4]. Either timestamp can be chosen as the service tag. [1] and [2] choose the former and the latter to be the service tag, respectively. The packet with the smallest service tag will be sent first. Besides, [2] implement spatial channel reuse by means of the backoff mechanism [5]. Each packet of a flow is associated with a backoff value. This backoff value is equal to the number of flows contending for a channel but with smaller service tags. Once this backoff value reaches zero, the packet is transmitted. These approaches, however, require to sorting all packets in the queue in the increasing order of their service tags, and the virtual clock cannot be reset unless the queue becomes empty [3]. In addition, they tend to distribute resource equally among all flows, which may not support QoS well. In [3], scheduling is based on the credit value, a non-zero positive integer. The unused credit will be cumulated for future use. The one with the highest credit value is allowed to transmit packets first. This approach, however, only consider best effort flows. Directly applying [3] to support QoS for guaranteed flows would starve best-effort flows.

0742-1303/02 $17.00 © 2002 **IEEE**

In this paper, we consider an environment with a mix of guaranteed and best effort flows. Our goal is to guarantee the minimum bandwidth requirements of guaranteed flows, while ensuring fair share of residual bandwidth among all flows (i.e., both best effort and guaranteed)[1]. We propose a mechanism called Two-Tier Slot Allocation Protocol (2TSAP), which uses the concept of "credit" as in [3], instead of "service tag" in [1,2], to make scheduling decisions.

The rest of this paper is organized as follows. Section 2 describes the proposed two-tier slot allocation protocol. Section 3 shows the simulation results. Finally, the conclusion is drawn in Section 4.

2. Two-Tier Slot Allocation Protocol (2TSAP)

This section describes the proposed Two-Tier Slot Allocation Protocol (2TSAP), which provides QoS support for guaranteed flows and ensures fairness in resource sharing among all flows. The detailed mechanism is described as follows.

2.1 Assumptions

1. Wireless media may exhibit time-varying errors due to co-channel interference, fading, link errors, and collisions. In this paper, we only consider the errors caused by collisions when the two mechanisms are compared.
2. We assume a TDMA-based system on a single channel shared by all hosts. To avoid collisions, CDMA can be overlaid on the top of the TDMA system. We assume a code assignment algorithm is running in the lower layer of our system. Each TDMA frame contains a fixed number of time slots. The network is synchronized on a frame and slot basis. A frame is comprised of a control phase and a data phase. The control phase is used for mobile nodes and schedulers to send time-slot requests and replies. The data phase is used for designated nodes to transmit packets [6].
3. We consider packet-switched multihop wireless networks, but do not consider host mobility as in existing work [1,2].

2.2 Data Structure

(1) Node Allocation Table (NAT): maintained by the schedule; each entry of the NAT is for a (node, service type) pair and has six fields: a node ID, a service type, a Resv, a Credit, a Usage, and an Excess.

(2) Flow Allocation Table (FAT): maintained by the node; each entry of the FAT is for a flow and has seven fields: a flow ID, a scheduler ID, a service type, a Resv, a Credit, a Usage, and an Excess.

2.3 Scheduling Parameters

There are four parameters defined in the two-tier scheduling mechanism: *Resv, Credit*, *Usage*, and *Excess*.

(1) *Resv*: for the guarantee service, *Resv* represents the minimum time slot requirement to meet the desired QoS level, with a value between 0 and 1; for the best effort service, the value is always zero.

(2) *Credit*: the cumulative time slots reserved for a guaranteed flow, or the remaining slot quota for a best effort flow per iteration[2]. For guaranteed service type, the value of an increase at each time slot is equal to a *Resv* value. For the best effort service type, the initial *Credit* value is equal to the total number of best effort flows currently scheduled by the node, and is decreased by one after a slot is assigned. Note that for a best effort flow, the *Credit* field is only valid in the NAT and is always set to zero in the FAT.

(3) *Usage*: cumulative time slots assigned to a request. The *Usage* value could be a zero or any positive integer.

(4) *Excess*: this value is used to determine to whom the next time slot is assigned. The next slot is assigned to the request with the smallest *Excess* value. For guaranteed-service entries, the *Excess* values are equal to "Usage minus Credit," and are updated at each time slot. For best effort-service entries, the initial *Excess* values are set to zero, and are increment at each "update." The *Excess* value only is updated when the corresponding *Credit* value is count down to zero. Once the *Excess* value has been updated, the *Credit* value is reset to a *Resv* value, and a new iteration starts.

2.4 Cluster-based Mechanism

2TSAP is a cluster-based mechanism. Each cluster has a scheduler (or master). Clusters may be overlapped, each with a designated code. The cluster can be formed as

[1] Note that our mechanism can work with different scenarios of residual bandwidth allocation, including fair scheduling for best effort flows only, for a mix of best effort and guaranteed flows, or for a mix of flows but with an upper limit on the resource share for guaranteed flows. In this paper, we just demonstrate the second case, i.e., for a mix of flows without an upper bound on each flow.

[2] Each iteration indicates a cycle in which the slot quota is counted down from the original value to zero.

follows. Each mobile node periodically broadcasts beacons. Based on the received beacons, mobile nodes learn their neighbors, and related information, such as node ID, node stability. According to the selected criterion [7,8], each cluster can determine its scheduler. To receive the service, each node must join a cluster and register with the scheduler. Each scheduler periodically advertises itself as the scheduler to its cluster, from which newly arriving mobile nodes learn where to register.

2.5 Path Selection

A flow sender must first determine a flow path before a transmission, using an ad hoc routing mechanism such as [9,10,11]. If the flow needs guaranteed service, the determined path must meet the desired QoS level of the flow. The sender first broadcasts a Route REQuest (RREQ) message with a flow ID, a destination ID, a Resv, and a service type to its neighbors. The value of the *Resv* depends on the service type of the RREQ, i.e., for guaranteed flow, the Resv value is set between 0 and 1, and for best effort, it is always zero.

On receiving a new request, say request *i*, to relay the flow's packets, the node will verify if the summation of the reservation levels (i.e., Resvs) of all flows scheduled by the node, including the newly arriving one, is less than or equal to the target link utilization. If it is the case, the node further sends a resource Allocation ReQuest (ARQ) to its scheduler to ensure that the verification holds at the scheduler as well (i.e., the summation of Resvs of all flows scheduled by the scheduler including the new ARQ is less than or equal to the target link utilization level.) If either verification fails, the request is denied; otherwise, the request is tentatively accepted. The scheduler creates a temporary entry for the node in the NAT, and responds to the node with an acceptance. The node then creates a temporary entry for the flow in the FAT. Once the request is accepted, the node rebroadcasts the RREQ to its neighbors. Duplicated RREQs are dropped. When the destination receives this RREQ, it sends back a Route REPly (RREP) message along the reverse path. The node changes the respective entry from temporary to regular upon receipt of the RREP, and informs its scheduler to do likewise. Those who do not receive the corresponding RREP after a period of time will clear the temporary entries in NAT and FAT, respectively.

2.6 Two-Tier Slot Allocation Algorithm

The slot allocation is based on a two-tier mechanism. The scheduler assigns time slots based on the first tier slot allocation mechanism. The node to whom the time slot is assigned then in turn assigns the time slot to a flow based on the second tier slot allocation mechanism. Both the scheduler and the node allocate slots only to regular entries of their respective tables. The two tier slot allocation mechanisms are repeated as long as there is a flow relayed through any node.

Suppose that there are *m* and *n* entries in both NAT and FAT, respectively. These *m* entries in NAT can be divided into two sets: S_g and S_b. S_g and S_b indicate guaranteed and best effort service type, respectively. The first tier slot allocation mechanism works as follows.

(1) The scheduler assigns the next time slot to the node with the smallest Excess value in the NAT at decision time.

The scheduler then updates the values of Credit, Usage, and Excess in all entries of the NAT, as follows.

(2) Increment the Usage value of the entry of the node scheduled to send at the next time slot, say node A, and leave all the others intact, i.e., Usage (A) ← Usage (A) +1, where Excess (A) = min {Excess (x)| x=1,2,…m}.

(3) Increase an amount of Resv to Credit for each guaranteed-service entries in the NAT, i.e., Credit (x) ← Credit (x) + Resv (x), $x \in S_g$. Besides, if next time slot is assigned to node A with best effort service type, decrement the Credit value of node A, i.e., Credit (A) ← Credit (A) – 1.

(4) Update the Excess values of all guaranteed-service entries in the NAT, i.e., Excess (x) ← Usage (x) – Credit (x), $x \in S_g$; in addition, if the next time slot is assigned to the best effort service type of node A, and the Credit value equals to zero, the Excess value is increased by one and the Credit value is reset to the Num value, i.e., if Credit (x) = 0, Excess (x) ← Excess (x) + 1, and Credit (x) ← Num (x). Otherwise, the Excess value stays unchanged.

The second tier slot allocation algorithm works as follows.

(1) The node cannot schedule any flow packet unless a time slot is assigned to it. Once a time slot is assigned, the node assigns the time slot to the flow which satisfies the following two conditions:(a) the flow has the smallest Excess value, and (b) the flow's service type matches the service type of the node granted by the scheduler.

The node then updates the values of Credit, Usage, and Excess in all entries of the FAT, as follows.

(2) Increment the Usage value of the entry of the flow scheduled to send at the next time slot, say flow F, and leave the others intact. Usage (F) = Usage (F) + 1, where Excess (F) = min {Excess (y)| y=1,2,…n}.

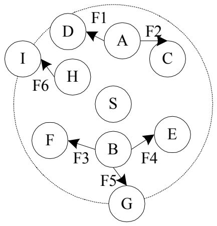
(3) Credit (y) = Credit (y) + Resv (y), y=$1,2,…n$.

(4) Update the Excess values of all entries in the FAT, i.e., Excess (y) = Usage (y) – Credit (y), y=$1,2,…n$.

2.7 Example

Figs. 1 and 2 show how 2TSAP works. Fig. 1(a) shows the node topology and Fig. 1(b), the flow information. All nodes are located within the transmission range of node S, and S is the cluster scheduler for nodes A to I. Node A sends two ARQs to node S: one for best effort service with a reservation level of 0, and the other for guaranteed service with a reservation level of 0.2. Node B sends two ARQs to node S: again, one for best effort service, and the other for guaranteed service with a reservation level of 0.3. Node H sends node S an ARQ with the best-effort service type. Scheduler S records the states of nodes A, B, and H in its NAT.

Fig. 2(a) shows the slot allocation by node S to nodes A, B, and H, and the updates on the NAT in a 6-slot time span. K(b) and K(r) denote the entries of the NAT for node K with the best effort and guaranteed service types, respectively. Initially (i.e., at time 0), the Resv values of the entries are 0, -0.2, 0, -0.3, and 0, respectively. Thus based on the first-tier slot allocation mechanism, the first time slot is assigned to the node with the least Excess value, i.e., node B with guaranteed service type. Fig. 2 (b) shows the slot allocation by node A to its flows, again in a 6-slot time span. We can see that node A gets scheduled at time slots 2, 3, and 6 when it schedules flows passing through using the second tier slot allocation mechanism.



Flow	Source	Dest.	Resv.	Scheduler
F1(b)	A	D	0	S
F2(r)	A	C	0.2	S
F3(b)	B	F	0	S
F4(b)	B	E	0	S
F5(r)	B	G	0.3	S
F6(b)	H	I	0	S

(a) Network topology (b) Flow information

Figure 1. Network topology and flow information

Time	0			1			2			3			4			5		
Node	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E
A(b)	1	0	+0.0	1	0	+0.0	1	1	+1.0	1	1	+1.0	1	1	+1.0	1	1	+1.0
A(r)	0.2	0	-0.2	0.4	1	+0.6	0.6	1	+0.4	0.8	1	+0.2	1.0	1	+0.0	1.2	2	+0.8
B(b)	2	0	+0.0	2	0	+0.0	2	0	+0.0	1	1	+0.0	1	1	+0.0	1	1	+0.0
B(r)	0.3	1	+0.7	0.6	1	+0.4	0.9	1	+0.1	1.2	1	-0.2	1.5	2	+0.5	1.8	2	+0.2
H(b)	1	0	+0.0	1	0	+0.0	1	0	+0.0	1	0	+0.0	1	0	+0.0	1	0	+0.0

(a) NAT updates at scheduler S

Time	0			1			2			3			4			5		
Flow	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E	C	U	E
F1(b)	0.0	0	+0.0	0.0	0	+0.0	0.0	1	+1.0	0.0	1	+1.0	0.0	1	+1.0	0.0	1	+1.0
F2(r)	0.2	0	-0.2	0.4	1	+0.6	0.6	1	+0.4	0.8	1	+0.2	1.0	1	+0.0	1.2	2	+0.8

(b) FAT updates at node A

Figure 2. 2TSAP operation

3. Simulation

This section evaluates the performance of 2TSAP by simulation. There are 20 mobile nodes randomly distributed in a 670-by-670 area. We randomly selected nodes, some as flow senders and some as flow receivers. Each flow may be best effort or guaranteed. For best effort flows, their Resv values are zeros, and for guaranteed flows, their values are randomly generated between zero and one. Each packet is assumed to occupy one time slot, and has fixed packet length.

We compare the result with [2], a virtual-time based fair scheduling protocol. [2], however, does not consider the QoS aspect while ensuring fairness. Thus we modify the definition of weight in [2] as follows to enable minimum bandwidth guarantee. We call the modified version as Q_EMLM-FQ in the rest of the paper. For each flow f passing through node N, the flow weight w_f is defined as

$$w_f = R_f + \frac{1 - \sum_{i \in \{bgflows\}} Ri}{N_f}$$, where R_f and N_f are the Resv value of flow f and the total number of flows passing through node N., respectively; *bgflows* indicates the set of backlogged flows with guaranteed service type.

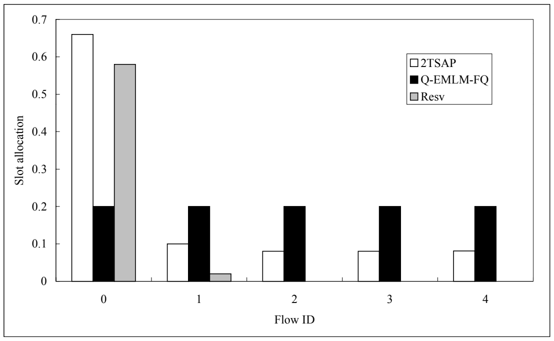
For a best effort flow, its R_f value is zero.

We first randomly generated five flows in a single cluster network. Flows 0 and 1 were guaranteed flows with Resv values of 0.58 and 0.02, respectively. Flows 2, 3, and 4 were best effort flows. Fig. 3(a) shows the slot allocation to all flows for both approaches. The white bars are for 2TSAP, the black bars are for Q_EMLM-FQ, and the gray bars indicate the initial reservation level of each flow. Fig. 3(a) shows that 2TSAP meets the minimum requirement of all flows, and Q_EMLM-FQ fails to satisfy the requirement of flow 1. In fact, Q_EMLM-FQ tends to distribute bandwidth equally among all flows, thus failing to meet the requirements of some guaranteed flows.

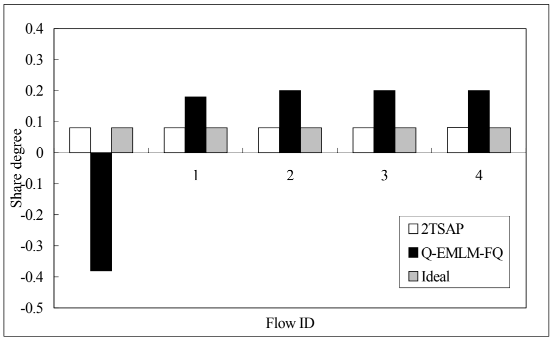
We now turn to the residual bandwidth. We define a share degree to better observe the fairness among all flows, as follows. The share degree is the percentage of residual bandwidth shared by each flow, with a value between zero and one. The larger the value, the more bandwidth the flow uses.

Ideally, the share of the residual bandwidth for each flow should be (1-0.58-0.02)/5=0.08, which is indicated as gray bars in Fig. 3(b). Again, 2TSAP allows flows to share the residual bandwidth fairly, as shown in Fig. 3(b), while Q_EMLM-FQ fails to do so.

Next, we generated five flows again, with three guaranteed flows (i.e., flows 0, 1, 2) and two best effort flows (i.e., flows 3 and 4) in the network logically partitioned into three clusters. Flows 1 and 3 are each located in a separate cluster; flows 0, 2, and 4 are located in another cluster. Fig. 4(a) summarizes the flow information of this scenario. Ideally, flows 0, 2 and 4 share the residual bandwidth in one cluster with each getting a fraction (1-0.54-0.01)/3=0.15. Flows 1 and 3 use all the residual bandwidths of 0.62 and 1 in their respective clusters. Again, from Figs. 4(b) and (c), 2TSAP meets the flows' minimum requirements, and provides per-cluster based, fairly sharing the residual bandwidth for each flow. Q_EMLM-FQ, however, cannot always meet the requirements.



(a) Slot allocation

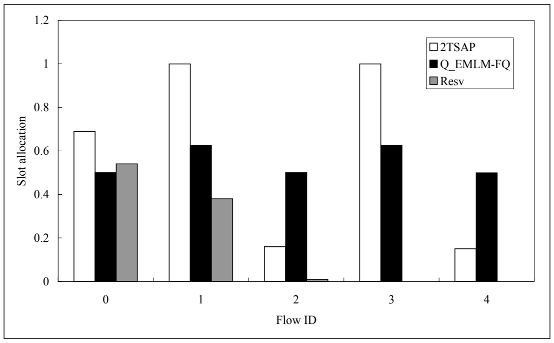


(b) Share degree of each flow

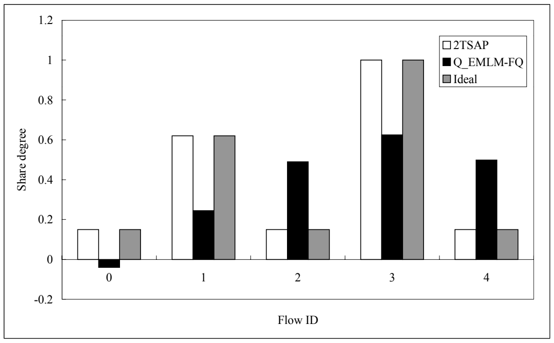
Figure 3. Single cluster

Flow ID	Resv	Source	Dest.	Scheduler
0 (r)	0.54	8	13	3
1 (r)	0.38	18	17	1
2 (r)	0.01	13	11	3
3 (b)	0.00	4	19	4
4 (b)	0.00	11	7	3

(a) Flow information



(b) Slot allocation



(c) Share degree of each flow

Figure 4. Multiple clusters

Fig. 5 shows the overall system throughput for both approaches. The white bars are for 2TSAP and the gray bars are for Q_EMLM-FQ. 2TSAP has better overall network throughput as the number of clusters increases, thanks to the cluster-based structure (similar to the channel reuse concept in cellular systems). In addition, nodes in the same cluster are coordinated with the same scheduler, and only one node can transmit a packet in one slot. Thus 2TSAP can further reduce collisions and increase the throughput.

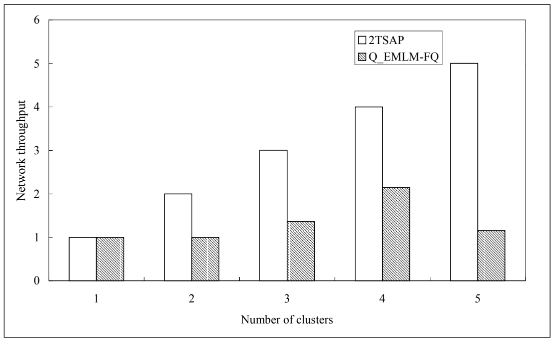


Figure 5. Overall network throughput

4. Conclusion

In this paper, we have proposed a two-tier slot allocation protocol (2TSAP), which provides fair scheduling with QoS support for multihop, multimedia wireless networks. 2TSAP uses the concept of "time-slot usage," instead of "service tag", to make scheduling decisions. It is a two-tier scheduling algorithm. The first tier is for schedulers to assign time slots to mobile nodes in their respective clusters. The second tier is for mobile nodes to assign such time slots to the flows passing through. The simulation result shows 2TSAP improves system throughput and ensures per-cluster fairness while satisfying the requirements of guaranteed flows. In the future, we will extend 2TSAP to handle host mobility in ad hoc wireless networks.

Acknowledgement

This work was supported in part by the National Science Council, Taiwan, under Grant Number 91-2213-E-002 –048, and in part by the Computer & Communication Research Labs of the Industrial Technology Research Institute, Hsinchu, under Grant Number 091E0052.

References

[1] Haiyun Luo and Songwu Lu, "A topology-independent fair queueing model in ad hoc wireless networks," *Proc. IEEE ICNP'00*, pp. 325-335.

[2] Haiyun Luo and Songwu Lu, "A self-coordinating approach to distributed fair queueing in ad hoc wireless networks," *Proc. IEEE INFOCOM'01*, pp. 1370-1379.

[3] E. Hossain and V. K. Bhargava, "A centralized TDMA-based scheme for fair bandwidth allocation in wireless IP networks," *IEEE J. Select. Areas Commun.*, Vol. 19, No.11, Nov. 2001, pp.2201-2214.

[4] P. Goyal, H. M. Vin, and H. Cheng, "Start-time fair queueing: a scheduling algorithm for integrated services packet switching networks," *IEEE/ACM Trans. Networking*, vol. 5, Oct. 1997, pp. 690-704.

[5] N. H. Vaidya and P. Bahl, "Fair scheduling in broadcast environments," *Microsoft Research Tech. Rep. MSR-TR-99-61*.

[6] C. R. Lin and J.-S. Liu, "QoS routing in ad hoc wireless networks," *IEEE J. Select. Areas Commun.*, Vol. 17, No. 8, Aug. 1999, pp.1426-1438.

[7] M. Gerla and J. T-C. Tsai, "Multicluster, mobile, multimedia radio network," *ACM J. Wireless Networks*, Vol. 1, No. 3, 1995, pp. 23-29.

[8] H.-C. Lin and Y.-H. Chu, "A clustering technique for large multihop mobile wireless networks," *IEEE VTC 2000*, pp. 1545-1549.

[9] C. E. Perkins and E. M. Royer, "Ad-hoc on demand distance vector routing," *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, Frbruary 1999, pp. 90-100.

[10] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *ACM SIGCOMM* '94, pp. 234-244.

[11] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in de hoc networks," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, Aug. 1999, pp.1488-1505.