

QUORUM—Quality of Service in Wireless Mesh Networks

Vinod Kone · Sudipto Das ·
Ben Y. Zhao · Haitao Zheng

© Springer Science + Business Media, LLC 2008

Abstract Wireless mesh networks (WMNs) can provide seamless broadband connectivity to network users with low setup and maintenance costs. To support next-generation applications with real-time requirements, however, these networks must provide improved quality of service guarantees. Current mesh protocols use techniques that fail to accurately predict the performance of end-to-end paths, and do not optimize performance based on knowledge of mesh network structures. In this paper, we propose QUORUM, a routing protocol optimized for WMNs that provides accurate QoS properties by correctly predicting delay and loss characteristics of data traffic. QUORUM integrates a novel end-to-end packet delay estimation mechanism with stability-aware routing policies, allowing it to more accurately follow QoS requirements while minimizing misbehavior of selfish nodes.

Keywords QoS · routing · wireless mesh

1 Introduction

Wireless mesh networks (WMNs) have emerged as a popular alternative to provide last-mile connectivity to Internet users. Wireless mesh networks [1] are dynamically self-organizing and self-configuring networks where participating nodes automatically establish and maintain connectivity amongst themselves. These networks are robust and have low up-front and network

maintenance costs. A WMN may be thought as a multi-hop mobile ad-hoc network (MANET) with extended connectivity. They provide a cheaper alternative last mile connectivity than ADSL or Cable networks, and have been adopted by numerous academic and industrial deployments (Champaign–Urbana community wireless network, <http://www.cuwireless.net/>; bay area wireless user group, <http://www.bawug.org/>; wireless LAN infrastructure mesh networks: capabilities and benefits, www.firetide.com; mesh dynamics structured mesh networking for mobile data, video and voice, http://meshdynamics.com/meshdynamics.documents/MD4000_BROCHURE.pdf, Tropos MetroMesh architecture overview, http://www.tropos.com/pdf/metromesh_datasheet.pdf) [2, 3].

As deployments of WMNs continue to grow, we expect these networks to have the ability to support the new generation of streaming-media applications, such as voice over IP (VoIP) and video on-demand (VOD) [4]. These applications require quality of service (QoS) guarantees in terms of minimum bandwidth and maximum end-to-end delay. Most existing work on wireless mesh networks rely on adapting protocols originally designed for mobile ad hoc networks, and offer little support for QoS.

In this paper, we propose a routing protocol for wireless mesh networks that provides QoS guarantees to applications based on metrics of *minimum bandwidth* (B_{\min}) and *maximum end-to-end delay* (T_{\max}). While issues such as end-to-end route discovery have been studied in great depth for WMNs and MANETs [5, 6], our goal is to build a WMN routing protocol that provides “strong” QoS guarantees. By “strong” we mean that our protocol will accept application requests for desired bandwidth and delay bounds for a flow, and

V. Kone (✉) · S. Das · B. Y. Zhao · H. Zheng
Computer Science, University of California, Santa Barbara,
UC Santa Barbara, Santa Barbara, CA 93106, USA
e-mail: vinod@cs.ucsb.edu

either reject the flow if such constraints are not possible, or accept the flow that satisfies those performance bounds at the time of the request. If and when a route is disrupted by a node or link failure, our protocol automatically detects the route breakages, and re-discovers alternate routes if they exist.

This paper makes three key contributions. First, we propose a mechanism that accurately predicts the end-to-end delay of a flow, and show how it can be integrated into flow setup to satisfy QoS requirements. Second, we define a robustness metric for link quality and demonstrate its utility in route selection. This robustness metric supports “intelligent” routing that not only deals with communication gray-zones and fluctuating neighbors [7], but also helps discourage selfish “Free-riding” behavior [8]. Finally, we perform extensive evaluation of our protocol in the Qualnet simulator under a variety of conditions and metrics.

The remainder of this paper is organized as follows. Section 2 describes previous work on routing and QoS support in wireless networks. Second, Section 3 describes our network model and design objectives, and highlights the major challenges in providing strong QoS guarantees. Next, Section 4 describes the details of the QUORUM routing protocol. Finally, we describe our simulation setup and results in Section 5 and conclude in Section 6.

2 Related work

Wireless networks has been an active area of research interest and a significant work has been done on routing in wireless networks [9–11] and MANETS [5, 6]. But there has been a relatively less focus on providing “strong” QoS guarantees for WMNs. Most of the existing work is either focused on MANETs or WMNs which have multiple radios. A review of relevant literature shows that various approaches have been taken to provide QoS guarantees.

Some researchers advocate for a stateless approach [12], while others have advocated maintaining state at intermediate nodes [13–15]. Providing a stateless solution in [12], the authors describe a way to achieve QoS routing without using explicit reservation mechanisms and give new distributed solution to oscillation and collision of flows. This paper proposes QoS based on OLSR, and has both the advantages and disadvantages of the underlying proactive routing protocol. Moving on to the stateful approaches, Chen et. al. [15] propose a Distributed QoS Routing Scheme where the path is computed by the exchange of control messages, and the state information kept at each node is collectively used

to find a path. WMR [14] is another stateful protocol that has been proposed to provide QoS enabled routing in WMNs and is the result of modifying its MANET counterpart, AQOR [13] to the wireless mesh context. Both AQOR and WMR cannot provide strong delay guarantees, since they perform delay estimation using route request (RREQ) and route reply (RREP) messages. Our experiments in Section 5 show these delay estimators to be highly inaccurate.

Other approaches include use of channel switching [16] where APs use multiple channels and Mobile Hosts (MHs), upon detection of a QoS violation, switch channels to connect to another AP. Another approach [17] proposes clustering of end hosts and use of orthogonal channels to reduce the effect of interference. A very different method [17] suggests the use of a statistical mechanics technique called Annealing. In [17], the authors propose a new QoS routing protocol for wireless mesh networks. They use delay and bandwidth as the QoS parameters and then use Mean Field Annealing for finding a suitable path. MFN_RS uses deterministic equations to replace stochastic processes in simulated annealing (SA) and saddle point approximation in the calculation of stationary probability distribution at equilibrium.

Even though there has been some work in designing solutions for providing QoS enabled routing for WMNs, none of these protocols deliver “strong” QoS guarantees in terms of latency or throughput metrics. Our work, QUORUM, is a stateful approach that performs *On-demand* route discovery and selection using multiple metrics like bandwidth, delay, and robustness (discussed in details in Section 3.2.1) while providing “strong” QoS guarantees. Even though the problem of providing QoS guarantees based on multiple constraints has been shown to be NP Complete [17], it is also known that with suitable heuristics, a multi-constrained QoS routing algorithm can work in polynomial time [18].

QUORUM differs from the work described above in various aspects. *Firstly*, our network uses single radios throughout (although the Mesh Routers use multiple radios, one radio is used to communicate with the clients while the other is used to communicate with other MRs). Although recent research [10, 11] propose the use of multiple radios, we go for single radios for simplicity, energy efficiency, and lower cost. *Secondly*, QUORUM is a reactive protocol that maintains “soft” state at all the nodes in the network. *Thirdly*, QUORUM uses a novel DUMMY-RREP phase to accurately predict the delay that will be experienced by the flows in the network. *Finally*, QUORUM uses a concept of Robustness of the routes. The robustness of the nodes

are calculated by using the HELLO messages used to exchange the information about the reservations at each node. The robustness metric not only allows rejection of transient routes and working around the fluctuating neighbor problem, but it also equips the protocol with the inherent ability to tackle a type of “Selfish” behavior referred in literature as “Free-riding” [8].

3 Network model and challenges

In this section, we define our context by outlining our model of the network, then describe two key challenges that motivated the development of the QUORUM routing protocol.

3.1 Network model

We use a hybrid mesh architecture [1] consisting of three types of nodes, *Mesh Clients* representing end users, *Mesh Routers* that communicate with clients and other mesh routers, and *Internet Gateways* that communicate with mesh routers and the external Internet. An example is shown in Fig. 1. Mesh routers and clients run the same routing protocol. Mesh routers have two interfaces operating on orthogonal channels, one for communicating with mesh clients and other for communicating with other mesh routers. Mesh clients have only one interface. Three types of routes are possible: those that connect two mesh clients served by the same mesh router, those that connect mesh clients served by different mesh routers, and those that connect a

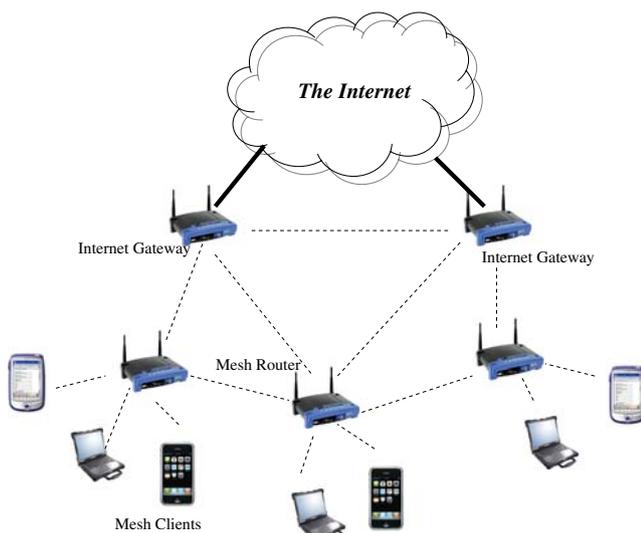


Figure 1 WMN hybrid network architecture consists of mesh routers, mesh clients and internet gateways. A router has two interfaces, one for clients and one for other routers

mesh client to an Internet host. Also, note that our network uses Layer 3 routing throughout in order to leverage the advantages of MANETs, including ease of deployment and extended connectivity.

3.2 Design challenges

This section describes two major design challenges for a QoS-enabled mesh routing protocol, and forms the basis for two of the major contributions made by this paper: detecting and avoiding “fragile” routes, and producing accurate estimates of a flow’s end-to-end delay.

3.2.1 Measuring link robustness

A significant challenge faced by existing routing protocols is the communication gray zone problem [7]. Most routing protocols such as AODV [5] and WMR [14] rely on control (RREQ) packets to detect and establish end to end routes. However, these control (broadcast) packets have properties that differ significantly from data packets. To be more specific, the three main differing characteristics are (1) broadcast packets are sent at lower bit rates which make them more reliable than data packets for decoding (2) smaller sizes of broadcast packets result in lower probability for collision and (3) no acknowledgment for broadcast packets means the flow is unidirectional whereas data with acks constitutes a bidirectional flow. As a result of these differences, each node has a strip of surrounding region called gray zone. Neighboring nodes present in this zone might be able to successfully receive broadcast packets but might not be able to receive data packets (Fig. 2). Note that, the next hop neighbor of a node for an end to end route is selected based on the neighbor’s capability of sending and receiving RREQs, which are broadcast packets. Hence, there is a significant chance that many of the nodes along the chosen route fall into the gray zones’ of their neighbors on the route. Consequently, once the data starts flowing along this “fragile” route some

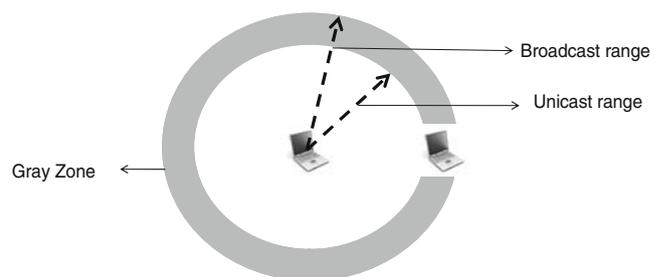


Figure 2 Communication gray zone. Neighbors in the gray zone may receive broadcast packets but may not receive data packets

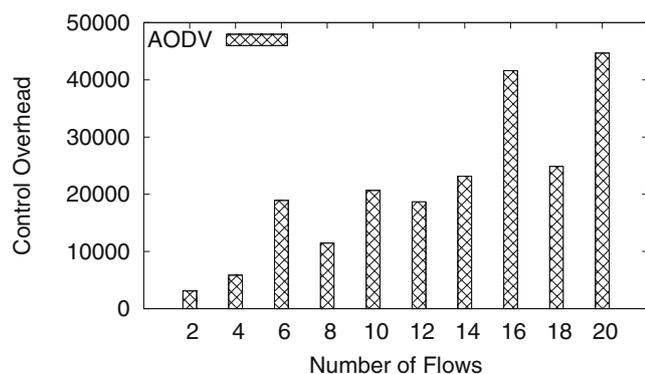


Figure 3 Control Overhead in a 50 node network. Overhead is calculated as the total number of RREQs forwarded by all the nodes in the network

nodes will not be able to receive data packets, triggering frequent link repairs. Since, link repairs usually entail re-routing process which includes broadcast flooding of RREQ packets and RREPs there would be enormous amount of overhead in the network. This results in a performance deterioration especially for QoS routing protocols because of the disruption of flows due to the overhead of the additional control traffic.

To demonstrate this, we quantify the control overhead in a network of 50 nodes (Fig. 8) with varying number of flows in the network. Each flow is required to send 10,000 data packets at a rate of 50 kbps. Flows are selected randomly in each run and the simulation is averaged over 20 runs. As Fig. 3 shows, there is an astronomically high amount of control packet overhead in the network as the number of flows increases. As explained earlier, this is due to the large amount of broadcast floods introduced into the network as part of frequent re-routing due to route breaks. Here we quantify overhead as the number of RREQs forwarded by all nodes in the network. Understanding the gray zone phenomenon motivates us to design a robust routing algorithm that detects and avoids fragile routes, thereby significantly reducing control overhead.

3.2.2 End-to-end delay estimation

A critical component of any QoS-enabled routing protocol is end-to-end delay estimation. Current protocols estimate end-to-end delay by measuring the time taken to route RREQ and RREP packets along the given path. We observe, however, that RREQ and RREP packets are significantly different from normal data packets, and are therefore unlikely to experience the same levels of traffic delay and loss as data packets.

We perform an experiment to quantify the error introduced by two estimation methods (RREP packets

and hop count) to measure end-to-end delay. RREP technique estimates the delay by measuring the delay experienced by RREQ packets and the corresponding RREP packets. The Hopcount technique estimates the delay as the number of hops \times the average per-hop delay. We select a small topology of 14 nodes and introduce a single 5-hop flow into the network. As shown in Fig. 4, both the techniques introduce significant estimation errors: RREP overestimates and Hopcount underestimates the actual delay experienced by the data packets.

There are two main reasons for the significant discrepancy between the RREP estimate and the actual end-to-end packet delay, both based on wireless interference. First, RREQ packets are flooded across multiple routes in the network during route discovery. The result is a burst of simultaneous traffic across a large number of links. These RREQ packets propagating along different routes interfere with each other, causing *inter-flow* interference, which unicast data does not experience because it follows a single path at any given time. The second factor is the *intra-flow* interference experienced by data packets. When a stream of packets traverse a route, the broadcast nature of the underlying wireless network means different packets of the same flow will interfere with each other, resulting in media contention and per-packet delays. Control packets such as RREQ do not experience intra-flow interference because there is no stream of packets following one single path. RREP overestimates because it accounts for the dominant but non-existent inter-flow interference whereas Hopcount underestimates the delay because it doesn't account for the intra-flow interference. This motivates an in-band delay estimation mechanism for end-to-end packet delay. To address this, QUORUM introduces a DUMMY-RREP latency estimator in Section 4.3.

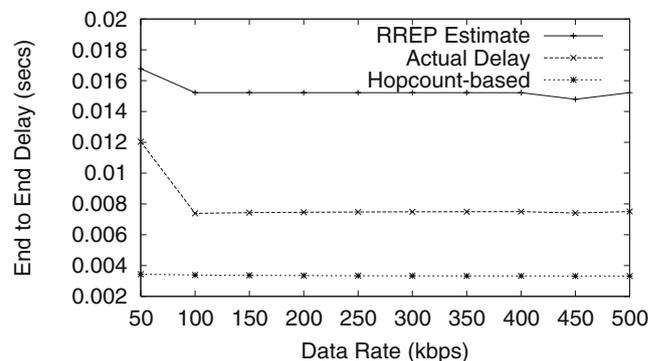


Figure 4 End to end delay estimation of the data packets. RREP overestimates end-to-end delay because of inter-flow interference. The hopcount-based approach underestimates the delay since it does not account for intra-flow interference

Table 1 Structure of a flow table

SRC	DEST	B/W Reserved	B_{\min}	T_{\max}	Quality	I/F
-32bits-	-32bits-	-32bits-	-32bits-	-32bits-	-32bits-	-32bits-

4 The QUORUM routing protocol

The goal of the QUORUM routing protocol is to provide a QoS-constrained route from source to destination. Specifically, the route selected by the protocol should deliver packets with minimum bandwidth B_{\min} and end-to-end latency less than T_{\max} , where both parameters are specified by the application at flow initiation time. In addition, QUORUM should choose the most robust among all possible candidate routes satisfying the above constraints.

QUORUM is a reactive protocol that discovers routes on-demand. During the route discovery phase of the protocol, each intermediate node uses an admission control scheme to check whether the flow can be accepted or not. If accepted, a Flow Table (Table 1) entry for that particular flow is created. For specifics of the admission control scheme, we refer the reader to protocols such as AQOR [13] and WMR [14]. Basically, each node collects the bandwidth reserved at its one hop neighbors (piggybacked on periodic HELLO packets) and stores it in its Neighbor Table (Table 2).

While QUORUM borrows the admission control scheme from AQOR and WMR, there are several key differences. The main drawback of WMR [14] is that it does not leverage knowledge of the mesh network topology. In contrast, QUORUM treats mesh routers and clients differently (See Section 4.2). Another difference is that AQOR and WMR select the route on which the first in-time RREP packet arrives, whereas QUORUM uses periodic messages to estimate link quality, and selects the most robust route whenever possible. This means that QUORUM considers three metrics (bandwidth, delay and robustness) while AQOR deals with only bandwidth and delay. We describe the novel aspects of QUORUM in the remainder of this section.

4.1 Estimating route robustness

Each node in the network estimates the robustness of its links to its one-hop neighbors. Nodes estimate a link's quality or robustness by measuring the number

of HELLO packets received during a rolling window of time. Measurements of the recent window is combined with a historical value (Q) as an exponentially weighted moving average (EWMA) to compute the updated estimate. Specifically, each node computes a rolling CQ , the percentage of HELLO packets received in the last ROBUSTNESS_INTERVAL seconds. A link's robustness is computed as: $R = \alpha \cdot CQ + (1 - \alpha) \cdot Q$.

Each node maintains estimates of link robustness to all of its neighbors in the Neighbor Table. Nodes compute the robustness of an end-to-end route as the average link quality across all links along the path. Link quality estimates are accumulated by RREQ packets on the reverse path, and RREP packets on the forward path.

By using end-to-end robustness to differentiate between candidate routes, QUORUM avoids unreliable routes or those that cross communication gray zones. Since gray zone neighbors have lower link quality with respect to their corresponding nodes, routes consisting of such neighbors have lower end to end stability compared to more robust routes. More precisely, a node doesn't forward packets (control and data) of its neighbor if its robustness is less than 50% and hence the protocol avoids routes with unstable nodes during route selection. This threshold is selected based on experimental runs and is a trade off between eliminating gray zone neighbors and avoiding false positives. The result is a more robust end-to-end path that avoids the high overhead of route repair messages.

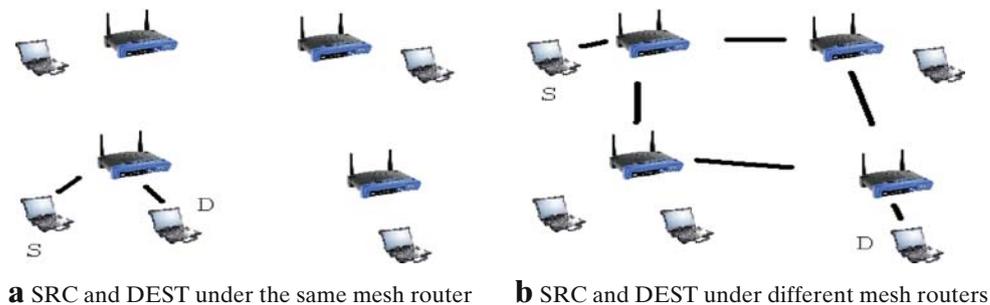
4.2 Topology-aware route discovery

We optimize QUORUM for hierarchical wireless mesh networks by limiting the flooding of control messages using explicit knowledge of the network topology. Recall that for streaming-media applications such as Video-on-Demand, much of the data traffic can be localized to a mesh group if the request can be met locally by data caches. In these cases, broadcasting control packets beyond the mesh group creates unnecessary network congestion and disruption to other flows.

Table 2 Structure of a neighbor table

DEST	B/W Reserved	# of Hello Pkts	Quality	I/F
-32bits-	-32bits-	-32bits-	-32bits-	-32bits-

Figure 5 Selective flooding. If source and destination reside in the same mesh group, we limit control packet flooding to the mesh group. Otherwise, control packet flooding is limited to the source and destination mesh groups and between mesh routers



We illustrate two examples of this technique in Fig. 5. Figure 5a shows a scenario where both the source and destination are under the same mesh router (MR). Here it is logical to limit the control flood to nodes served by the local router. If the source and the destination are under different MRs, as in Fig. 5b, then control traffic should be limited to the two mesh groups and all mesh routers, avoiding unnecessary congestion in mesh groups without the source or destination. We achieve this topology awareness by requiring mesh clients in the same mesh group to reside in the same unique subnet. Mesh routers then make intelligent decisions that limit the propagation of control packets.

Also, as explained earlier, we limit control packet flooding by having nodes accept flooding messages only from “robust” neighbors (those with link quality >50%). As can be seen later in the experimental section, this scheme reaps huge benefits in terms of reduction of control overhead in addition to addressing the communication gray zone problem.

4.3 Estimating end to end delay

As we showed in Section 3.2.2, end-to-end delay reported by RREQ-RREP measurements differs substantially from delay experienced by actual data packets. To address this, we introduce a DUMMY-RREP phase during route discovery. The aim of this phase is to accurately estimate the delay that will be experienced by the actual data packets by sending a dummy stream of packets having identical characteristics to data packets. When a source receives RREP packets, it saves them in a RREP_TABLE. The source then takes the RREP for a route from this table and sends a stream of DUMMY data packets along the path traversed by this RREP. DUMMY packets have the same size, priority and data rate as real data packets, effectively emulating the interference experienced by the actual data packets on a particular path. Each stream includes $2H$ number of packets, where H is the hopcount reported by the RREP. This parameter balances the trade-off between control overhead and measurement accuracy, and is

based on our experiments over a number of flows with varying path lengths. Figure 6 plots the result of one of our experiments where we introduce a single 5 hop flow into the network and observe the delay prediction of dummy stream with varying number of constituent packets. We can see that, dummy stream of 10 packets ($2 \times$ hopcount) or more experiences similar intra-flow interference as the actual data packets and hence predicts the end to end delay fairly accurately.

The destination calculates the average delay of all DUMMY packets received, and reports it back to the source via a RREP. If the average delay reported by this RREP is within the bound requested by the application, the source selects this route and starts sending data packets. Soft-state timers are included at both source and destination to take care of lost packets. If the reported delay exceeds the requested limit, the source does a linear back-off and sends the DUMMY stream on a different route selected from its RREP_TABLE. Figure 7 shows the approximate route setup delay of QUORUM compared to AODV. For this experiment, flows of 50 kbps were randomly chosen from the 50 node topology

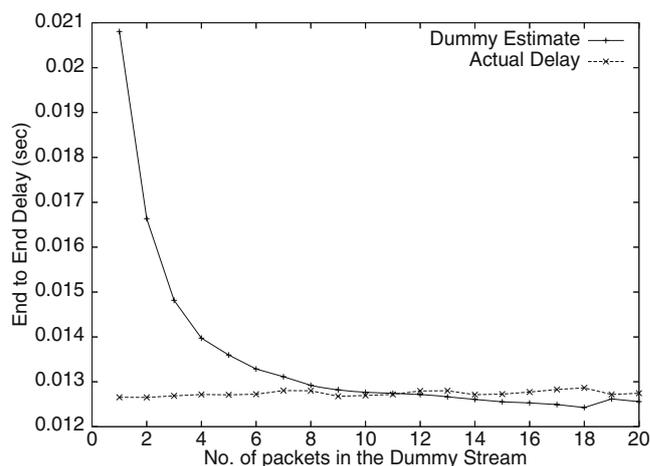


Figure 6 Delay estimation vs Number of packets in dummy stream. For a five hop flow, we can observe that ten packets in the dummy stream is sufficient to predict the actual delay experienced by the data packets

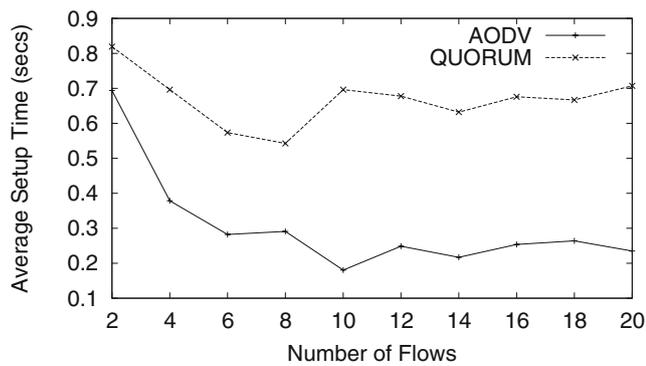


Figure 7 Route setup delay vs number of flows. Each point corresponds to the average setup time taken by the corresponding number of flows in the network. QUORUM takes relatively longer time due to the additional DUMMY-RREP phase

in Fig. 8. We see that QUORUM takes longer than AODV to set up a route because of the DUMMY-RREP phase, but it is a reasonable trade-off given the resulting accurate end-to-end delay estimation.

4.4 Tackling misbehaving nodes

Another key advantage that QUORUM has over other QoS routing protocols is the ability to punish and discourage selfish “free-riding” behavior. In real networks, selfish nodes can utilize the network infrastructure for routing while avoiding forwarding other nodes’ packets. In QUORUM, a misbehaving node can achieve this by not broadcasting HELLO packets while listening to neighbors’ broadcasts. Since neighbors have no information about the misbehaving node, they select their routes via other neighbors. Meanwhile, the misbehaving node can still use its neighbors to route its own packets.

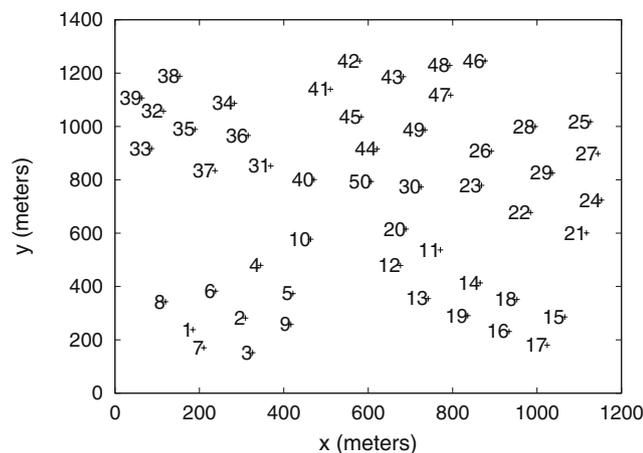


Figure 8 Our simulation topology of 45 mesh clients and 5 mesh routers (10, 20, 30, 40, 50). Each router (n) is responsible for 9 mesh clients ($n - 9, n - 8, \dots, n - 1$)

To discourage this behavior, we propose a simple variant of the popular “Tit-for-Tat” rule based on link robustness. According to this rule, a node does not forward the packets of a neighbor if the neighbor’s link quality is lower than a certain threshold. In this case, neighbors of a selfish node will estimate its link quality as 0 and the node’s packets are dropped by the neighbors due to low robustness. In effect, the robustness metric provides an incentive for a cooperative environment in the network. Recall that to deal with the communication gray zone problem, a node only accepts control packets from nodes that have robustness above a particular threshold. This link quality/robustness threshold serves a dual purpose and is a critical component of the QUORUM protocol.

4.5 QoS violation and recovery

QUORUM detects changes in path quality that violate QoS guarantees with the help of reservation timeouts of Flow Table entries. We identify two different QoS violations as follows: In the first case, an intermediate node receives a data packet but does not have a corresponding Flow Table entry for that flow. This means that the node has deleted the Flow Table entry because of a reservation timeout. Hence, it sends a Route Error (RERR) packet back to the source which re-initiates route discovery and re-routes the packets. A second case is where the destination detects, with the help of its Flow Table, that data packets arriving at it are exceeding the T_{max} requested by the source. In this case, the destination increments its sequence number and broadcasts an unsolicited RREP back to the source. On receipt of this RREP, the source immediately re-routes packets via the path traveled by this RREP thus avoiding the lengthy re-routing process. This scheme is similar to the recovery scheme used by AQOR [13].

5 Simulation results

We perform detailed evaluation of our protocol using the Qualnet (<http://scalable-networks.com>) simulator. Despite our best efforts, however, we were unable to obtain a Qualnet implementation of an existing QoS-routing protocol to compare with.

5.1 Experimental setup

Our network topology, shown in Fig. 8, consists of 50 nodes (5 mesh routers and 45 mesh clients). Each mesh

router is responsible for forwarding outgoing traffic of clients in its group. All nodes are static and are placed in an area of $1500\text{ m} \times 1500\text{ m}$. The protocol is implemented on top of 802.11b MAC protocol with a raw channel bandwidth of 11 Mbps at each node. Note that unlike our architecture, there is no explicit gateway node in our simulations, since any mesh router can act as a gateway node. For any traffic destined outside the WMN, our routing protocol provides guarantees only till the Internet Gateway.

Application traffic is sent as CBR with 512 byte packets. Each flow source sends a maximum of 10,000 packets to its destination. After 10 s for network stabilization, flows are introduced gradually into the network. Each flow is alive for 10 min, and each simulation run lasts for 15 min. For our robustness computations, nodes broadcast HELLO packets every 200 ms, and compute robustness values once per second, with the EWMA weight factor, $\alpha = 0.5$.

5.2 QoS routing behavior

The experiments in this sub-section demonstrate the effectiveness of QUORUM in guaranteeing QoS to the different flows in the network. In order to signify this, we develop a scenario where we congest the network so that admission control comes into play. As AODV is best effort, it will try to deliver packets from all the sources, while QUORUM would try and provide QoS guarantees and in its quest to do so, it might reject some flows based on the load in the network. As is evident from Fig. 9a, we select 5 flows (F:S – D in the figure refers to the flow whose source is S and destination D) in the topology in Fig. 8. As can be seen, all the flows originate in the same subnet of MR 30 and

all flows except one end in the same subnet. Each of the flows request a bandwidth of 500 kbps. The experiment is repeated for 20 seeds, but the flows remain the same in each run. The flows are started in the order they are shown in Fig. 9a.

As shown in Fig. 9a, QUORUM rejects F:27–28 in a number of scenarios since it is the last requested flow, but provides delivery within the requested delay to all admitted flows. On the other hand, AODV tries to deliver packets from all the flows resulting in excessive contention and very high delays. Figure 9b demonstrates the fact that QUORUM does not compromise on the delivery of the packets for the flows accepted. Packet delivery ratio (PDR) is calculated as the ratio of packets received by the destination to the packets sent by the source. From the figure it is clear that QUORUM guarantees high PDR for those flows that are admitted into the network. In the rest of the paper, we use PDR only to refer to flows that have been admitted into the network by AODV or QUORUM.

5.3 Control overhead

Control overhead of the protocol is defined as the total number of control packets forwarded by all the nodes in the network. For QUORUM, we also include the DUMMY packets as an overhead induced by the protocol (in addition to RREQs which is common in both). This experiment shows the benefits of intelligent routing in QUORUM. We note that AODV has an inherent advantage when it comes to overhead, because a source node does not need to send RREQ packets to a destination whose route it knows implicitly from previous flows. In QUORUM, a source must send RREQ packets to all destinations, even for those whose routes are

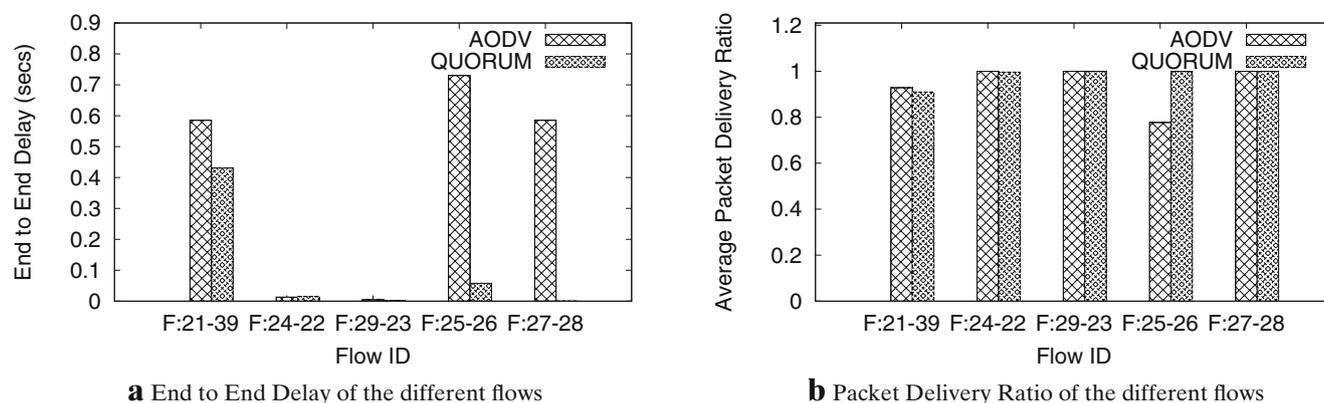


Figure 9 In the figure we can observe that F:27–28 is almost always rejected by QUORUM to provide relatively lower delays to the accepted flows. Figure shows that QUORUM provides higher packet delivery ratio to all the accepted flows

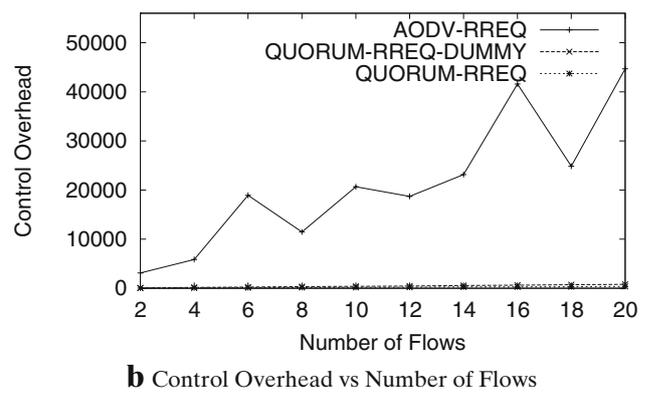
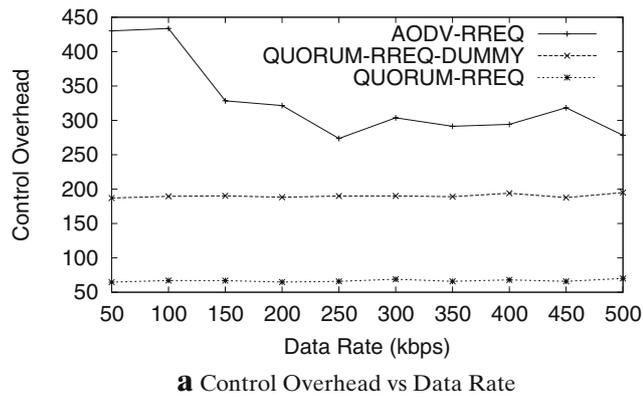


Figure 10 From the figure we can see that control overhead in QUORUM is reduced by 40% compared to AODV due to intelligent routing. Figure shows the high overhead induced by

AODV due to its susceptibility to select unstable routes, unlike QUORUM which selects stable routes and hence keeps the control overhead to a minimum

known. The control messages are required to estimate whether bandwidth and delay constraints are satisfied on any given path.

routes by having a robustness threshold as described in Section 4.2.

Figure 10a and b plots the control overhead of the protocols with varying data rates and varying number of flows in the network. We also plot the RREQ-only overhead of QUORUM to see the amount of overhead actually reduced by the intelligent routing. Figure 10 shows that control overhead of QUORUM is lower than AODV by 30-35%.

5.4 End to end delay estimation

Figure 10b shows a drastic increase in the control overhead of AODV with increase in the number of flows in the system. The reason for this astronomically high overhead is the susceptibility of AODV protocol to choose unstable routes which are better in terms of hop count. Due to this, the sources in AODV end up selecting unstable routes which break often, resulting in re-routing and hence higher control overhead. QUORUM refrains from accepting unstable

This section evaluates one of the major contributions of this paper, end to end delay estimation during route setup. We show the usefulness of the DUMMY-RREQ phase in the estimation of end to end delay. The delay estimated by the RREP packets, delay estimated by the DUMMY-RREQ phase and actual delay experienced by the data packets are analyzed for varying data rates (50–100 kbps) and varying flows (each requesting a B_{min} of 50 kbps). As shown in Fig. 11, delay estimated by the RREQ-RREP phase differs from the actual delay by a considerable margin. By having the DUMMY stream emulate the data packets, QUORUM is able to accurately estimate the actual end to end delay.

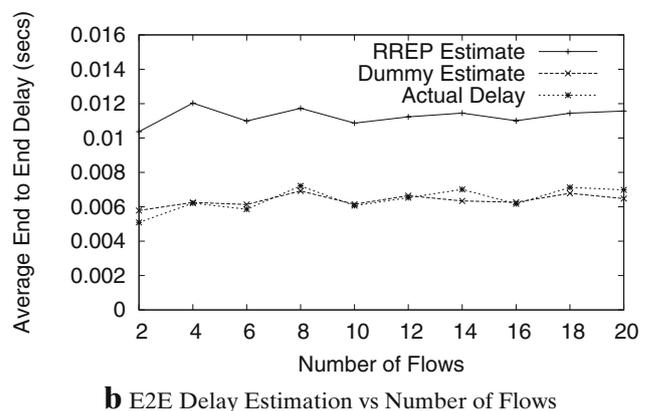
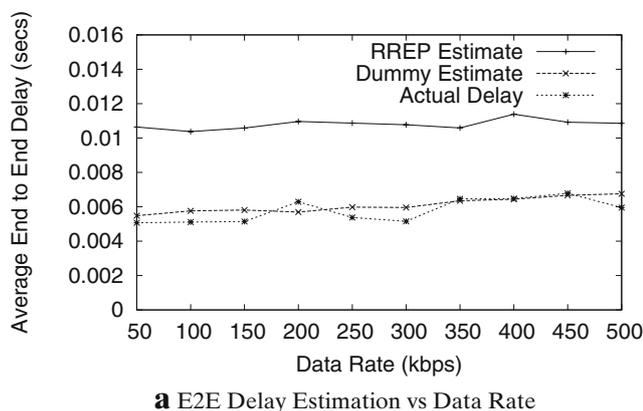
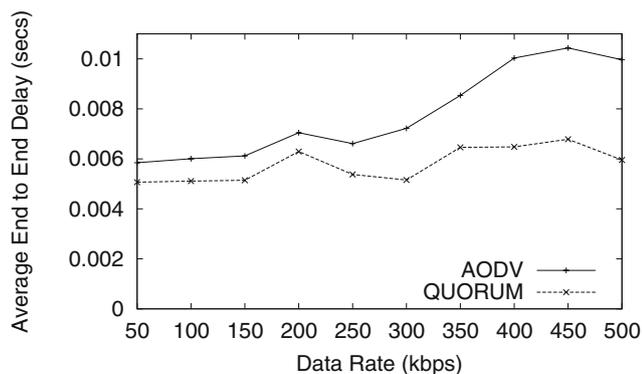
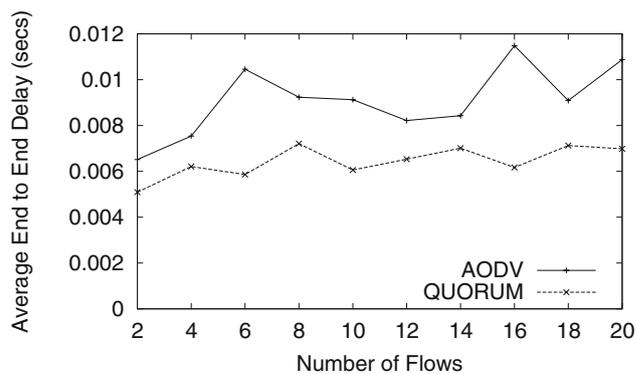


Figure 11 End to end delay estimation. In (b) each flow requests a B_{min} of 50 kbps. We can observe that, in both the cases, QUORUM does a good job of estimating the actual delay fairly accurately



a Average End to End Delay vs Data Rate



b Average End to End Delay vs Number of Flows

Figure 12 Average end to end delay experienced by the data packets is consistently lower in QUORUM when compared to AODV. This is because QUORUM always selects paths which

satisfy the delay bound requested by the application. In (b) the average delay of the flows in QUORUM is 33% lower than that experienced in AODV

5.5 Scalability

The main goal of this experiment is to test how QUORUM scales as the number of flows in the network increases and when the data rates of the flows increase. The metrics used are average system throughput, packet delivery ratio and average end to end delay. For the varying data rate experiment we randomly picked five flows and for varying flows experiment each randomly picked flow requested a B_{min} of 50 kbps. We do not plot the graphs for the average system throughput and the packet delivery ratio because both AODV and QUORUM achieved equally high system throughput and PDR of 0.98 and higher.

Figure 12 plots the average end to end delay experienced by the accepted flows in the network. In both the experiments QUORUM out-performs AODV. This is

because of the ability of QUORUM to select stable routes in which the data packets experience acceptable delays, verified by the DUMMY-RREP phase.

5.6 Tackling misbehaving nodes

Another contribution of QUORUM is its inherent ability to tackle selfish misbehavior or free-riding by providing an incentive to co-operate. The misbehavior model is as described in Section 4.4. In this experiment, the average throughput of “bad” and “good” flows are calculated separately. A flow is considered “bad” if either its source or destination is a misbehaving node, otherwise its considered “good”. The misbehaving nodes are selected randomly for each of the 20 runs and only mesh clients can misbehave. Ten flows of 50 kbps data rate are randomly picked for each run.

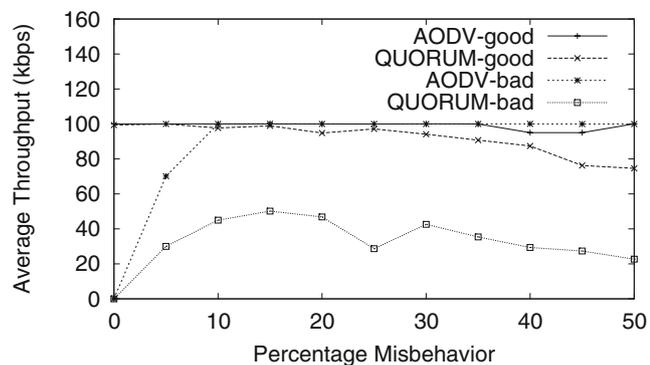


Figure 13 We plot throughput values for both “good” and “bad” flows in AODV and QUORUM. QUORUM clearly discourages selfish behavior by denying them network bandwidth

From Fig. 13, we can see that AODV allows free-riding of the misbehaving nodes, while the throughput of the misbehaving nodes is considerably reduced in QUORUM. Percentage misbehavior refers to the percentage of the nodes in the network that misbehave. It is interesting to note that AODV is not affected greatly by this kind of misbehavior because it doesn’t rely on the HELLO packets for its routes unlike QUORUM. Though QUORUM gets affected by the HELLO packet misbehavior, we can observe that free-riding of misbehaving nodes is reduced to a great extent. This is because misbehaving nodes have robustness of zero in the eyes of their neighbors and hence their control packets for routing are not forwarded because of their low robustness.

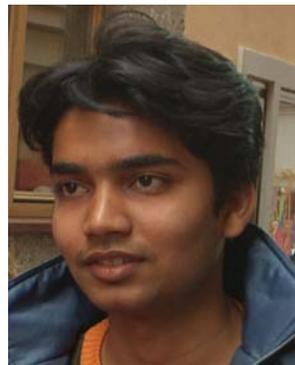
6 Conclusion

In this paper, we have developed QUORUM, a novel QoS-aware routing protocol for wireless mesh networks. Specifically, QUORUM takes three QoS metrics into account: bandwidth, end to end delay and route robustness. To optimize QUORUM for wireless mesh networks, we propose several mechanisms including topology-aware route discovery that drastically reduce the control overhead and network congestion from route discovery. In addition, we introduce the novel DUMMY-RREP data latency estimator, and show it to be effective in providing accurate estimates of end-to-end delay experienced by data packets. Finally, our proposed link robustness metric allows QUORUM to punish and discourage free-riding behavior by selfish nodes, a side effect of the robustness metric.

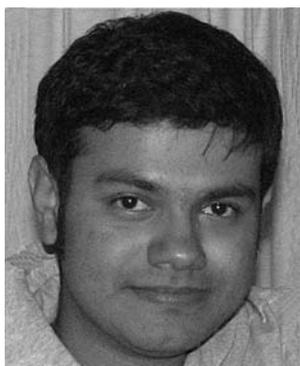
Acknowledgements We would like to thank the anonymous reviewers for their helpful comments, and Prof. Elizabeth Belding for her feedback on earlier versions of this work. This work is supported in part by NSF under CAREER Award #CNS-0546216 and DARPA under the Control Plane project (BAA 04-11).

References

- Akyildiz I, Wang X, Wang W (2005) Wireless mesh networks: a survey. *Comput Networks* 47:445–487
- Bicket J, Aguayo D, Biswas S, Morris R (2005) Architecture and evaluation of an unplanned 802.11b mesh network. In: *Proc. of MobiCom*, New York, NY
- Raman B, Chebrolu K (2007) Experiences in using WiFi for rural internet in India. *IEEE Commun Mag* 45(1):104–110, January
- Allen M, Zhao BY, Wolski R (2007) Deploying video-on-demand services on cable networks. In: *Proc. of ICDCS*, Toronto, Canada, June 2007
- Perkins C, Royer E (1999) Ad hoc on-demand distance vector routing. In: *Proc. of WMCSA*, February 1999
- Johnson DB, Maltz DA, Broch J (2001) DSR: the dynamic source routing protocol for multi-hop wireless Ad Hoc networks. *Ad hoc Networking* 5:139–172
- Lundgren H, Nordstrom E, Tschudin C (2002) The gray zone problem in IEEE 802.11b based Ad hoc Networks. *MC2R* 6(3):104–105
- Mahajan R, et al (2005) Sustaining cooperation in multi-hop wireless networks. In: *Proc. of NSDI*
- Nandiraju NS, Agrawal DP (2006) Multipath routing in wireless mesh networks. In: *Proc. of MASS*
- Draves R, Padhye J, Zill B (2004) Routing in multi-radio, multi-hop wireless mesh networks. In: *Proc. of MobiCom*, New York, NY
- Draves R, Padhye J, Zill B (2004) Comparison of routing metrics for static multi-hop wireless networks. In: *Proc. of SIGCOMM*, Portland, OR
- Badis H, Gawedzki I, Al Agha K (2004) QoS routing in ad hoc networks using QOLSR with no need of explicit reservation. In: *Proc. of VTC*, September 2004
- Xue Q, Ganz A (2003) Ad hoc QoS on-demand routing (AQOR) in mobile ad hoc networks. *J Parallel Distrib Comput* 63:154–165
- Xue Q, Ganz A (2002) QoS routing for mesh-based wireless LANs. *Int J Wirel Inf Netw* 9:179–190, July
- Chen S, Nahrstedt K (1999) A distributed quality-of-service routing in ad-hoc networks. *IEEE JSAC* 17(8):1488–1505, August
- Wang H, Yow KC (2006) QoS routing in multi-channel multihop wireless networks with infrastructure support. In: *InterSense*, New York, NY
- Liu L, Feng G (2005) Mean field network based QoS routing scheme for wireless mesh networks. In: *Proc. of WiCOM*, September
- Kuipers FA, Miegheem PFAV (2005) Conditions that impact the complexity of QoS routing. *IEEE/ACM Trans Netw* 13(4):717–730



Vinod Kone received his B.Tech (Hons.) degree in Computer Science and Engineering from IIT Guwahati, India in 2006. He is currently a Ph.D. student in the Department of Computer Science at U. C. Santa Barbara. His research interests include wireless networks, network protocols and distributed systems. He is a recipient of the Pratibha Merit Scholarship (2002–2006) and the Citrix Online Graduate Fellowship (2006–07).

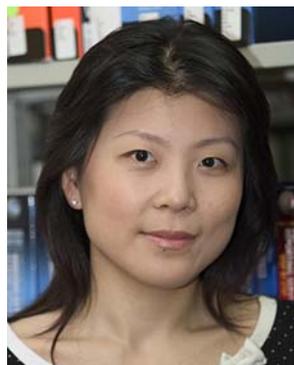


Sudipto Das received his B.E. degree in Computer Science and Engineering from Jadavpur University, India in 2006 and is currently a Ph.D. student in the Department of Computer Science at U. C. Santa Barbara. He has worked in the area of wireless networks, especially performance optimizations for wireless ad-hoc routing protocols. He is currently associated with the DatabaseSystems Lab at UCSB and his current research interests lie in investigating the use of parallel hardware to accelerate various Database and Data Stream Queries.



Ben Y. Zhao is a faculty member at the Computer Science department, U.C. Santa Barbara. Before UCSB, he completed his

M.S. and Ph.D. degrees in Computer Science at U.C. Berkeley, and his B.S. degree from Yale University. His research spans the areas of security and privacy, networking and distributed systems. He is a recent recipient of the National Science Foundation's CAREER award (2005), the MIT Tech Review's TR-35 Award (Top 35 Young Innovators Under 35) in 2006, and Computer World's Top 40 IT Innovators Award (2007).



Haitao Zheng received her B.S. from Xian Jiaotong University in 1995, her M.S.EE and Ph.D. degrees in Electrical and Computer Engineering from University of Maryland, College Park, in 1998 and 1999, respectively. She has held research positions at the wireless research lab at Bell-Labs and as a project lead at Microsoft Research Asia. In 2005, she joined the Computer Science department at U. C. Santa Barbara. Dr. Zheng is a recipient of the 2005 MIT Technology Review Top 35 Innovators under 35 award, best student paper award at IEEE DySPAN 2007, 2002 Bell Laboratories President's Gold Award, and the 1998–1999 George Harhalakis Outstanding Graduate Student Award from University of Maryland. Dr. Zheng's research interests include wireless systems and networking and multimedia computing.