QUORUM – QUality Of service RoUting in wireless Mesh networks

Vinod Kone, Sudipto Das, Ben Y. Zhao and Haitao Zheng
University of California, Santa Barbara
{vinod, sudipto, ravenben, htzheng}@cs.ucsb.edu

ABSTRACT

Wireless Mesh Networks (WMNs) can provide seamless broadband connectivity to network users, with the advantage of low setup and maintenance costs. To support next-generation applications with real-time requirements, however, these networks must provide improved Quality of Service guarantees. Most current mesh network routing protocols are adapted from MANET protocols, and do not optimize for mesh network properties. In this paper, we propose QUORUM (QUality Of service RoUting in wireless Mesh networks), a routing protocol optimized for WMNs that addresses these drawbacks. 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.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

Keywords

QoS, Routing, Wireless Mesh

1. INTRODUCTION

The last couple of years has seen the emergence of Wireless Mesh Networks (WMNs) as a new networking paradigm. A WMN is a dynamically self-organizing and self-configuring network where participating nodes can automatically establish and maintain connectivity amongst themselves. Wireless mesh 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. Widespread popularity of the WMNs is evident from the number of academic and industrial deployments [4, 5, 7, 15, 19, 20].

As WMNs become an increasingly popular replacement technology for last-mile connectivity to the home, they must adapt to support a new generation of streaming-media applications, such as Voice over IP (VoIP) and Video On-Demand (VOD) [2]. These applications require Quality of Service (QoS) guarantees in terms of minimum bandwidth and maximum end-to-end delay. However, most existing work on wireless meshes 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 application 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 has been studied in WMNs and MANETs [10, 17], our goal is to build a WMN routing protocol that provides "strong" QoS guarantees. By "strong" we mean that the routes discovered by our protocol will accept application requests for desired bandwidth and delay bounds for the flow, and either reject the flow if such constraints are not possible, or deliver an end-to-end flow that satisfies those performance bounds at the time of the request. If and when the route is disrupted by node or link failure, the 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 predicts the end-to-end delay of a flow with good accuracy, and show how such a mechanism 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 [13], but also helps discourage selfish "Free-riding" behavior [14]. 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 related work on wireless routing and QoS support. Second, Section 3 describes our network model and design objectives. Next, Section 4 describes the details of the QUORUM routing protocol. Finally, we describe simulation results in Section 5 and conclude in Section 6.

2. RELATED WORK

Despite significant work done on routing in wireless networks [8–10, 16, 17], very little work has been done to provide "strong" QoS guarantees in WMNs. A review of relevant literature shows that various approaches have been taken to provide QoS guarantees. Some advocate for a stateless approach based on a proactive routing protocol such as OLSR [3], while others have advocated maintaining state at intermediate nodes [6, 23, 24]. WMR [23] is a protocol that has been proposed to provide QoS enabled routing in WMNs. It is the result of modifying its MANET counterpart, AQOR [24] to

QShine'07 August 14–17, 2007, Vancouver, BC Canada

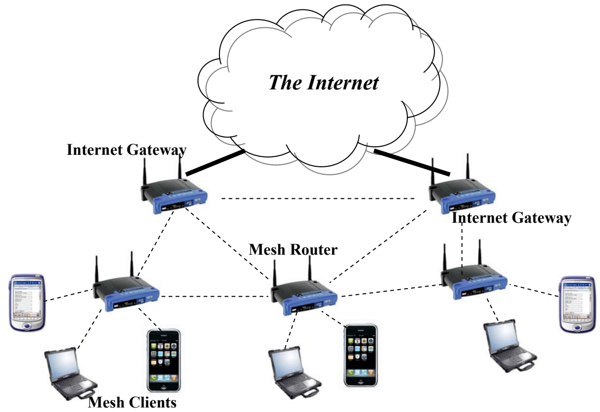


Figure 1: WMN Hybrid Network Architecture consists of Mesh Routers, Mesh Clients and Internet Gateways. A router has two interfaces, one for clients and one for other routers.

the wireless mesh context. Both AQOR and WMR cannot provide hard 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 [22] and clustering of end hosts and, use of orthogonal channels [21] to reduce the effect of interference. A very different method [12] suggests the use of a technique called Annealing used in statistical mechanics.

Although there has been some work in having solutions for providing QoS enabled routing, 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. Eventhough the problem of providing QoS guarantees based on multiple constraints has been shown to be NP Complete [12], it is also known that with suitable heuristics, a multi-constrained QoS routing algorithm can work in polynomial time [11].

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 Figure 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 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.

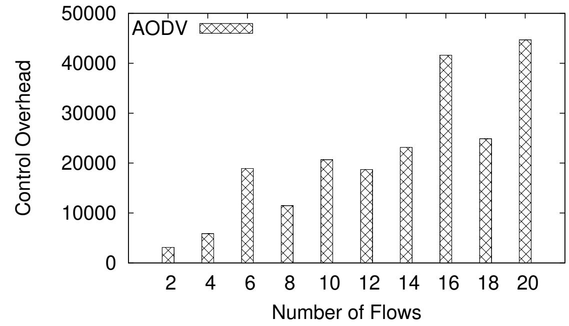


Figure 2: Control Overhead in a 50 node network. Overhead is calculated as the total number of `RREQ`s forwarded by all the nodes in the network.

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 [13]. Most routing protocols such as AODV [17] and WMR [23] rely on control (`RREQ`) packets to detect and establish end to end routes. However, these control packets have properties that differ significantly from data packets. For example, `RREQ` packets are typically much smaller than data packets, and also sent at significantly lower bit rates. As a result, nodes that are far away in distance might be able to send and receive bidirectional `RREQ` packets, but cannot send/receive high bit-rate data packets. The result is a highly "fragile" link that triggers link repairs, thereby incurring high control overhead for protocols such as AODV [13].

To demonstrate this, we quantify the control overhead in a network of 50 nodes (Figure 6) 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 Figure 2 shows, there is an astronomically high amount of control packet overhead in the network as the number of flows increases. Here we quantify overhead as the number of `RREQ`s 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 therefore are unlikely to experience the same levels of traffic delay and loss as data packets.

We perform an experiment to quantify the error introduced by the two estimation methods (`RREP` packets and hop count) to measure end-to-end delay. We select a small topology of 14 nodes and introduce a single 5-hop flow into the network. The Hopcount technique estimates the delay as the number of hops $\times$ the average per-hop

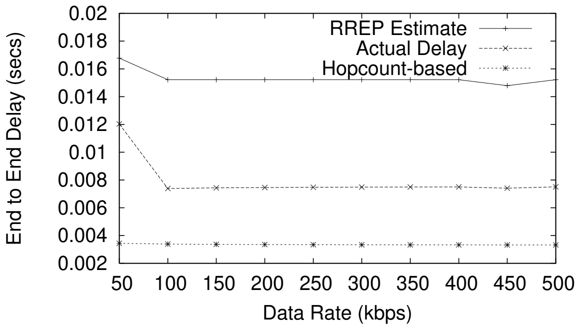


Figure 3: End to End Delay Estimation of the data packets. `RREP` **overestimates end-to-end delay because of inter-flow interference. The hopcount-based approach underestimates the delay since it does not account for intra-flow interference.**

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

Table 1: Structure of a flow table.

delay. As shown in Figure 3, we can see that both techniques introduce significant estimation errors: `RREP` Estimate 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. 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 in 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, and therefore incorrectly estimate the data packet delay. 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.

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 [24] and WMR [23]. 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).

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

Table 2: Structure of a neighbor table.

While QUORUM borrows the admission control scheme from AQOR and WMR, there are several key differences. The main drawback of WMR [23] 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. We 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. We 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. 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.

We illustrate two examples of this technique in Figure 4. Figure 4(a) 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 Figure 4(b), 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, we limit control packet flooding by by having nodes accept flooding messages only from "robust" neighbors (those with link quality $> 50\%$). This ensures that suboptimal paths are not discovered, addressing the communication gray zone problem. As can be seen later in the experimental section, this scheme reaps

(a) SRC and DEST under the same mesh router

(b) SRC and DEST under different mesh routers

Figure 4: 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.

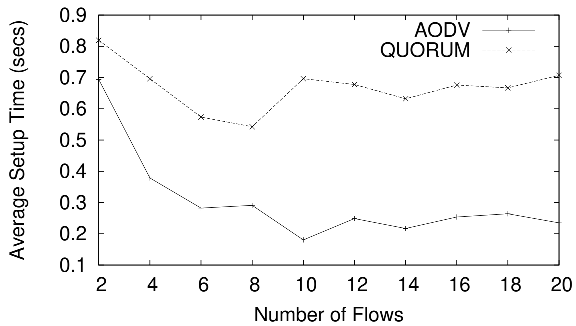


Figure 5: Route Setup Delay vs Number of Flows. Each point corresponds to the average setup time taken by the corresponding number of flows in the network. QUORUM takes relatively longer time due to the additional `DUMMY-RREP` **phase.**

huge benefits in terms of reduction of control overhead.

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. When the 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 real 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 tradeoff between control overhead and measurement accuracy, and is based on our experiments over a large set of data rates and path lengths. Those results are omitted for brevity.

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 sends data packets. Soft-state timers are included at both source and destination to take care of lost packets. If the delay exceeds the required limit, the source does a linear back-off and sends the `DUMMY` stream on a different route selected from its `RREP_TABLE`. Figure 5 shows the approximate route setup delay of QUORUM compared to AODV. For this experiment, 10 flows of 50 kbps were randomly chosen from the 50 node topology in Figure 6. 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 estimates.

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.

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. So the misbehaving 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 [24].

5. SIMULATION RESULTS

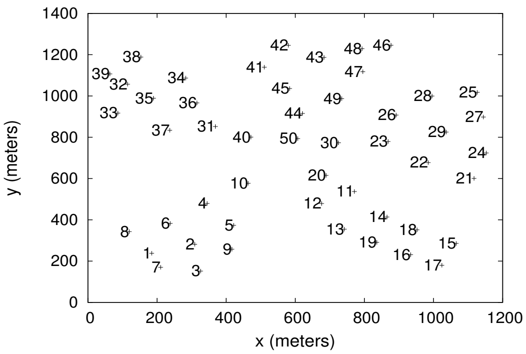


Figure 6: Our simulation topology of 45 mesh clients and 5 mesh routers (10, 20, 30, 40, 50). Each router (n) is responsible for 9 Mesh Clients ($n-9, n-8, \ldots, n-1$).

We perform detailed evaluation of our protocol using the Qualnet [18] simulator. Despite our best efforts, however, we were unable to obtain a Qualnet implementation of an existing QoS-routing protocol. While our experiments compare against AODV as the baseline, we are actively implementing a version of AQOR for comparison purposes on the Qualnet platform.

5.1 Experimental Setup

Our network topology, shown in Figure 6, 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 $1500m \times 1500m$. 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 seconds for network stabilization, flows are introduced gradually into the network. Each flow is alive for 10 minutes, and each simulation run lasts for 15 minutes. For our robustness computations, nodes broadcast `HELLO` packets every 200ms, 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 to guarantee 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 Figure 7(a), 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 Figure 6. 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 Figure 7(a).

As shown in Figure 7(a), 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 7(b) 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.

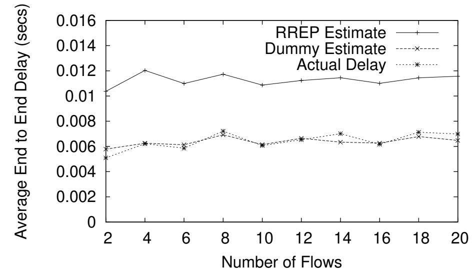


Figure 9: End to End Delay Estimation vs Number of Flows. Each flow requests a B_{min} of 50 kbps. We can observe that, in both the cases, QUORUM does a good job of estimating the actual delay fairly accurately.

5.3 Control Overhead

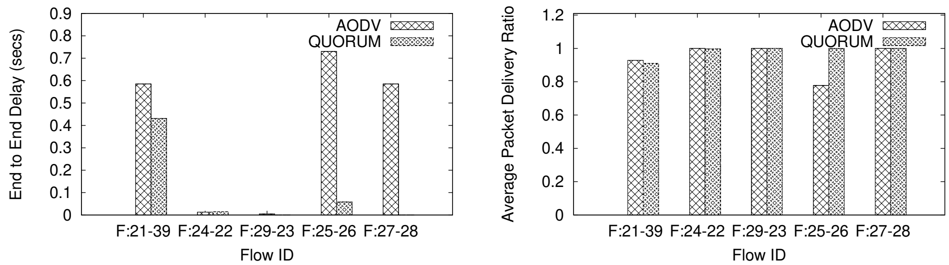
Control Overhead of the protocol is defined as the total number of control packets forwarded by all the nodes in the network. For QUORUM, we also include the `DUMMY` packets as an overhead induced by the protocol (in addition to `RREQ`s which is common in both). This experiment shows the benefits of 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 known. The control messages are required to estimate whether bandwidth and delay constraints are satisfied on any given path.

Figures 8(a) and 8(b) plot 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 8(a) shows that control overhead of QUORUM is lower than AODV by 30-35%.

Figure 8(b) shows a drastic increase in the control overhead of AODV with increase in the number of flows in the system. The main 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-discovery and hence higher control overhead. QUORUM refrains from accepting unstable routes by having a robustness threshold as described in Section 4.2.

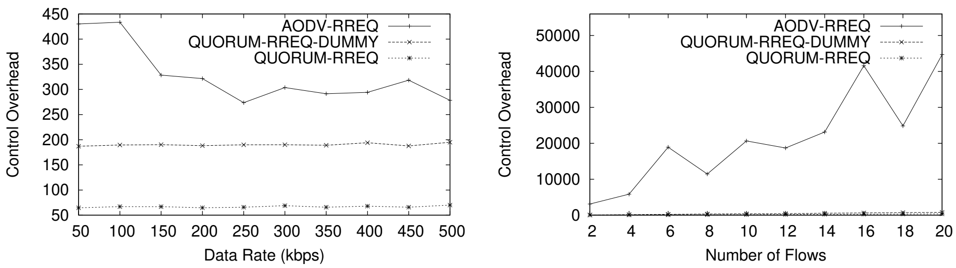
5.4 End to End Delay Estimation

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-RREP` phase in the estimation of end to end delay. The delay estimated by the `RREP` packets, delay esti-



(a) End to End Delay of the different flows (b) Packet Delivery Ratio of the different flows

Figure 7: In Figure 7(a) we can observe that F:$27-28$ is almost always rejected by QUORUM to provide relatively lower delays to the accepted flows. Figure 7(b) shows that QUORUM provides higher packet delivery ratio to all the accepted flows.



(a) Control Overhead vs Data Rate (b) Control Overhead vs Number of Flows

Figure 8: From Figure 8(a) we can see that control overhead in QUORUM is reduced by 40% compared to AODV due to intelligent routing. Figure 8(b) 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.

mated by the DUMMY−RREP 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 the behavior is similar in both the cases, we only plot the graph for varying flows experiment. As shown in Figure 9, delay estimated by the RREQ−RREP phase differs from the actual delay by a huge margin. By having the DUMMY stream emulate the data packets, QUORUM is able to accurately estimate the real delay.

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 5 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 10 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 cooperate. The misbehavior model is as described in Section 4.4. In this experiment, the average throughput of "bad" and "good" flows

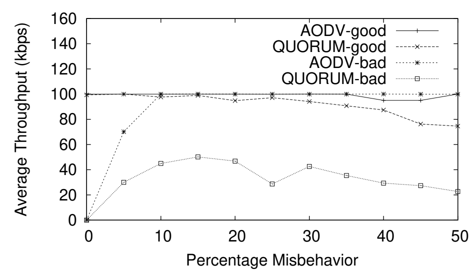
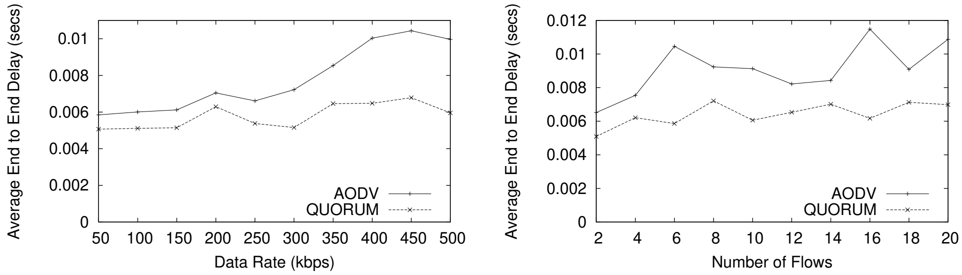


Figure 11: We plot throughput values for both "good" and "bad" flows in AODV and QUORUM. QUORUM clearly discourages selfish behavior by denying them network bandwidth.

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 clients can misbehave. 10 flows of 50 kbps data rate are randomly picked for each run.

From Figure 11, we se 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 be-



(a) Average End to End Delay vs Data Rate (b) Average End to End Delay vs Number of Flows

Figure 10: 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 Figure 10(b) the average delay of the flows in QUORUM is 33% lower than that experienced in AODV.

cause 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 for 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.

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).

7. REFERENCES

[1] Akyildiz, I., Wang, X., and Wang, W. Wireless Mesh Networks: A Survey. *Computer Networks* (2005), 445–487.

[2] Allen, M., Zhao, B. Y., and Wolski, R. Deploying video-on-demand services on cable networks. In *Proc. of ICDCS* (Toronto, Canada, June 2007).

[3] Badis, H., Gawedzki, I., and Al Agha, K. QoS Routing in Ad Hoc Networks using QOLSR with no need of Explicit Reservation. In *Proc. of VTC* (Sept 2004).

[4] Bay Area Wireless User Group. `http://www.bawug.org/`.

[5] Bicket, J., Aguayo, D., Biswas, S., and Morris, R. Architecture and evaluation of an unplanned 802.11b mesh network. In *Proc. of MobiCom* (New York, NY, 2005).

[6] Chen, S., and Nahrstedt, K. A Distributed Quality-of-Service Routing in Ad-Hoc Networks. *IEEE JSAC 17*, 8 (August 1999).

[7] Champaign-Urbana Community Wireless Network. `http://www.cuwireless.net/`.

[8] Draves, R., Padhye, J., and Zill, B. Comparison of Routing Metrics for Static Multi-hop Wireless Networks. In *Proc. of SIGCOMM* (Portland, OR, 2004).

[9] Draves, R., Padhye, J., and Zill, B. Routing in Multi-radio, Multi-hop Wireless Mesh Networks. In *Proc. of MobiCom* (New York, NY, 2004).

[10] Johnson, D. B., Maltz, D. A., and Broch, J. DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks. *Ad Hoc Networking 5* (2001), 139–172.

[11] Kuipers, F. A., and Mieghem, P. F. A. V. Conditions that Impact the Complexity of QoS Routing. *IEEE/ACM Trans. on Networking 13*, 4 (2005), 717–730.

[12] Liu, L., and Feng, G. Mean Field Network Based QoS Routing Scheme for Wireless Mesh Networks. In *Proc. of WiCOM* (Sept 2005).

[13] Lundgren, H., Nordstrom, E., and Tschudin, C. The Gray Zone Problem in IEEE 802.11b based Ad hoc Networks. *MC2R 6*, 3 (2002), 104–105.

[14] Mahajan, R., et al. Sustaining Cooperation in Multi-Hop Wireless Networks. In *Proc. of NSDI* (2005).

[15] Mesh Dynamics Structured Mesh Networking for Mobile Data, Video and Voice. `http://meshdynamics.com/documents/MD4000_BROCHURE.pdf`.

[16] Nandiraju, N. S., and Agrawal, D. P. Multipath Routing in Wireless Mesh Networks. In *Proc. of MASS* (2006).

[17] Perkins, C., and Royer, E. Ad hoc on-demand distance vector routing. In *Proc. of WMCSA* (Feb. 1999).

[18] Qualnet Simulator. `http://scalable-networks.com`.

[19] Raman, B., and Chebrolu, K. Experiences in using WiFi for rural internet in India. *IEEE Communications Magazine 45*, 1 (Jan 2007), 104–110.

[20] Tropos MetroMesh Architecture Overview. `http://www.tropos.com/pdf/metromesh_datasheet.pdf`.

[21] Tung, T., Jia, Z., and Walrand, J. A Practical Approach to QoS Routing for Wireless Networks. In *Proc. of WIOPT* (2005).

[22] Wang, H., and Yow, K. C. QoS routing in Multi-Channel Multihop Wireless Networks with Infrastructure Support. In *InterSense* (New York, NY, 2006).

[23] Xue, Q., and Ganz, A. QoS Routing for Mesh-Based Wireless LANs. In *International Journal of Wirelss Information Networks* (July 2002), vol. 9.

[24] Xue, Q., and Ganz, A. Ad hoc QoS On-demand Routing (AQOR) in Mobile Ad hoc Networks. *J. Parallel and Distributed Computing* (2003), 154–165.