This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

A QoS-Oriented Distributed Routing Protocol for Hybrid Wireless Networks

Ze Li, *Student Member, IEEE,* Haiying Shen*, *Member, IEEE*

Abstract—As wireless communication gains popularity, significant research has been devoted to supporting real-time transmission with stringent Quality of Service (QoS) requirements for wireless applications. At the same time, a wireless hybrid network that integrates a mobile wireless ad hoc network (MANET) and a wireless infrastructure network has been proven to be a better alternative for the next generation wireless networks. By directly adopting resource reservation-based QoS routing for MANETs, hybrids networks inherit invalid reservation and race condition problems in MANETs. How to guarantee the QoS in hybrid networks remains an open problem. In this paper, we propose a QoS-Oriented Distributed routing protocol (QOD) to enhance the QoS support capability of hybrid networks. Taking advantage of fewer transmission hops and anycast transmission features of the hybrid networks, QOD transforms the packet routing problem to a resource scheduling problem. QOD incorporates five algorithms: (1) a QoS-guaranteed neighbor selection algorithm to meet the transmission delay requirement, (2) a distributed packet scheduling algorithm to further reduce transmission delay, (3) a mobility-based segment resizing algorithm that adaptively adjusts segment size according to node mobility in order to reduce transmission time, (4) a traffic redundant elimination algorithm to increase the transmission throughput, and (5) a data redundancy elimination based transmission algorithm to eliminate the redundant data to further improve the transmission QoS. Analytical and simulation results based on the random way-point model and the real human mobility model show that QOD can provide high QoS performance in terms of overhead, transmission delay, mobility-resilience and scalability.

Index Terms—Hybrid wireless networks, Multi-hop cellular networks, Routing algorithms, Quality of service

1 Introduction

The rapid development of wireless networks has stimulated numerous wireless applications that have been used in wide areas such as commerce, emergency services, military, education and entertainment. The number of WiFi capable mobile devices including laptops and handheld devices (e.g. smartphone and tablet PC) has been increasing rapidly. For example, the number of wireless Internet users has tripled world-wide in the last three years, and the number of smartphone users in US has increased from 60.2 million in 2010 to 90.1 million in 2011, and will reach around 120 million by 2013 [1]. Nowadays, people wish to watch videos, play games, watch TV and make long-distance conferencing via wireless mobile devices "on the go." Therefore, video streaming applications such as Qik [2], Flixwagon [3] and FaceTime [4] on the infrastructure wireless networks have received increasing attention recently. These applications use an infrastructure to directly connect mobile users for video watching or interaction in real time. The widespread use of wireless and mobile devices and the increasing demand for mobile multimedia streaming services are leading to a promising near future where wireless multimedia services (e.g., mobile gaming, online TV, and on-line conferences) are widely deployed. The emergence and the envisioned future of real-time and multimedia applications have stimulated the need of high Quality of Service (QoS) support in wireless and mobile networking environments [5]. The *QoS support* reduces end-to-end transmission delay and enhances throughput to guarantee the seamless communication between mobile devices and wireless infrastructures.

At the same time, hybrid wireless networks (i.e., multi-hop cellular networks) have been proven to be a better network structure for the next generation wireless networks [6–9], and can help to tackle the stringent end-to-end QoS requirements of different applications. Hybrid networks synergistically combine infrastructure networks and MANETs to leverage each other. Specifically, infrastructure networks improve the scalability of MANETs, while MANETs automatically establish self-organizing networks, extending the coverage of the infrastructure networks. In a vehicle opportunistic access network (an instance of hybrid networks), people in vehicles need to upload or download videos from remote Internet servers through access points (APs) (i.e., base stations) spreading out in a city. Since it is unlikely that the base stations cover the entire city to maintain sufficiently strong signal everywhere to support an application requiring high link rates, the vehicles themselves can form a MANET to extend the coverage of the base stations, providing continuous network connections.

How to guarantee the QoS in hybrid wireless networks with high mobility and fluctuating bandwidth still remains an open question. In the infrastructure wireless networks, QoS provision (e.g. Intserv [10], RSVP [11]) has been proposed for QoS routing, which often requires node negotiation, admission control, resource reservation, and priority scheduling of packets [12]. However, it is more difficult to guarantee QoS in MANETs due to their unique features including user mobility, channel variance errors and limited bandwidth. Thus, attempts to directly adapt the QoS solutions for infrastructure networks to MANETs generally do not have great success [13]. Numerous reservation-based QoS routing protocols have been proposed for MANETs [14–22] that create routes formed by nodes and links that reserve their resources to fulfill QoS requirements. Although these protocols can increase the

- * Corresponding Author. Email: shenh@clemson.edu; Phone: (864) 656 5931; Fax: (864) 656 5910.
- The authors are with the Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, 29634. E-mail: {zel, shenh}@clemson.edu

Digital Object Indentifier 10.1109/TMC.2012.258

QoS of the MANETs to a certain extent, they suffer from invalid reservation and race condition problems [12]. Invalid reservation problem means that the reserved resources become useless if the data transmission path between a source node and a destination node breaks. Race condition problem means a double allocation of the same resource to two different QoS paths.

However, little effort has been devoted to support QoS routing in hybrid networks. Most of the current works in hybrid networks [23–27] focus on increasing network capacity or routing reliability but cannot provide QoS-guaranteed services. Direct adoption of the reservation-based QoS routing protocols of MANETs into hybrid networks inherits the invalid reservation and race condition problems.

In order to enhance the QoS support capability of hybrid networks, in this paper, we propose a QoS-Oriented Distributed routing protocol (QOD). Usually, a hybrid network has widespread base stations. The data transmission in hybrid networks has two features. First, an AP can be a source or a destination to any mobile node. Second, the number of transmission hops between a mobile node and an AP is small. The first feature allows a stream to have anycast transmission along multiple transmission paths to its destination through base stations, and the second feature enables a source node to connect to an AP through an intermediate node. Taking full advantage of the two features, QOD transforms the packet routing problem into a dynamic resource scheduling problem. Specifically, in QOD, if a source node is not within the transmission range of the AP, a source node selects nearby neighbors that can provide QoS services to forward its packets to base stations in a distributed manner. The source node schedules the packet streams to neighbors based on their queuing condition, channel condition and mobility, aiming to reduce transmission time and increase network capacity. The neighbors then forward packets to base stations, which further forward packets to the destination. In this paper, we focus on the neighbor node selection for QoS-guaranteed transmission. QOD is the first work for QoS routing in hybrid networks. This paper makes five contributions.

- *QoS-guaranteed neighbor selection algorithm.* The algorithm selects qualified neighbors and employs deadline-driven scheduling mechanism to guarantee QoS routing.
- *Distributed packet scheduling algorithm.* After qualified neighbors are identified, this algorithm schedules packet routing. It assigns earlier generated packets to forwarders with higher queuing delays, while assigns more recently generated packets to forwarders with lower queuing delays to reduce total transmission delay.
- *Mobility-based segment resizing algorithm.* The source node adaptively resizes each packet in its packet stream for each neighbor node according to the neighbor's mobility in order to increase the scheduling feasibility of the packets from the source node.
- *Soft-deadline based forwarding scheduling algorithm.* In this algorithm, an intermediate node first forwards the packet with the least time allowed to wait before being forwarded out to achieve fairness in packet forwarding.
- *Data redundancy elimination based transmission.* Due to the broadcasting feature of the wireless networks, the APs and mobile nodes can overhear and cache packets. This algorithm eliminates the redundant data to improve the QoS of the packet transmission.

2 THE QOD PROTOCOL

2.1 Network and Service Models

We consider a hybrid wireless network with an arbitrary number of base stations spreading over the network. N mobile nodes are moving around in the network. Each node n_i $(1 \leq i \leq N)$ uses IEEE 802.11 interface with the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol [28]. Since a hybrid network where nodes are equipped with multi-interfaces that transmit packets through multi-channels generate much less interference than a hybrid network where nodes are equipped with a single WiFi interface, we assume that each node is equipped with a single WiFi interface in order to deal with a more difficult problem. Therefore, the base stations considered in this paper are access points (APs). The WiFi interface enables nodes to communicate with both APs and mobile nodes. For example, in a University campus, normally only buildings have APs. Therefore, people that do not have WiFi access but close to buildings can use two-hop relay transmissions to connect to the APs in the buildings. Feeney *et al.* [29] considered the similar scenario in his work.

We use R_i and $R_i^{'}$ to denote the packet transmission range and transmission interference range of node n_i, respectively. We use $d_{i,j}$ to denote the distance between nodes n_i and n_j. A packet transmission from n_i to n_j is successful if both conditions below are satisfied [30]: (1) $d_{i,j} \leq R_i$, and (2) any node n_k satisfying $d_{k,j} \leq R_k'$ is not transmitting packets, where $0 < k < N$ and $k \neq j$. Table 1 lists the symbols used in this paper for reference.

TABLE 1: List of symbols.

N	# of network nodes	T_p	Transmission time of packet p
m	# of neighbors of a node	T_a	Packet arrival interval
n_i	Node i	T_{QoS}	Delay QoS requirement
R_i	Transmission range of n_i	T_w	Queuing delay
R_i'	Interference range of n_i	$d_{i,j}$	Distance between n_i and n_j
C_i	Link capacity of node n_i	$\widetilde{T}_{U_s}$	Threshold of space utility
$\Re_I$	Interference region	D_p	Deadline of packet p
$\Re_T$	Transmission region	S_p	The size of packet p
$\phi_\Re$	Node density in region $\Re$	$\overline{T}$	Utility update interval
$S_\Re$	Area size of region $\Re$	U_c	Channel utility
U_s	Space utility	U_{as}	Available space utility
W_i	Bandwidth of n_i	λ	Arrival rate
v_i	Moving speed of n_i	$\widetilde{T}_{U_s}$	Space utility threshold

The QoS requirements mainly include end-to-end delay bound, which is essential for many applications with stringent real-time requirement. While throughput guarantee is also important, it is automatically guaranteed by bounding the transmission delay for a certain amount of packets [31]. The source node conducts admission control to check whether there are enough resources to satisfy the requirements of QoS of the packet stream. Figure 1 shows the network model of a hybrid network. For example, when a source node n_1 wants to upload files to an Internet server through APs, it can choose to send packets to the APs directly by itself or require its neighbor nodes n_2, n_3 or n_4 to assist the packet transmission.

We assume that queuing occurs only at the output ports of the mobile nodes [32]. After a mobile node generates the packets, it first tries to transmit the packets to its nearby APs that can guarantee the QoS requirements. If it fails (e.g. out of the transmission range of APs or in a hot/dead spot), it relies on its neighbors that can guarantee the QoS requirements for relaying packets to APs. Relaying for a packet stream can be modeled as a process, in which packets from a source node traverse a number of queuing servers to some APs [31]. In this

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

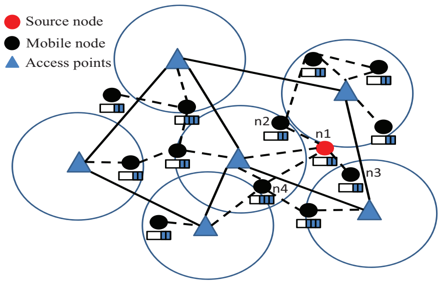


Fig. 1: The network model of the hybrid networks.

model, the problem of how to guarantee QoS routing can be transformed to the problem of how to schedule the neighbor resources between nodes to ensure QoS of packet routing.

2.2 An Overview of the QOD Protocol

Scheduling feasibility is the ability of a node to guarantee a packet to arrive at its destination within QoS requirements. As mentioned, when the QoS of the direct transmission between a source node and an AP cannot be guaranteed, the source node sends a request message to its neighbor nodes. After receiving a forward request from a source node, a neighbor node n_i with space utility less than a threshold replies the source node. The reply message contains information about available resources for checking packet scheduling feasibility (Section (2.4)), packet arrival interval T_a, transmission delay $T_{I\rightarrow D}$, and packet deadline D_p of the packets in each flow being forwarded by the neighbor for queuing delay estimation and distributed packet scheduling (Section 2.5) and the node's mobility speed for determining packet size (Section (2.6)). Based on this information, the source node chooses the replied neighbors that can guarantee the delay QoS of packet transmission to APs. The selected neighbor nodes periodically report their statuses to the source node, which ensures their scheduling feasibility and locally schedules the packet stream to them. The individual packets are forwarded to the neighbor nodes that are scheduling feasible in a round-robin fashion from a longer-delayed node to a shorter-delayed node, aiming to reduce the entire packet transmission delay. Algorithm 1 shows the pseudo-code for the QOD routing protocol executed by each node.

Algorithm 1 Pseudo-code for the QOD routing protocol executed by a source node.

if receive a packet forwarding request from a source node **then**
  if this.SpaceUtility<threshold **then**
    Reply to the source node.
  end if
end if
if receive forwarding request replies for neighbor nodes **then**
  Determine the packet size $S_p(i)$ to each neighbor i based on Equation (5).
  Estimate the queuing delay T_w for the packet for each neighbor based on Equation (4).
  Determine the qualified neighbors that can satisfy the deadline requirements based on T_w
  Sort the qualified nodes in descending order of T_w
  Allocate workload rate A_i for each node based on Equation (3)
  for each intermediate node n_i in the sorted list **do**
    Send packets to n_i with transmission interval $\frac{S_p(i)}{A_i}$.
  end for
end if

The packets travel from different APs, which may lead to different packet transmission delay, resulting in a jitter at the receiver side. The jitter problem can be solved by using token buckets mechanism [33] at the destination APs to shape the traffic flows. This technique is orthogonal to our study in this paper and its details are beyond the scope of this paper.

Before introducing the details of QOD in the system, we justify that QOD is feasible to be used in a network with the IEEE 802.11 protocol in Section 2.3. We then present the details of QoS by answering the following questions in QoS routing in hybrid networks.

(1) How to choose qualified neighbors for packet forwarding? (Section 2.4)
(2) How to schedule the packets to the qualified neighbor nodes? (Section 2.5)
(3) How to ensure the QoS transmission in a highly dynamic situation? (Section 2.6)
(4) How to schedule the packets in the relay node in forwarding to destinations? (Section 2.7)
(5) How to reduce the data redundancy in transmission to further enhance QoS? (Section 2.8)

2.3 Applicability of the QOD Distributed Routing Algorithm

The QOD distributed routing algorithm is developed based on the assumption that the neighboring nodes in the network have different channel utilities and workloads using IEEE 802.11 protocol. Otherwise, there is no need for packet scheduling in routing, since all neighbors produce comparative delay for packet forwarding. Therefore, we analyze the difference in node channel utilities and workloads in a network with IEEE 802.11 protocol in order to see whether the assumption holds true in practice.

2.3.1 Theoretical Analysis of Channel Utility and Workload Differences

In order to avoid medium access contention and hidden terminal problem, IEEE 802.11 uses the CSMA/CA protocol as MAC access control protocol. Before a node sends out packets, it sends a Request To Send (RTS) message to the next hop node indicating the duration time of the subsequent transmission. The destination node responds with a Clear To Send (CTS) message to establish a connection with the source node. The neighbor nodes overhearing RTS and/or CTS set their Virtual Carrier Sense indicator (i.e., Network Allocation Vector (NAV)) to the message's transmission duration time, so that it can avoid transmitting data into the channel within the time duration. We define *channel utility* as the fraction of time a channel is busy over a unit time. Assume $\overline{T}$ is a constant time interval used for channel utility updating, by referring to NAV and update interval $\overline{T}$, each node n_i can statistically calculate its channel utility by $U_c(i) = \frac{T_{NAV}(i)}{\overline{T}}$, where $T_{NAV}(i)$ is the number of time units that n_i is interfered that is recorded by NAV. The available bandwidth for n_i is $W_i = (1 - U_c(i)) \cdot C_i$, where C_i is the transmission link capacity of node n_i.

Figure 2 shows a graph to demonstrate the interference between two neighboring nodes n_i and n_j. The solid circles around n_i and n_j denote their packet transmission ranges, and the dotted circles denote their interference ranges (sensing ranges). We use $\Re_{I(n_i)}$ to represent the interference region of n_i that is not overlapped with that of n_j, use $\Re_{I(n_j)}$ to represent the interference region of n_j that is not overlapped with that of n_i, and

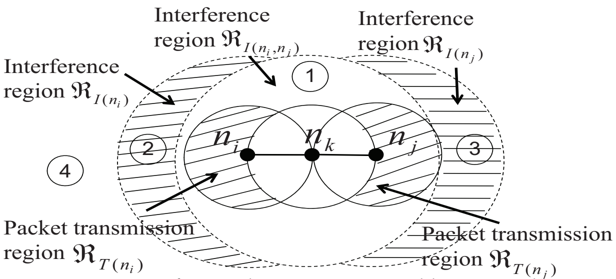


Fig. 2: Interference between two neighboring nodes.

use $\Re_{I(n_i,n_j)}$ to denote the overlapped region of the interference regions of n_i and n_j. We first analyze whether the nodes in $\Re_{I(n_i)}$, $\Re_{I(n_j)}$ and $\Re_{I(n_i,n_j)}$ have different channel utilities. We see that when n_j is communicating with n_k, the signal will not be received by the nodes in $\Re_{I(n_i)}$, i.e., they can receive or send packets with other nodes at the same time with no interference from n_j. Similarly, when n_i is communicating with n_k, the nodes in $\Re_{I(n_j)}$ can receive or send packets with other nodes at the same time with no interference from n_i. Thus, the nodes in $\Re_{I(n_i)}$ are independent from n_j and the nodes in $\Re_{I(n_i)}$ are independent from n_i. As a result, the differences between the time durations of transmitting packets of node n_i and node n_j lead to different channel utilities of the nodes in $\Re_{I(n_i)}$, $\Re_{I(n_j)}$ and $\Re_{I(n_i,n_j)}$.

Proposition 2.1: The different data transmission amounts of two neighboring nodes lead to different channel utilities of the nodes in their common interference region and individual independent interference regions.

We then analyze the workload difference between two neighboring nodes n_i and n_j. We define the workload of a node as the accumulated number of packets received by the node through the entire simulation period. The workloads in n_i and n_j are determined by the packets received by n_i and n_j from the nodes in $\Re_{T(n_i)}$ and $\Re_{T(n_j)}$, respectively, where $\Re_{T(n_i)}$ denotes the packet transmission region. We use $\phi_{\Re}$ to represent the density of nodes in area $\Re$. Suppose each node's transmission range is R, each node's interference range is $R' = \alpha \cdot R$ ($\alpha > 1$), and the distance between two neighboring nodes is $d = \beta \cdot R$ ($\beta \leq 1$), we can get Lemma 2.1.

Lemma 2.1: In the different interference regions of two neighboring nodes n_i and n_j, the ratio of the number of nodes in $\Re_{I(n_i)}$, $\Re_{I(n_i,n_j)}$ and $\Re_{I(n_j)}$ that have different channel utilities equals $\phi_{\Re_{I(n_i)}} : \eta \cdot \phi_{\Re_{I(n_i,n_j)}} : \phi_{\Re_{I(n_j)}}$, where $\eta = 2.46$.

Proof: Let $S_{\Re}$ denote the size of $\Re$ and suppose $\beta \approx 1$. From Figure 2, we can get

$$S_{\Re_{I(n_i)}} = S_{\Re_{I(n_j)}} = (\pi - 2\arccos(\frac{\beta}{2\alpha}))\alpha^2 R^2 + \frac{\beta R^2\sqrt{4\alpha^2-\beta^2}}{2}$$

and
$$S_{\Re_{I(n_i,n_j)}} = \pi \cdot \alpha^2 \cdot R^2 - S_{\Re_{I(n_i)}}.$$

Therefore, the ratio of the number of nodes in $\Re_{I(n_i)}$, $\Re_{I(n_i,n_j)}$ and $\Re_{I(n_j)}$ is $\phi_{\Re_{I(n_i)}} : \eta \cdot \phi_{\Re_{I(n_i,n_j)}} : \phi_{\Re_{I(n_j)}}$.

Since $\alpha \approx 2$ according to the specification of IEEE 802.11,

$$\eta = \frac{S_{\Re_{I(n_i,n_j)}}}{S_{\Re_{I(n_i)}}} = \frac{2\arccos(\frac{\beta}{2\alpha}) \cdot \alpha^2 R^2 - \frac{\beta R^2}{2}\sqrt{4\alpha^2-\beta^2}}{(\pi - 2\arccos(\frac{\beta}{2\alpha})) \cdot \alpha^2 R^2 + \frac{\beta R^2}{2}\sqrt{4\alpha^2-\beta^2}} \approx 2.46.$$
□

Theorem 2.1: The number of different channel utility values n_u in the system with N nodes are bounded by $\Theta(N) < n_u < \Theta(2^N)$.

Proof: According to Lemma 2.1, there are 3 different utility values in the areas of 1, 2 and 3 indicated in Figure 2. In addition, there is another channel utility value in the area of 4 in Figure 2. Every time when we add a new transmitting node into the system, the number of different utility values increases at least linearly because at least one different utility value is introduced into the system, and at most exponentially because at most the same number of existing utility values are introduced into the system. Therefore, for a system with N nodes, the number of different utilities N_u is bounded by $\Theta(N) < n_u < \Theta(2^N)$. □

Suppose the probability of node n_i receiving a packet from nodes in $\Re_{T(n_i)}$ and n_j receiving a packet from $\Re_{T(n_j)}$ is q. The packet size is S_p. Then, we can retrieve

Theorem 2.2: The difference of the workload in node n_i and node n_j is

$$(\phi_{T(n_i)} - \phi_{T(n_j)})((\pi - 2\arccos(\frac{\beta}{2}))R^2 + \frac{\beta R^2}{2}\sqrt{4-\beta^2})q \cdot S_p.$$

Proof: Based on the geometric calculation, we can get

$$S_{\Re_{T(n_i)}} = S_{\Re_{T(n_j)}} = (\pi - 2\arccos(\frac{\beta}{2}))R^2 + \frac{\beta R^2}{2}\sqrt{4-\beta^2}.$$

The difference of the workload in node n_i and node n_j is affected by the traffic from $\Re_{T(n_i)}$ and $\Re_{T(n_j)}$, which is equal to $(\phi_{T(n_i)} - \phi_{T(n_j)}) \cdot S_{\Re_{T(n_i)}} \cdot q \cdot S_p$. □

The theoretical analysis show that if the source nodes are independent and identically distributed in a system with random packet generation rate, the nodes with the IEEE 802.11 protocol present diversified channel utilities and workloads, which is suitable for distributed resource scheduling. In Section 6 in the supplemental file, we present our experimental evaluation of channel utility and workload differences.

2.4 QoS-Guaranteed Neighbor Selection Algorithm

Since short delay is the major real-time QoS requirement for traffic transmission, QOD incorporates the Earliest Deadline First scheduling algorithm (EDF) [34], which is a deadline driven scheduling algorithm for data traffic scheduling in intermediate nodes. In this algorithm, an intermediate node assigns the highest priority to the packet with the closest deadline and forwards the packet with the highest priority first. Let us use $S_p(i)$ to denote the size of the packet steam from node n_i, use W_i to denote the bandwidth of node i and $T_a(i)$ to denote the packet arrival interval from node n_i.

Theorem 2.3: The QoS of the packets going through node n_i can be satisfied if $\frac{S_p(1)}{T_a(1)} + \frac{S_p(j)}{T_a(j)} + ... + \frac{S_p(m)}{T_a(m)} \leq W_i$.

Proof: Liu *et. al* [34] proposed a job scheduling model, where a task consists of a number of jobs. The authors proved that for a given set of $\overline{m}$ jobs for an operating system, the deadline-driven scheduling algorithm is feasible for the job scheduling iff

$$\frac{T_{cp}(1)}{T_g(1)} + \frac{T_{cp}(2)}{T_g(2)} + \frac{T_{cp}(j)}{T_g(j)} + ... + \frac{T_{cp}(\overline{m})}{T_g(\overline{m})} \leq 1, \quad (1)$$

where $T_g(j)$ denotes the job arrival interval time period and $T_{cp}(j)$ denotes the job computing time of task j and 1 is the CPU utility when the CPU is busy all the time.

Recall that *space utility* $U_s(i)$ is the fraction of time a node n_i is busy with packet forwarding over a unit time. In a communication network, the transmission time of a packet in packet stream from node n_j can be regarded as the computing time $T_{cp}(j)$ of a job from task j, the packet arrival interval T_a can be regarded as T_g, and the CPU utility can be regarded as node *space utility* in the job scheduling model. Then, Equation (2) is reduced to

$$U_s(i) = \frac{S_p(1)/W_i}{T_a(1)} + \frac{S_p(2)/W_i}{T_a(2)} + \frac{S_p(j)/W_i}{T_a(j)} + ... + \frac{S_p(m)/W_i}{T_a(m)} \leq 1$$

$$\Rightarrow \frac{S_p(1)}{T_a(1)} + \frac{S_p(j)}{T_a(j)} + ... + \frac{S_p(m)}{T_a(m)} \leq W_i, \quad (2)$$

where $W_i = (1 - U_c(i)) \cdot C_i$, $S_p(j)$ is the size of the packet steam from node n_j. □

Equation (2) indicates that the scheduling feasibility of a node is affected by packet size S_p, the number of packet streams from m neighbors and its bandwidth W_i.

Similar to the Random Early Detection (RED) algorithm [35], in which a queue length threshold is set to avoid queuing congestion, we set up a space utility threshold $\widetilde{T}_{U_s}$ for each node as a safety line to make the queue scheduling feasible. We use $U_{as}(i)$ to denote the available space utility and $U_{as}(i) = \widetilde{T}_{U_s} - U_s(i)$. In QOD, after receiving a forward request from a source node, an intermediate node n_i with space utility less than threshold $\widetilde{T}_{U_s}$ replies the source node. The replied node n_i informs the source node about its available workload rate $U_{as}(i) * W_i$, and the necessary information to calculate the queuing delay of the packets from the source node. The source node selects the replied neighbor nodes that can meet its QoS deadline for packet forwarding based on the calculated queuing delay. We will explain the details of this step in Section 2.5.

After the source node determines the N_q nodes that can satisfy the deadline requirement of the source node, the source node needs to distribute its packets to the N_q nodes based on their available workload rate $U_{as}(i) * W_i$ to make the scheduling feasible in each of the neighbor nodes. Then, the problem can be modeled as a linear programming process. Suppose the packet generating rate of the source node is $W_g$kb/s, the available workload rate of the intermediate node i is $U_{as}(i) * W_i$, and the workload rate allocation from source node to immediate node i is $A_i = \frac{S_p(i)}{T_a(i)}$, where $0 < i < n$. Then, we need to solve the following equations to get an allocation set $\mathbf{A}$:

$$\mathbf{A} = \begin{cases} W_g = \sum_{i=1}^{N_q} A_i \\ A_i \leq U_{as}(i) * W_i \end{cases} \quad (3)$$

Any results satisfy the Equation (3) can be used by the source node. If the equation cannot be solved, which means the QoS of the source node cannot be satisfied, then the source node stops generating packets based on the admission control policy. How to determine the packet size $S_p(i)$ for n_i will be introduced in Section 2.6. Based on $S_p(i)$, the source node can calculate the packet arrival interval for node n_i: $T_a(i) = \frac{S_p(i)}{A_i}$.

For example, suppose the bandwidth W_i of the intermediate node n_i is 70kb/s, the threshold of the workload is 80% of the overall space utility, which is 56kb/s. Node n_i schedules the packet traffic from three different source nodes n_1, n_2 and n_3 periodically. The packet size of traffic from n_1, n_2 and n_3 are 1kb, 10kb and 20kb with arrival interval 0.1s, 0.5s and 1s, respectively. Then, $\frac{S_1}{T_1} + \frac{S_2}{T_2} + \frac{S_3}{T_3}$=50kb/s. When another node n_4 sends a request to the intermediate n_i, n_i checks its own available workload rate and replies to the source node its available workload rate 6kb/s. If n_4 accepts the reply and sends 20kb/s traffic to the intermediate node n_i, n_i will reject the request and inform n_4 to reduce the traffic to 6kb/s. Once the bandwidth of the intermediate node drops to 60kb/s because of the interference, n_i's overall space utility is reduced to 48kb/s. Then, n_i informs node n_3 under scheduling that has the largest $\frac{S_p(j)}{T_a(j)}$= 20kb/s to change its traffic to 18kb/s.

2.5 Distributed Packet Scheduling Algorithm

Section 2.4 solves the problem of how to select intermediate nodes that can guarantee the QoS of the packet transmission and how a source node assigns traffic to the intermediate nodes to ensure their scheduling feasibility. In order to further reduce the stream transmission time, a distributed packet scheduling algorithm is proposed for packet routing. This algorithm assigns earlier generated packets to forwarders with higher queuing delays and scheduling feasibility, while assigns more recently generated packets to forwarders with lower queuing delays and scheduling feasibility, so that the transmission delay of an entire packet stream can be reduced.

We use t to denote the time when a packet is generated, and use T_{QoS} to denote the delay QoS requirement. Let W_S and W_I denote the bandwidth of a source node and an intermediate node respectively, we use $T_{S\to I} = \frac{S_p}{W_S}$ to denote the transmission delay between a source node and an intermediate node, and $T_{I\to D} = \frac{S_p}{W_I}$ to denote the transmission delay between an intermediate node and an AP. Let T_w denote the packet queuing time and $T_w(i)$ denote the packet queuing time of n_i. Then, as Figure 3 shows, the queuing delay requirement is calculated as $T_w < T_{QoS} - T_{S\to I} - T_{I\to D}$. As T_{QoS}, $T_{S\to I}$ and $T_{I\to D}$ are already known, the source node needs to calculate T_w of each intermediate node to select intermediate nodes that can send its packets by the deadline, i.e., that can satisfy $T_w < T_{QoS} - T_{S\to I} - T_{I\to D}$.

Below we introduce how to calculate T_w. Recall that QOD incorporates the EDF [34], in which an intermediate node assigns the highest priority to the packet with the closest deadline and forwards the packet with the highest priority first. An intermediate node can determine the priorities of its packets based on their deadlines D_p. A packet with a smaller priority value x has a higher priority. Before introducing the details of the distributed packet scheduling algorithm, we explain how to estimate the queuing time $T_w^{(x)}$ of a packet with priority x. It is estimated by

$$T_w^{(x)} = \sum_{j=1}^{x-1} (T_{I\to D}^{(j)} \cdot \lceil T_w^{(x)}/T_a^{(j)} \rceil) \ (0 < j < x), \quad (4)$$

where x denotes a packet with the xth priority in the queue, and $T_{I\to D}^{(j)}$ and $T_a^{(j)}$ respectively denote the transmission delay and arrival interval of a packet with the jth priority. $\lceil T_w^{(x)}/T_a^{(j)} \rceil$ is the number of packets arriving during the packet's queuing time $T_w^{(x)}$, which are sent out from the queue before this packet.

As mentioned in Section 2.4, after receiving the reply messages from neighbor nodes that includes the scheduling information of all flows in their queues, the source node calculates the T_w of its packets in each intermediate node and then chooses the intermediate node n_i that satisfies $T_w(i) < T_{QoS} - T_{S\to I} - T_{I\to D}$.

After scheduling traffics to qualified intermediate nodes based on Equation (3), the earlier generated packet

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

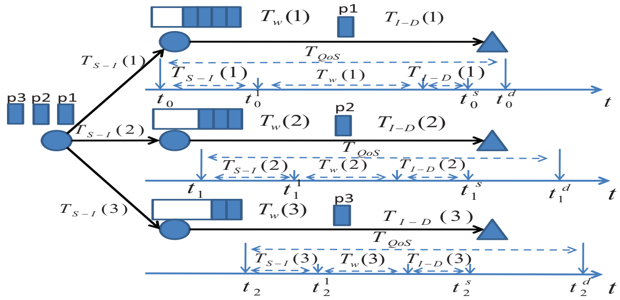


Fig. 3: Distributed queuing mechanism.

from source node is transmitted to a node with longer queuing delay but still within the deadline bound. Taking advantage of the different T_w in different neighbor nodes, the transmission time of the entire traffic stream can be decreased by making the queuing of previous generated packets and the generating of new packets be conducted in parallel.

As Figure 3 shows, a source node generates three packets p_1, p_2 and p_3 with the same size at times t_0, t_1 and t_2 ($t_0 < t_1 < t_2$), respectively. A packet p's total transmission delay equals: $T_{S\to I}(i) + T_w(i) + T_{I\to D}(i)$. Since all these packets are generated from the same node, the transmission delay from the source node to each intermediate node $T_{S\to I}(1)$, $T_{S\to I}(2)$ and $T_{S\to I}(3)$ are almost the same. To simplify the analysis, we suppose $T_{I\to D}(1) = T_{I\to D}(2) = T_{I\to D}(3)$. If the queuing delay in each intermediate node satisfies $T_w(1) > T_w(2) > T_w(3)$, then packet p_1 should be sent to the first intermediate node, packet p_2 should be sent to the second intermediate node, and packet p_3 should be sent to the third intermediate node. As a result, the final packet delivery time for the three packets from the intermediate nodes to the destination node can be reduced.

Theorem 2.4: Given a certain amount of packets to transmit, QOD produces higher throughput than the single shortest path transmission method [20], in which a source node always transmits packets through a single shortest path, and the distributed randomized method, in which packets are randomly assigned to the selected intermediate nodes.

Proof: Suppose the packet generating rate of a source node n_i is λ. That is, the packet arrival interval is $T_a = \frac{1}{\lambda}$. The queuing delay time T_w in different intermediate nodes are different, and $T_w(1) > T_w(2) > T_w(3) > T_w(j) > ... > T_w(m)$ ($1 \leq i \leq m$). According to the distributed packet scheduling algorithm in QOD, the time to transmit m packets to different APs through m qualified intermediate nodes is $T^{[d]} = max(i \cdot T_a + T_w(i)) + T_{S\to I} + T_{I\to D}$ ($1 \leq i \leq m$). According to the Little's law [36], for a stable system, the packet queuing time is less than the packet arrival interval, i.e., $T_w(i) < T_a$. As i increases, $i \cdot T_a$ increases and $T_w(i)$ decreases. Therefore, $T^{[d]} = mT_a + T_w(m) + T_{S\to I} + T_{I\to D}$. Since a packet in a queue needs to wait for a number of packets to be sent out before it can be forwarded, $T_w(m) \gg T_{S\to I}$ and $T_{I\to D}$. Therefore, $T^{[d]} \approx mT_a + T_w(m)$. Similarly, the transmission time of distributed randomized algorithm is $T^{[r]} \approx mT_a + T_w(k)$ ($1 \leq k \leq m$). In the single shortest path transmission method, the time for transmitting m packets is $T^{[s]} = mT_a + \sum_{i=1}^{m} T_w(i) + T_{S\to I} + T_{I\to D} \approx mT_a + \sum_{i=1}^{m} T_w(i)$. Then,

$$T^{[d]} - T^{[s]} = T_w(m) - \sum_{i=1}^{m} T_w(i) \leq 0.$$

Therefore, $T^{[d]} \leq T^{[s]}$.
Similarly, $T^{[d]} - T^{[r]} = mT_a + T_w(m) - (mT_a + T_w(k))$.
Since $T_w(m) \leq T_w(k)$, $T^{[d]} \leq T^{[r]}$. □

As the throughput in two-hop transmission is normally less than the throughput of direct transmission, the two-hop transmission is only used in two cases: (1) when the packet sender is out of the range of an AP, and (2) APs in range are congested. In these two cases, the direct communication to an AP cannot provide QoS guarantee, and the two-hop transmission is needed.

2.6 Mobility-based Packet Resizing Algorithm

In a highly dynamic mobile wireless network, the transmission link between two nodes is frequently broken down. The delay generated in the packet re-transmission degrades the QoS of the transmission of a packet flow. On the other hand, a node in a highly dynamic network has higher probability to meet different mobile nodes and APs, which is beneficial to resource scheduling. As Equation (2) shows, the space utility of an intermediate node that is used for forwarding a packet p is $\frac{S_p}{W_i \cdot T_a}$. That is, reducing packet size can increase the scheduling feasibility of an intermediate node and reduces packet dropping probability. However, we cannot make the size of the packet too small because it generates more packets to be transmitted, producing higher packet overhead. Based on this rationale and taking advantage of the benefits of node mobility, we propose a mobility-based packet resizing algorithm for QOD in this section. The basic idea is that the larger-size packets are assigned to lower-mobility intermediate nodes and smaller-size packets are assigned to higher-mobility intermediate nodes, which increases the QoS-guaranteed packet transmissions. Specifically, in QOD, as the mobility of a node increases, the size of a packet S_p sent from a node to its neighbor nodes i decreases as following

$$S_p(new) = \frac{\gamma}{v_i} S_p(unit), \tag{5}$$

where γ is a scaling parameter and v_i is the relative mobility speed of the source node and intermediate node and S_p(unit)=1kb.

Proposition 2.2: The QOD protocol can provide soft state QoS of packet routing in a highly dynamic network.

Proof: Since the packet size of a packet that is sent from a source node to an intermediate node n_i is $\frac{\gamma}{v_i}$, and the average packet arrival interval is $\Theta(\frac{1}{\lambda})$, the space utility of a node n_i with bandwidth W_i is $U_s(i) = \frac{S_p}{W_i \cdot T_a} = \frac{\frac{\gamma}{v_i}}{W_i \cdot \frac{1}{\lambda}} = \frac{\gamma \cdot \lambda}{W_i \cdot v_i}$. As v_i increase, space utility $U_s(i)$ decreases. Then, the traffic from the source node is more easily to be scheduled. Therefore, based on QOD routing protocol, QoS of the traffic can be guaranteed in a highly dynamic situation. □

2.7 Soft-Deadline Based Forwarding Scheduling

Recall that in the EDF algorithm, an intermediate node forwards the packets in the order from the packets with the closest deadlines to the packets with the farthest deadlines. If an intermediate node has no problem to meet all packets' deadlines in forwarding, that is, the packets are scheduling feasible, the EDF algorithm works satisfactorily. However, when an intermediate node has too many packets to forward out and the deadlines of some packets must be missed, EDF forwards out the packets with the closest deadlines

	Transmission time $T_{I \to D}$	Deadline	Packet arrival interval
Packet flow a	2	Arrival time +2	32
Packet flow b	3	Arrival time +3	32
Packet flow c	2	Arrival time +3	32

Fig. 4: An example of packets received by the forwarder.

Deadline table

Time	0	2	5
Packet flow a	$\underline{\mathbf{2}}^1$		
Packet flow b	$\underline{\mathbf{3}}^1$	3^1	
Packet flow c	$\underline{\mathbf{3}}^1$	3^1	3^1

Slack time table

time	0	1	2	3	4	5	6
Packet flow a	$\underline{\mathbf{0}}^1$	0^1	-1^1	-2^1			
Packet flow b	$\underline{\mathbf{0}}^1$	-1^1	-1^1	-2^1	-3^1	-3^1	-4^1
Packet flow c	$\underline{\mathbf{1}}^1$	0^1	-1^1	-1^1	-2^1	-3^1	

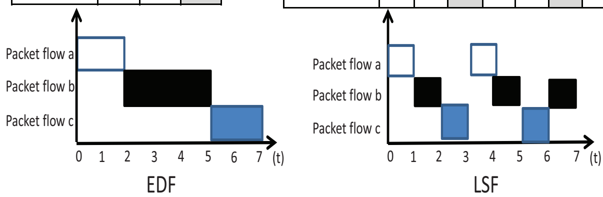


Fig. 5: Comparison of the forwarding scheduling methods.

but may delay the packets with the farthest deadlines. Therefore, EDF is suitable for hard-deadline driven applications (e.g., on-line conferences) where packets must be forwarded before their deadlines but may not be fair to all arriving packets in soft-deadline driven applications (e.g., online TV), where the deadline missing is sometimes acceptable.

In order to achieve fairness in the packet forwarding scheduling for soft-deadline driven applications, a forwarding node can use the least slack first (LSF) scheduling algorithm [37]. The slack time of a packet p is defined as $D_p - t - c'$, where t is the current time and c' is the remaining packet transmission time of the packet. For example, a packet's remaining forwarding time is 5s and the time interval from current time to its deadline is 20s. Then, its slack time equals 20-5=15s. With the LSF algorithm, an intermediate node periodically calculates the slack time of each of its packets, and forwards the packet with the least slack time. If all packets have the same slack time value, one packet is randomly chosen to be sent out.

Therefore, the objective of LSF is different from that of EDF. LSF does not aim to complete transmitting the packet flows before their deadlines. Rather, it aims to make delays and the sizes of delayed part in the delayed packets (delayed size in short) of different packet flows almost the same. If the packets are scheduling feasible according to Equation (2), the LSF algorithm can meet all deadlines of packets. Otherwise, the forwarding node takes turns to forward the packets based on their slack times. Therefore, LSF can achieve more fairness than EDF. QOD can choose either LSF or EDF based on the applications and we can apply Equation (4) for LSF the same as EDF. The priorities of the packets are determined by the chosen policy.

Below, we use an example of an intermediate node's packet forwarding scheduling to compare the effect of EDF and LSF. To help readers understand the process more clearly, in the supplementary file, we present a more complex example where each packet flow has multiple packet arrivals during a certain time period.

Figure 4 shows the parameters in the example. A forwarding node is receiving packets from three packet flows: a, b and c. Because of the resizing algorithm, the incoming packets from different source nodes may have different packet sizes, hence have different packet transmission time periods. For example, the first row indicates that the forwarding node receives packets from packet flow a with arrival interval 32s. Each packet has the deadline equals "arrival time+2s", and needs 2s to be transmitted to its destination.

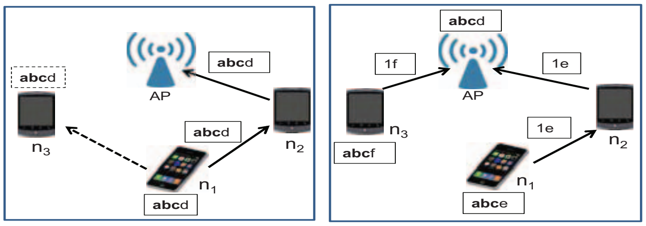
The tables in Figure 5 show the deadline and slack time at each second for each packet, and the figures below show the forwarding scheduling results of EDF and LSF. In the tables, we use the superscript to denote the packet sequence number. For example, in the deadline table, in the line of "packet flow a", 2^1 at time 0 means that the deadline of the first packet of packet flow a is 2 at time 0. Similarly, in the slack time table, in the line of "packet flow a", 0^1 at time 0 means that the slack time of the first packet of packet flow a is 0 at time 0. The slack time 0^1 at time 0 is calculated by subtracting current time 0 and the remaining transmission time 2 from the deadline 2. The gray color means that this packet has the smallest deadline or slack time and is chosen to be forwarded. A bold value with underscore mean it is for the newly arrived packet.

In EDF, packet b and packet c missed deadline for delay 2s and 4s respectively. The variance is 1. Their delayed sizes are 2kb and 2kb respectively. In LSF, all of the packets missed deadline with delays equal to 2s, 4s and 3s respectively. The variance is 0.67. In LSF, suppose the forwarding rate of a node is 1kb/s, the delayed size of flows a, b and c are 1kb, 2kb and 1kb, respectively. Comparing the results of LSF with those of EDF, we see that LSF distributes delayed time and delayed size to more packet flows, achieving fairness in forwarding scheduling while EDF reduces the number of delayed packets.

2.8 Data Redundancy Elimination

As we introduced in Section 2.3.1, the mobile nodes set their NAV values based on the overhearing message's transmission duration time. A large NAV leads to a small available bandwidth and a small scheduling feasibility of the mobile nodes based on Equation (2). Therefore, by reducing the NAV value, we can increase the scheduling feasibility of the intermediate nodes and sequentially increase the QoS of the packet transmission. Due to the broadcasting feature of the wireless networks, in a hybrid network, the APs and mobile nodes can overhead and cache packets, we use an end-to-end Traffic Redundancy Elimination (TRE) algorithm [38] to eliminate the redundancy data to improve the QoS of the packet transmission in QOD. TRE uses a chunking scheme to determine the boundary of the chunks in a data stream. The source node caches the data it has sent out and the receiver also caches its received data.

In QOD with TRE, the AP and mobile nodes overhear and cache packets. From the overhearing, the nodes know who have received the packets. When a source node begins to send out packets, it scans the content for duplicated chunks in its cache. If the sender finds a duplicated chunk and it knows that the AP receiver has received this chunk before, it replaces this chunk with its signature (i.e., SHA-1 hash value). When the AP receives the signature, it searches the signature in its local cache. If the AP caches the chunk associated with the signature, it sends a confirmation message to the sender and replaces the signature with the matched data chunk. Otherwise, the AP requests the chunk of the signature from the sender.



(a) Packet overhearing (b) Redundant elimination

Fig. 6: An example of packet redundant elimination.

Figure 6 shows an example of packet redundant elimination in hybrid networks. As shown in Figure 6 (a), when node n_1 sends a message "abcd" to a nearby AP through n_2, n_3 overhears the message and caches a chunk "abc" in its local memory. The AP receiver also caches the message. Later on, when n_3 sends a message "abcf" and n_1 sends a message "abce" to the AP, since they know the AP has cached "abc", they only need to send "1f" or "1e", where "1" is the signature of the chunk "abc". Then, the AP is able to reconstruct the full chunk using its signature. The reduction in the size of the message increases the scheduling feasibility of the mobile nodes, which further enhances the QoS performance of the system.

3 Performance Evaluation

This section demonstrates the distinguishing properties of QOD compared to E-AODV [20], S-Multihop [39], Two-hop [27] through simulations on NS-2 [40]. E-AODV is a resource reservation based routing protocol for QoS routing in MANETs. This protocol extends AODV by adding information of the maximum delay and minimum available bandwidth of each neighbor in a node's routing table. To apply E-AODV in hybrid networks, we let a source node search for the QoS guaranteed path to an AP. The intermediate nodes along the path reserve the resources for the source node. In S-Multihop, a node always forwards a packet to a next hop node that has small buffer usage than itself until the packet reaches an AP. In Two-hop, the source node source node adaptively chooses direct transmission (i.e., directly transmit packets to the AP) and forward transmission (i.e, transmit packets through a forwarding node) to forward packets to APs.

In the simulation, the setup was the same as Section 6. Six APs with IEEE 802.11 MAC protocol are uniformly distributed in the area. We randomly selected two source nodes to send packets to APs in every ten seconds. A node's traffic is generated with constant bit rate (CBR) sources. The generation rate of the CBR traffic is 100kb/s. Unless otherwise specified, the speeds of the nodes were randomly selected from [1-40]m/s. Since the number of successfully delivered packets within a certain delay is critical to the QoS of video streaming applications, we define a new metric, namely QoS guaranteed throughput (QoS throughput in short), that measures the throughput sent from a source node to a destination node satisfying a QoS delay requirement as 1s. This metric can simultaneously reflect delay, throughput and jitter features of packet transmission. We directly use the threshold parameter in RED queue [35] as our space utility threshold. We run each experiment for 10 times. We first selected the simulation results within confidence interval of 95% of the ten simulations, and then calculated the average result as the final result. The warmup time was set to 100s and the simulation time was set to 200s per round.

3.1 Performance with Different Mobility Speeds

In this experiment, a node's mobility speed was randomly selected from $[1, x]$m/s (x=1, 10, 20, 30, 40). Figure 7 plots the QoS throughputs of all systems versus the node mobility speed. It shows that the QoS throughputs of all systems decrease as node mobility increases. This is because higher mobility causes higher frequent link breakages, which leads to more packet drops. Re-establishing the broken links results in a long transmission delay for subsequent packets. We can also see that the QoS throughputs of QOD and Two-hop slightly decrease, but those of E-AODV and S-Multihop decrease sharply. E-AODV and S-Multihop have much more hops in the routing paths from the source nodes to APs than QOD and Two-hop. A longer routing path produces higher probability of link breakdown during the packet transmission. As Two-hop and QOD only have two hops in the routing paths to APs, the short paths have lower probability to break down. Even if a link breaks down, the source node can quickly choose another forwarder. Therefore, node mobility does not greatly affect these two protocols.

E-AODV has much smaller QoS throughput than others with different node mobility speeds. This is because in E-AODV, the routing resources in each link are reserved for QoS traffic. In a highly dynamic network, the reserved links constantly break down, which leads to the invalid reservation problem, forcing the source node to search for a new path to an AP. The delay resulted from the path searching degrades the ability to meet the QoS requirements. The race condition problem further decrease the QoS throughput as the same resources are reserved for different source nodes at the same time. Then some source nodes cannot obtain the resources as scheduled. Therefore, the QoS of the packet traffic in E-AODV is very difficult to guarantee in a highly dynamic network. Since a node in S-Multihop directly forwards a packet to the next hop with smaller buffer usage without reserving resource, it generates higher QoS throughput than E-AODV that suffers from delay from path discovery. Meanwhile, in S-Multihop, as several source nodes may send packets to the node with smaller buffer usage at the same time, the node is very easily congested. Although the routing path length in Two-hop is always two as QOD, as Two-hop only concerns node bandwidth in packet forwarding rather than buffer usage, it may suffer severe buffer congestion in the selected node with high bandwidth. Therefore, S-Multihop produces higher QoS throughput than Two-hop in a low-mobility network. However, S-Multihop suffers severely from node mobility due to it long paths while Two-hop is mobility-resilient due to its short path. Thus, S-Multihop generates less QoS throughput than Two-hop in a high node mobility.

In QOD, rather than reserving the resources in each transmission link, the intermediate nodes periodically report their queuing status to the source node. The source node adaptively schedules the packets to the neighbor nodes based on their current space utilities. In this way, there is no need for retransmission caused by invalid resource reservation. Moreover, since every intermediate node can receive scheduled packets for the forwarding transmission, and the same scheduled

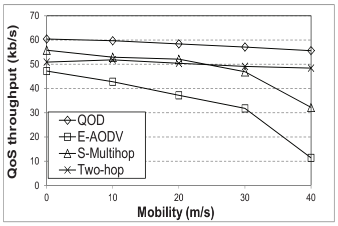


Fig. 7: QoS throughput versus mobility.

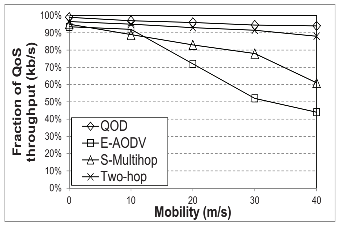


Fig. 8: Fraction of QoS throughput versus mobility.

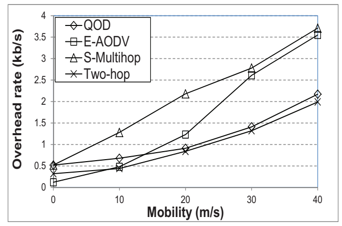


Fig. 9: Overhead versus mobility.

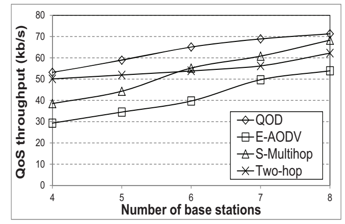


Fig. 10: QoS throughput versus number of APs.

resource will not be allocated to more than one source node at the same time, the race contention problem can be avoid. Furthermore, a packet resizing algorithm is used for traffic scheduling by leveraging the various mobility of the nodes for QoS guaranteed routing. It can increase the scheduling feasibility of an intermediate node on a packet. Because of the increased overhead due to more packet heads in QOD in a higher dynamic network, its QoS throughput decreases slightly as node mobility increases. Although both QOD and Two-hop have at most two hops from source nodes to APs, QOD constantly generates higher QoS throughput than Two-hop. This is caused by two reasons. First, QOD dynamically schedules the packets to the neighbors that can guarantee QoS routing based on Equation (2), while Two-hop forwards the packets to the nodes with high bandwidth which may become congested. Second, Two-hop does not take advantage of low-bandwidth nodes which may still support the QoS routing due to lower queue delay, while QoD makes full use of the resources of the nodes around of a source nodes, and distributively forwards the packets to the APs, improving the QoS throughput of the system.

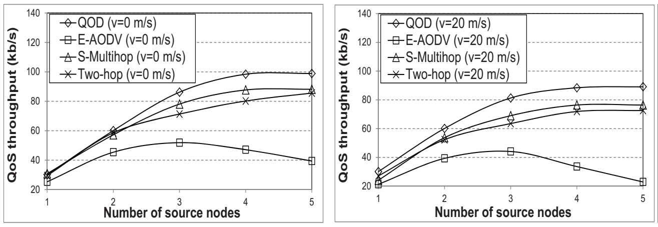
Though both QOD and S-Multihop schedule the packet forwarding to the APs, QOD constantly outperforms S-Multiple with different mobility speeds due to three reasons. First, QOD is more mobility-resilient due to its short path length and its mobility-based segment resizing algorithm. While S-Multiple is not mobility-resilient due to its long path length. Second, QOD provides a QoS-guaranteed neighbor selection algorithm to ensure that the selected next hop node can guarantee QoS routing. It also uses admission control mechanism to prevent source nodes from generating packets if no neighbor nodes can satisfy the QoS requirement. S-Multihop only focuses on buffer usage of the next hop node, which increases the QoS of packet transmission to a certain extent but cannot ensure the QoS of the forwarding. Consequently, the QoS throughput in QOD is higher than that in S-Multihop.

We define the *fraction of QoS throughput (QoS fraction in short)* as the ratio of QoS throughput to total packet throughput. This metric shows the effectiveness of different systems in supporting QoS routing. Figure 8 shows the QoS fraction of all the systems. The figure shows that as the node mobility speed increases, the fraction of QoS throughput of all systems decreases, i.e., more received packets cannot meet their QoS requirements. Specifically, the QoS fraction in S-Multihop and E-AODV drops sharply, while that of QOD and Two-hop drops marginally as the average mobility of the nodes in the system increases. This is due to the reason that S-Multihop and E-AODV generate longer path lengths than QOD and Two-hop, and hence suffer from more severe link breakdown, which prevents packets from arriving at APs in time. Since QOD's distributed packet scheduling algorithm can avoid race contention as previously explained, and the packet resizing algorithm can increase the scheduling feasibility of the intermediate nodes, its QoS fraction decreases only slightly. This decrease is caused by the increased overhead in the system with higher node mobility. The packet resizing algorithm generates smaller packet size in higher node mobility, thus producing more packets for a given data stream and hence more transmission overhead. We see that the QoS fraction in Two-hop also slightly decreases as the node mobility increases. This is because faster mobility leads to higher frequency of link breakdown and hence more dropped packets on the fly. Figure 8 also shows that QOD has the highest fraction of QoS throughput, and Two-hop constantly outperforms S-Multihop and E-AODV that exhibit the worst performance due to the same reasons as in Figure 7.

We define the overhead rate as the size of all control packets generated by the system in one second. The control packets include all control packets and packet headers excluding the data packets. Figure 9 plots the overhead rates of different systems with different node mobility speeds. We see that the overhead rates of all systems increase as node mobility increases, and the result follows S-Multihop>E-AODV>QOD>Two-hop when node mobility is larger than 15m/s. The overhead of QOD mainly consists of two parts. The first part is caused by periodical status information exchanges. A source node needs to exchange its status information with its neighbor nodes periodically during the packet transmission time for packet scheduling. With higher node mobility, a source node meets more nodes, leading to more exchanged information. The second part is caused by the packet heads. Although the packet size of each packet is reduced as node mobility increases, more packets are generated for a given data stream. The extra packet heads increase the overhead of QOD. Consequently, as node mobility increases, the overhead of QOD also increases.

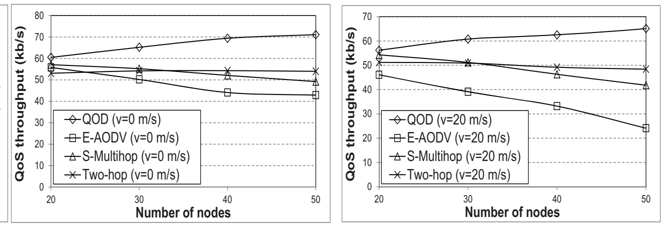
E-AODV does not need to periodically exchange channel information for packet scheduling, and its overhead is mainly caused by routing re-establish. With low node mobility, routing paths break down infrequently. Thus, E-AODV has the least overhead rate in a system with low mobility. The overhead in S-Multihop mainly consists of two parts: routing re-establish overhead and buffer information exchange overhead. Therefore, the overhead of S-Multihop is significantly higher than that in E-AODV when the mobility is low. With high node mobility, the overhead rate of S-Multihop is only slightly higher than that E-AODV. This is because higher node mobility makes the path maintenance overhead rate dominate the entire overhead rate.

The overhead in Two-hop is mainly resulted from



(a) Ave. node mobility v=0m/s (b) Ave. node mobility v=20m/s

Fig. 11: QoS throughput versus number of source nodes.



(a) Ave. node mobility v=0m/s (b) Ave. node mobility v=20m/s

Fig. 12: QoS throughput versus network size and node mobility.

channel information exchange for the dynamic packet forwarding and path re-establish overhead. As the forwarding path length from source nodes to the APs is only two, which is much less than those of E-AODV and S-Multihop, the overhead rate of Two-hop is lower than those of E-AODV and S-Multihop, especially in the system with node mobility. Because QOD produces more exchanged control packets than Two-hop for packet scheduling, it generates slightly larger overhead rate than Two-hop. However, with this slightly additional overhead, QoD greatly increases the QoS throughput of Two-hop as shown in Figure 7.

3.2 Performance with Different Number of APs

Figure 10 shows the QoS throughput versus the number of APs in the different systems. The figure shows that the increase of APs leads to higher QoS throughput in all systems. This is because more APs help to reduce path lengths and physical distances physical distances between source nodes and APs, leading to lower packet transmission the signal power, leading to higher data transmission rate. More APs significantly reduce the significantly reduce the lengths of originally long paths to the APs in E-AODV and S-Multihop, thus dramatically increasing their QoS throughput. In contrast, as QOD and Two-hop two-hop short path length, their QoS throughput increase rate is smaller than those of S-Multihop Due to the same reasons explained in Figure 7, E-AODV produces less QoS throughput than S-Multihop. When the number of the APs in the system is small, the routing path lengths of S-Multihop and E-AODV are longer than those of QOD and Two-hop. Therefore, the QoS throughputs of QOD and Two-hop are larger than those of S-Multihop and E-AODV. It is very interesting to see that S-Multihop has higher QoS throughput than Two-hop when the number of APs in the system is larger than 6. In this case, S-Multihop generates much shorter path lengths. Also, S-Multihop uses scheduling algorithm that considers buffer usage for packet routing, which reduces the packet queuing delay. However, Two-hop only considers channel condition for the packet routing and ignores the buffer usage, making high-bandwidth nodes easily congested. As a result, S-Multihop produces higher QoS throughput than Two-hop. Since E-AODV also suffers from congestion on the nodes close to the APs and its average path length is larger than Two-hop, its QoS throughput is less than Two-hop. As QOD can effectively schedule the channel resources around the source node for packet forwarding, its QoS throughput remains constantly the highest.

3.3 Performance with Different Workloads

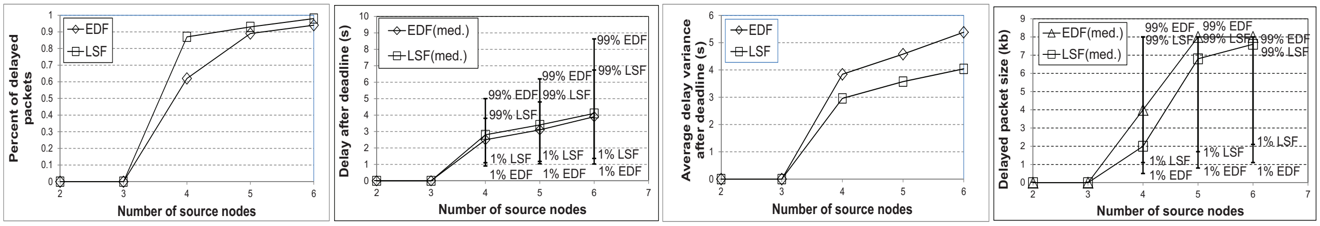
Figure 11 (a) and (b) plot the QoS throughput of the systems with different number of source nodes when the average node mobility is 0m/s and 20m/s, respectively. Each node's mobility speed is randomly chosen from the range from 0m/s to the average mobility. More source nodes generate more workload in the system. We see from both figures that as the number of source nodes increases from 0 to 3, the QoS performance of QOD increases almost linearly. In these cases, the capacity of the system is not saturated, and hence the QoS throughput increases almost linearly as the workload grows. When the number of source nodes increases to 5, the QoS throughput increases at a slower rate. In QOD, when a source node finds that all of its neighbors cannot guarantee the QoS of its packets, it stops generating new packet flows into the system based on the admission control policy. Generating more packets into the networks may further decrease the QoS performance of other source nodes. S-Multihop produces less QoS throughput increase rate than QOD, which means the system with S-Multihop saturates much earlier than that with QOD. Although S-Multihop schedules the packet forwarding by forwarding a packet to the next hop with less buffer usage, which can reduce packet buffering latency, it does not have a mechanism to prevent a node with a full buffer from receiving packets from other nodes in order to ensure the forwarding QoS. Also, as shown in Figure 7, QOD is much more mobility-resilient to S-Multihop, which increases QOD's QoS throughput. In addition, QOD's distributed packet scheduling algorithm can further reduce packet transmission delay, which enhances QOD's capability to handle the increasing workload in the system. As a result, QOD constantly produces higher QoS throughput than S-Multihop.

The figure also shows that Two-hop has less QoS throughput increase rate than S-Multihop as the number of source nodes increases. In Two-hop, the packets are always forwarded to the nodes with higher transmission link rate. Without any buffer management strategy, the nodes with higher transmission links are very easily overloaded as the workload in the system increases. It is very intriguing to see, as the number of source nodes increases, E-AODV's QoS throughput increases initially but decreases later. This is because in E-AODV, when the workload of the system increases, the probability that two or more source nodes simultaneously reserve the same resources at a node increases due to the race condition problem. Also, the nodes close to the APs are more likely to be congested as E-AODV does not have a resource scheduling mechanism. Therefore, the QoS throughput of E-AODV decreases in a highly loaded system. Comparing Figure 11 (a) and (b), we can find that the increasing mobility of the nodes in the system leads to a decrease of QoS throughput of all protocols. The reason is the same as in Figure 7.

3.4 Performance with Different Network Sizes

Figure 12 (a) and (b) illustrate the QoS throughput of the systems with different number of nodes at the average

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.



(a) Number of delays (b) Delay variance (c) Average delay variance (d) Delayed size

Fig. 13: Comparison of EDF and LSF for packet forwarding scheduling.

mobility speed of 0m/s and 20m/s, respectively. Both figures show that as the number of nodes in the system increases, the QoS throughput of QOD increases, that of Two-hop remains constant, but those of E-AODV and S-Multihop decrease. The throughput increase in QOD is caused by the increasing number of nodes in the system, which leads to an increasing number of neighbors of a node, enabling it to have more available resources for packet traffic scheduling. As the Two-hop always lets the source node forward the packets to the next hop node with high link rate without any resource scheduling as used in QOD, the source nodes cannot take advantage of those increased resource nodes around themselves as the number of nodes in the system increases, leading to constant QoS throughput.

In S-Multihop, although more resources are available to transmit packets based on the buffer usage as the network size increases, the average number of routing hops from the source node to destination node also increases, which leads to a higher frequency of link breakdown. The reduced QoS throughput due to the longer routing path dominates the increased QoS throughput due to the more available resources. Thus, the QoS throughput of S-Multihop decreases slightly. In E-AODV, as the number of nodes in the system increases, the average path length grows, which increases the probability of path breakdown and decreases its QoS throughput.

Comparing Figure 12 (a) and Figure 12 (b), we see that as the mobility of the nodes increases from 0m/s to 20m/s, the QoS throughput of E-AODV and S-Multihop decreases more significantly as the network size increases. For QOD and Two-hop, the increase of mobility does not affect their QoS throughput significantly because of their mobility-resilience due to short paths. More details of the reasons are presented as in Figure 7.

The figure also shows that in QOD, a system with a larger number of nodes and source nodes has higher throughput increasing rate. This is due to the reason that a larger number of neighbor nodes can provide more resources for the packet scheduling. Therefore, even though there is more workload in the system, QOD can effectively digest the workload. In Two-hop and S-Multihop, because of the inefficient forwarding node selection, their workload digest ability is not as good as QOD. Therefore, their QoS throughput decreases. The QoS throughput in Two-hop with 2 source nodes decreases slightly when the number of nodes in the system increases because more nodes near the source nodes generate more interference between each other. However, as the number of source nodes increases to 4, its QoS throughput increases. In this case, other idle nodes can be used for packet transmission, and the increased QoS throughput due to idle nodes surpasses the decreased QoS throughput due to the interference, leading to overall QoS throughput increase. In E-AODV, as the number of nodes in the system increases, its QoS throughput decreases. Moreover, as the number of source nodes in the system increases, its QoS throughput exhibits dramatic decrease. This is caused by two reasons: (1) a large number of source nodes produce a high workload in the system, resulting in a high probability of race contention and link breakdown, and (2) a larger number of nodes generate more transmission hops, resulting in a high probability of link breakdown.

3.5 Comparison of EDF and LSF

In this section, we compare EDF with LSF for packet forwarding scheduling in QOD. We let the forwarding nodes receive as many packets from neighbor nodes as possible without admission control to show the performance of EDF and LSF when the packets are scheduling infeasible. In each experiment, during 50s, we continually selected a certain number of random nodes to transmit packets to their randomly selected destinations for a time period randomly chosen from [1-5]s. The link rate from source nodes to relay nodes and from relay nodes to BSs was set to 2M/s. A forwarding node calculates the deadline and slack time every 1ms.

Figure 13(a) shows the percentage of the delayed packets of EDF and LSF. When there are 2 or 3 source nodes, all packets are scheduling feasible, thus there is no delayed packet in both EDF and LSF. As the number of source nodes in the system increases, the percentage of the delayed packets increases. This is because as more packets are generated, every packet in the scheduling queue needs to wait for more time to be forwarded out, which leads to higher delay and hence more delayed packets. We also see that the percentage of the delayed packets in LSF is higher than that of EDF. This is because EDF always tries to meet the deadlines of packets with the earliest deadlines, while LSF tries to balance the delay among the packets. Therefore, EDF is able to meet more deadlines than LSF.

Figure 13(b) shows the 1st percentile, median, and 99th percentile of the delay after deadline of LSF and EDF. We see that EDF has lower 1st percentile delay and higher 99th percentile delay than LSF, i.e., EDF has a greater variance than LSF. This is because EDF forwards the packets with the earliest deadlines but significantly delays the packets with the farthest deadlines. LSF always tries to balance the delay between different packets, so it has much smaller delay variance than EDF. This result is also confirmed by Figure 13(c), which shows that the average delay variance of EDF is higher than that of LSF. Therefore, with the admission control policy that prevents scheduling infeasible packets from being generated in the system, EDF can be used to ensure more packets to meet their hard-deadlines. Without the admission control policy, LSF can be used to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

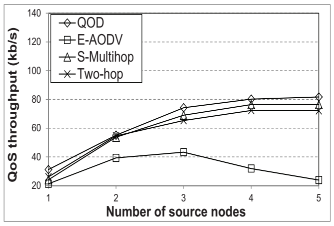


Fig. 14: QoS throughput versus number of source nodes.

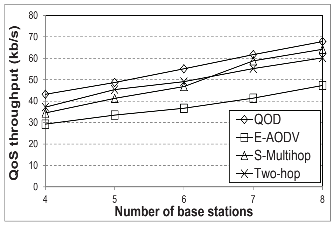
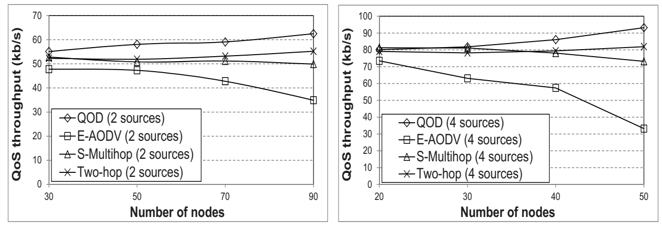


Fig. 15: QoS throughput versus number of APs.



(a) 2 source nodes (b) 4 source nodes

Fig. 16: QoS throughput versus network size and workload.

provide more fairness for the packet forwarding using soft-deadlines. Figure 13(d) shows the 1st percentile, median, 99th percentile of the delayed packet size of LSF and EDF. We can see that the median delayed size of LSF is smaller than that of EDF and LSF also has smaller variance than EDF. This is because EDF aims to meet the hard-deadlines of packets while LSF tries to balance the delayed size among different packet flows, which provides higher fairness for different packet flows.

3.6 Trace-Driven Experiments

In this section, we evaluate the performance of QOD using a more realistic human mobility model based on the trace dataset from the MIT Reality mining project [41] involving 94 students and staffs at MIT. We used the records of the connections with cellular towers in the real trace to infer each node's mobility for the simulation. We also used 6 APs that are uniformly distributed in the system in this simulation. All other setups are the same as in Section 6. Figure 14 shows the QoS throughput versus the number of source nodes. We see that the QoS throughput follows QOD>S-Multihop>Two-hop>E-AODV. Also, unlike others that produce more QoS throughput with more source nodes, E-AODV's QoS throughput decreases when the number of source nodes is more than 3. The reasons for these results are the same as in Figure 11.

Comparing Figure 14 and Figure 11, we can find that each system produces less QoS throughput in the human mobility model than in the random movement model. As nodes in the human mobility model are likely to be clustered and move together due to the community clustering property, the communication interference between the nodes in the same cluster may increase the channel utility of each node, thus reducing the scheduling feasibility of the nodes. Therefore, the QoS throughput of the nodes is reduced in the human mobility model. Since the performance of E-AODV does not depend on the channel utility and space utility of the nodes as it always finds a routing path to an AP through flooding, the QoS throughput of E-AODV does not change greatly.

Figure 15 shows the QoS throughput versus different number of APs. We can see that QOD still generates the highest QoS throughput with different number of APs, which is followed by Two-hop and S-Multihop, and E-AODV produces the smallest amount of QoS throughput. The reasons for the results are the same as in Figure 10. Comparing Figure 15 and Figure 10, we find that the QoS throughputs of QOD, S-Multihop and Two-hop in Figure 15 are lower than those in Figure 10. This is also because the community clustering property reduces the channel utilization of each node due to the interference between nodes within a community. As the routing in E-AODV does not reply on the states of the neighboring nodes, its QoS throughput does not decrease significantly in the human mobility model. We also find that for QOD, its QoS throughput in the human mobility model increases much faster than in the random movement model as the number of APs increases. The nodes in human mobility model suffer more interference than in random movement model as explained above. As APs can reduce the influence of the interference between nodes in the same community since the nodes can directly directly transmit packets to the APs, the increases rate of QoS throughput in human mobility model is much faster than in random movement model. Since a node in S-Multihop and Two-hop depends on fewer neighbors in routing, the routing in S-Multihop and Two-hop suffers less influence from interference than QOD. Thus, as the number of APs increases, their QoS throughput is smaller than that of QOD.

Figure 16 shows the QoS throughput of nodes versus different network sizes and workloads. We see that QOD still generates higher QoS throughput than S-Multihop and Two-hop with different network sizes and workloads, and E-AODV still has the least QoS throughput due to the same reasons as in Figure 12.

3.7 Evaluation of TRE based transmission

In this section, we evaluate the performance of QOD-TRE using a real Internet traffic trace we captured from an access link from a large university in China to the backbone. We still used the MIT trace as the mobility trace. The Internet trace is 120s long and contains 1.9GB HTTP traffic between 8277 host pairs. We used the Internet trace to simulate the web access behaviors of the mobile nodes in the hybrid network. The cache size of each mobile node and AP was set to 250KB and 100MB, respectively. The signature of a chunk is 32 bytes. There are about 5% redundancy data in the trace after being parsed with the TRE algorithm [38].

Figure 17 and Figure 18 show the QoS throughput versus the number of source nodes and the number of nodes, respectively. We see as the number of source nodes or the network size increases, the QoS throughput of both QoD and QOD-TRE increases. We can also see that the system with higher node mobility has less throughput. The reasons are the same as in Figure 14 and Figure 16. We can also see that the QoS throughput of QOD-TRE is consistently larger than the QoS throughput of QOD in both figures. The reason is that instead of sending out a whole packet stream chunks, sending out the chunk signature with smaller packet size can greatly increase the scheduling feasibility of the nodes, leading to an increase in QoS throughput.

4 Related Work

Infrastructure networks. Existing approaches for providing guaranteed services in the infrastructure networks are based on two models: Integrated Services (IntServ) [10] and Differentiated Service (DiffServ) [42].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

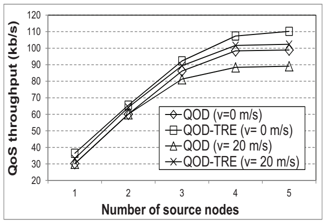


Fig. 17: QoS throughput versus number of source nodes

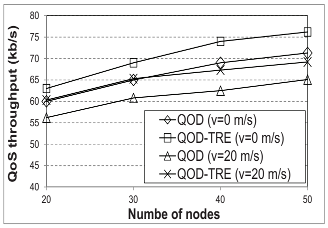


Fig. 18: QoS throughput versus network size.

IntServ is a stateful model that uses resource reservation for individual flow, and uses admission control [10] and a scheduler to maintain the QoS of traffic flows. In contrast, DiffServ is a stateless model which uses coarse-grained class-based mechanism for traffic management. A number of queuing scheduling algorithms have been proposed for DiffServ to further minimize packet droppings and bandwidth consumption [43–47]. Stoica *et. al.* [48] proposed a Dynamic Packet Service (DPS) model to provide unicast IntServ-guaranteed service and Diffserv-like scalability.

MANETs. A majority of QoS routing protocols are based on resource reservation [12], in which a source node sends probe messages to a destination to discover and reserve paths satisfying a given QoS requirement. Perkins *et al.* [20] extended the AODV routing protocol [49] by adding information of the maximum delay and minimum available bandwidth of each neighbor in a node's routing table. Jiang *et al.* [15] proposed to reserve the resources from the nodes with higher link stability to reduce the effects of node mobility. Liao *et al.* [50] proposed an extension of the DSR routing protocol [51] by reserving resources based on time slots. Venataramanan *et al.* [39] proposed a scheduling algorithm to ensure the smallest buffer usage of the nodes in the forwarding path to base stations. However, these works focus on maximizing network capacity based on scheduling but fail to guarantee QoS delay performance.

Some works consider providing multi-path routing to increase the robustness of QoS routing. Conti *et al.* [16] proposed to use nodes' local knowledge to estimate the reliability of routing paths and select reliable routes. The works in [17, 18] balance traffic load among multiple routes to increase routing reliability. Shen *et al.* [19] proposed to let a source node fetch the lost packets from its neighbors to recover the multicast traffic. Shen and Thomas [21] proposed a unified mechanism to maximize both the QoS and security of the routing. Li *et al.* [22] proposed a centralized algorithm to optimize the QoS performance by considering cross-layer design among the physical layer, MAC layer and network layer.

Wireless sensor networks (WSNs). RAP [52] and SPEED [53] give a high delivering priority to the packets with longer distance/delay to the destination. However, both methods require each sensor to know its own location, thus they are not suitable for a highly dynamic environment. Felemban *et al.* [54] and Deb *et al.* [55] proposed to improve routing reliability by multi-path routing. However, the redundant transmission of the packets may lead to high power consumption.

Hybrid wireless networks. Very few methods have been proposed to provide QoS-guaranteed routing for hybrid networks. Most of the routing protocols [23–27] only try to improve the network capacity and reliability to indirectly provide QoS service but bypass the constrains in QoS routing that require the protocols to provide guaranteed service. Jiang *et al.* [56] proposed a resource provision method in hybrid networks modeled by IEEE802.16e and mobile WiMax to provide service with high reliability. Ibrahim *et al.* [23] and Bletasa *et al.* [24] also tried to select "best" relay that has the maximum instantaneous value of a metric which can achieve higher bandwidth efficiency for data transmission. Ng *et al.* [25] considered cooperative networks that use physical layer relaying strategies, which take advantage of the broadcast nature of wireless channels and allow the destination to cooperatively "combine" signals sent by both the source and the relay to restore the original signal. Cai *et al.* [26] proposed a semi-distributed relaying algorithm to jointly optimize relay selection and power allocation of the system. Wei *et al.* [57] proposed to use the first-order finite state Markov channels to approximate the time variations of the average received signal-to-noise ratio (SNR) for the packet transmission and use the adaptive modulation and coding scheme to achieve high spectral efficiency. Lee *et al.* [58] presented a framework of link capacity analysis for optimal transmission over uplink transmission in multi-hop cellular networks. Wei *et al.* [27] proposed a two-hop packet forwarding mechanism, in which the source node adaptively chooses direct transmission and forward transmission to base stations. Unlike the above works, QOD aims to provide QoS-guaranteed routing. QOD fully takes advantage of the widely deployed APs, and novelly treats the packet routing problem as a resource scheduling problem between nodes and APs.

5 Conclusions

Hybrid wireless networks that integrate MANETs and infrastructure wireless networks have proven to be a better network structure for the next generation networks. However, little effort has been devoted to supporting QoS routing in hybrid networks. Direct adoption of the QoS routing techniques in MANETs into hybrid networks inherits their drawbacks. In this paper, we propose a QoS-Oriented Distributed routing protocol (QOD) for hybrid networks to provide QoS services in a highly dynamic scenario. Taking advantage of the unique features of hybrid networks, i.e., anycast transmission and short transmission hops, QOD transforms the packet routing problem to a packet scheduling problem. In QOD, a source node directly transmits packets to an AP if the direct transmission can guarantee the QoS of the traffic. Otherwise, the source node schedules the packets to a number of qualified neighbor nodes. Specifically, QOD incorporates five algorithms. The QoS-guaranteed neighbor selection algorithm chooses qualified neighbors for packet forwarding. The distributed packet scheduling algorithm schedules the packet transmission to further reduce the packet transmission time. The mobility-based packet resizing algorithm resizes packets and assigns smaller packets to nodes with faster mobility to guarantee the routing QoS in a highly mobile environment. The traffic redundant elimination based transmission algorithm can further increase the transmission throughput. The soft-deadline based forwarding scheduling achieves fairness in packet forwarding scheduling when some packets are not scheduling feasible. Experimental results show that QOD can achieve high mobility-resilience, scalability and contention reduction. In the future, we place to evaluate the performance of QOD based on the real testbed.

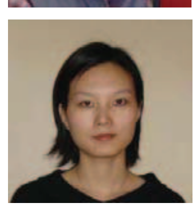
ACKNOWLEDGEMENTS

This research was supported in part by U.S. NSF grants OCI-1064230, CNS-1249603, CNS-1049947, CNS-1156875, CNS-0917056 and CNS-1057530, CNS-1025652, CNS-0938189, CSR-2008826, CSR-2008827, Microsoft Research Faculty Fellowship 8300751, and U.S. Department of Energy's Oak Ridge National Laboratory including the Extreme Scale Systems Center located at ORNL and DoD 4000111689. An early version of this work was presented in the Proc. of MASS'10 [59].

REFERENCES

[1] The state of the smartphone market. http://www.allaboutsymbian.com.
[2] Qik. http://qik.com/.
[3] Flixwagon. http://www.flixwagon.com/.
[4] Facebook. http://www.facebook.com.
[5] H. Wu and X. Jia. QoS multicast routing by using multiple paths/trees in wireless ad hoc networks. *Ad hoc Networks*, 2009.
[6] H. Luo, R. Ramjeey, P. Sinhaz, L. Liy, and S. Lu. UCAN: A unified cell and ad-hoc network architecture. In *Proc. of MOBICOM*, 2003.
[7] P. K. Mckinley, H. Xu, A. Esfahanian, and L. M. Ni. Unicast-based multicast communication in wormhole-routed direct networks. *IEEE TPDS*, 1992.
[8] H. Wu, C. Qiao, S. De, and O. Tonguz. Integrated cell and ad hoc relaying systems: iCAR. *J-SAC*, 2001.
[9] J. Zhou and Y. R. Yang. PAR CelS: Pervasive ad-hoc relaying for cell systems. In *Proc. of Med-Hoc-Net*, 2002.
[10] R. Braden, D. Clark, and S. Shenker. RFC1633: Integrated services in the internet architecture: an overview. *IETF*, 1994.
[11] E. Crawley, R. Nair, B. rajagopalan, and H. Sandick. RFC2205: Resource reservation protocol RSVP. *IETF*, 1998.
[12] I. Jawhar and J. Wu. Quality of service routing in mobile ad hoc networks. *Springer, Network Theory and Appli.*, 2004.
[13] T. Reddy, I. Karthigeyan, B. Manoj, and C. Murthy. Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. *Ad hoc Networks*, 2006.
[14] X. Du. QoS routing based on multi-class nodes for mobile ad hoc networks. *Ad hoc Networks*, 2004.
[15] S. Jiang, Y. Liu, Y. Jiang, and Q. Yin. Provisioning of adaptability to variable topologies for routing schemes in MANETs. *IEEE J-SAC*, 2004.
[16] M. Conti, E. Gregori, and G. Maselli. Reliable and efficient forwarding in ad hoc networks. *Ad hoc Networks*, 2004.
[17] G. Chakrabarti and S. Kulkarni. Load banlancing and resource reservation in mobile ad hoc networks. *Ad hoc Networks*, 2006.
[18] A. Argyriou and V. Madisetti. Using a new protocol to enhance path reliability and realize load balancing in mobile ad hoc networks. *Ad hoc Networks*, 2006.
[19] C. Shen and S. Rajagopalan. Protocol-independent multicast packet delivery improvement service for mobile ad hoc networks. *Ad Hoc Networks*, 2007.
[20] C. E. Perkin, E. M. Royer, and S. R. Quality of service in ad hoc on-demand distance vector routing. *IETF*, 2001.
[21] Z. Shen and J. P. Thomas. Security and QoS self-optimization in mobile ad hoc networks. *IEEE TMC*, 2008.
[22] Y. Li and A. Ephremides. A joint scheduling, power control, and routing algorithm for ad hoc networks. *Ad Hoc Networks*, 2008.
[23] S. Ibrahim, K. Sadek, W. Su, and R. Liu. Cooperative communications with relay-selection: when to cooperate and whom to cooperate with? *IEEE TWC*, 2008.
[24] A. Bletsas, A. Khisti, D. P. Reed, and A. Lippman. A simple cooperative diversity method based on network path selection. *IEEE JSAC*, 2006.
[25] T. Ng and W. Yu. Joint optimization of relay stratefies and resource allocations in cellular networks. *IEEE JSAC*, 2004.
[26] J. Cai, X. Shen, J. W. Mark, and A. S. Alfa. Semi-distributed user relaying algorithm for amplify-and forward wireless relay networks. *IEEE Trans. on wireless Commun.*, 2007.
[27] Y. Wei and D. Gitlin. Two-hop-relay architecture for next-generation WWAN/WLAN integration. *IEEE Wireless Communication*, 2004.
[28] IEEE-SA Standards Board. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
[29] L. Feeney, B. Cetin, D. Hollos, M. Kubisch, S. Mengesha, and H. Karl. Multi-rate relaying for performance improvement in ieee 802.11 wlans. *Wired/Wireless Internet Communications*, 2007.
[30] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE TIT*, 2000.
[31] H. Zhang. Service disciplines for guaranteed performance service in packet-switching networks. In *Proc. of IEEE*, 1995.
[32] M. Karol, M. Hluchyj, and S. Mogan. Input versus output queueing on a space-division packet switch. *IEEE TOC*, 1987.
[33] S. Sahu, P. Nain, C. Diot, V. Firoiu, and D. Towsley. On achievable service differentiation with token bucket marking for tcp. In *Proc. of SIGMETRICS*, 2000.
[34] C. Liu and J. Layland. Scheduling algorithms for multiprogramming in a hard real-time enviroment. *JACM*, 1973.
[35] D. Lin and R. Morris. Dynamics of random early detection. In *Proc. of SIGCOMM*, 1997.
[36] D. P. Bertsekas and Gallager. *Data Networks*. Prentice Hall, 1992.
[37] A. Cheng. Real-time systems: Scheduling, analysis, and verification. *Wiley-Interscience; 1 edition*, 2002.
[38] E. Zohar, I. Cidon, and O. Mokryn. The power of prediction: cloud bandwidth and cost reduction. In *Proc. of SIGCOMM*, 2011.
[39] V. Venkataramanan, X. Lin, L. Ying, and S. Shakkottai. On scheduling for minimizing end-to-end buffer usage over multihop wireless networks. In *Proc. of INFOCOM*, 2010.
[40] Ns-2. http://www.isi.edu/nsnam/ns/.
[41] The reality mining dataset. http://reality.media.mit.edu/.
[42] Y. E. Sung, C. Lund, M. Lyn, S. Rao, and S. Sen. Modeling and understanding end-to-end class of service policies in operational networks. In *Proc. of SIGCOMM*, 2009.
[43] J. Kurose and K. Ross. Computer networking: A top-down approach featuring the internet. *Addison Wesley*, 2004.
[44] L. Kleinrock. Queueing system. *Computer Applications*, 1976.
[45] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control. In *Proc. of Infocom*, 1992.
[46] J. C. R. Bennett and H. Zhang. WF^2Q: Worst-case fair weighted fair queueing. In *Proc. of Infocom*, 1996.
[47] S. Golestani. A self-clocked fair queueing scheme for broadband applications. In *Proc. of Infocom*, 1994.
[48] I. Stoica and H. Zhang. Providing guaranteed services without per flow management. In *Proc. of SIGCOMM*, 1999.
[49] C. Perkins, E. Belding-Royer, and S. Das. RFC 3561: Ad hoc on demand distance vector (AODV) routing. *IETF*, 2003.
[50] W. H. Liao, Y. C. Tseng, and K. P. Shih. A tdma-based bandwidth reservation protocol for QoS routing in a wireless mobile ad hoc network. In *Proc. of ICC*, 2002.
[51] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *IEEE Mobile Computing*, 1996.
[52] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, and T. He. RAP: A real-time communication architecture for large-scale wireless sensor networks. In *Proc. of RTAS*, 2002.
[53] T. He, J. Stankovic, C. Lu, and T. Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. In *Proc. Of ICDCS*, 2003.
[54] E. Felemban, C. Lee, and E. Ekici. MMSPEED: multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. *IEEE TMC*, 2006.
[55] B. Deb, S. Bhatnagar, and B. Nath. ReInForm: Reliable information forwarding using multiple paths in sensor networks. In *Proc. Of LCN*, 2003.
[56] P. Jiang, J. Bigham, and J. Wu. Scalable QoS provisioning and service node selection in relay based cellular networks. In *Proc. of WiCOM*, 2008.
[57] Y. Wei, M. Song, F. R. Yu, Y. Zhang, and J. Song. Distributed optimal relay selection for QoS provisioning in wireless multihop cooperative networks. In *Proc. of GLOBECOM*, 2009.
[58] S. Lee and S. Lee. Optimal transmission methodology for qos provision of multi-hop cellular network. *Journal of Wireless Networks*, 2010.
[59] Z. Li and H. Shen. A QoS-oriented distributed routing protocol for hybrid networks. In *Proc. of MASS*, 2010.
[60] J. Li, C. Blake, S. Douglas, H. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *Proc. of MobiCom*, 2001.

Ze Li Ze Li received the BS degree in Electronics and Information Engineering from Huazhong University of Science and Technology, China, in 2007. He is currently a Ph.D. student in the Department of Electrical and Computer Engineering of Clemson University. His research interests include distributed networks, with an emphasis on peer-to-peer and content delivery networks. He is a student member of IEEE.



Haiying Shen Haiying Shen received the BS degree in Computer Science and Engineering from Tongji University, China in 2000, and the MS and Ph.D. degrees in Computer Engineering from Wayne State University in 2004 and 2006, respectively. She is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Clemson University. Her research interests include distributed computer systems and computer networks, with an emphasis on P2P and content delivery networks, mobile computing, wireless sensor networks, and cloud computing. She is a Microsoft Faculty Fellow of 2010.