

# Reduced Packet Probing Multirate Adaptation For Multihop Ad Hoc Wireless Networks

Jun Cheol Park and Sneha Kumar Kasera  
School of Computing, University of Utah  
{jcpark, kasera}@cs.utah.edu

## Abstract

*In this paper, we conduct a systematic experimental study to identify the challenges for multirate adaptation in the context of multihop ad hoc networks. We first investigate the existing multirate adaptation algorithms to study how they operate in multihop ad hoc networks and obtain insights on the sources of their behavior. Second, we design and implement a novel multirate adaptation scheme, Reduced Packet Probing (RPP), that allows a sender node to effectively approximate channel-error loss in the presence of collisions. We find that our RPP scheme, implemented in the Emulab wireless mesh network [11], achieves 41%–53% higher TCP throughput in comparison to the default multirate adaptation scheme, SampleRate [7], in the MadWifi [18] IEEE 802.11 driver.*

## 1 Introduction

Multirate support in the IEEE 802.11 PHY (physical layer) [3] allows a wireless network interface card to select different data rates<sup>1</sup> at the physical layer so that it can dynamically adapt to diverse channel conditions. Such multirate adaptation capability is critical in improving the performance of IEEE 802.11-based multihop wireless networks although it is left unspecified in the IEEE 802.11 standard [1, 2]. The key idea of multirate adaptation schemes is to find a data rate that can maximize the product of the selected data rate and its associated packet delivery ratio. However, developing an efficient multirate adaptation scheme in multihop ad hoc wireless networks is challenging. This is because accurate assessment of the instantaneous channel conditions in multihop ad hoc wireless networks with multiple collision domains is more difficult than on a single IEEE 802.11 collision domain with multiple contending nodes in an infrastructure mode.

<sup>1</sup>We use the term of *data rate* for “IEEE 802.11 PHY transmission rate.”

None of the existing multirate adaptation algorithms (ARF [4], AARF [19], SampleRate [7], HRC [14], RBAR [12], OAR [6], CARA [16], RRAA [23], CHARM [13]) has been designed or developed for the multihop ad hoc wireless networks. Although a few of these [16, 23, 13] have been evaluated in simple hidden node scenarios, they have not been fully examined for multihop ad hoc paths. Therefore, it is not clear how these existing multirate adaptation algorithms perform in real multihop ad hoc networks. Moreover, while there are some existing multihop ad hoc network testbeds (e.g., Roofnet [15], Net-X [20], QuriNet [22]), several factors including (i) limited or no remote accessibility, (ii) lack of code maintenance especially for compatibility with new device drivers, and (iii) difficulty in programming at different layers, have impeded development of new multirate adaptation algorithms in multihop ad hoc networks. For example, the Roofnet [15] multihop ad hoc network, developed at MIT, using the Click toolkit [8] is not fully compatible with the recent versions of the MadWifi [18] IEEE 802.11 Linux driver.

We conduct a systematic experimental study to identify the challenges for multirate adaptation in the context of multihop ad hoc networks. We first investigate the existing multirate adaptation algorithms to study how they operate in multihop ad hoc networks and obtain insights on the sources of their behavior. As expected, our experiments show that the existing multirate adaptation algorithms are perturbed by collisions in rate selection. We also find that the interaction of channel-error and collision loss in multihop ad hoc paths causes an interesting problem that we call *packet loss anomaly*. We explain this anomaly as follows: When a MAC (Medium Access Control) packet is lost due to channel-errors, a typical multirate adaptation algorithm at the sender of the frame, would reduce its data rate to reduce the channel-errors. However, if the loss is due to a collision, the reduction in the data rate would increase the transmission time of the MAC packets, resulting in an increased probability of collisions. Designing a multirate adaptation scheme that carefully considers the reasons for packet loss in a multihop ad hoc network with multiple

collision domains is difficult.

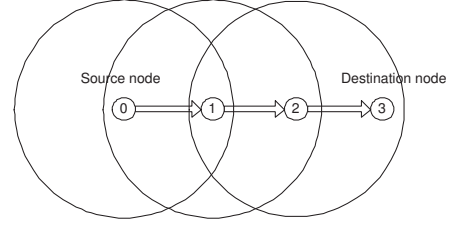
In this paper, we design and implement a novel multirate adaptation scheme, Reduced Packet Probing (RPP), that allows a sender node to effectively approximate channel-error loss in the presence of collisions. Our RPP scheme is based on the observation that the collision loss mainly depends on the actual medium occupancy time. In our RPP scheme, a sender node periodically invokes a *probing phase*. In the probing phase, the sender node examines the delivery ratio for different data rates starting from a higher data rate down to lower data rates, to find a suitable data rate. While downgrading the data rate, it reduces the size of the packet to be transmitted so that the actual transmission time of the reduced packet at the lowered data rate is same as the one of the packet at the previously higher data rate. We implement our RPP multirate adaptation scheme in a real wireless mesh network which is part of the Emulab [11]. Note that this network is not emulated, but a true multihop mesh network with nodes placed inside the Merrill Engineering Building at the University of Utah. However, like other nodes in the Emulab, the nodes of this mesh network can be programmed remotely. Our implementation results show that our RPP scheme achieves up to 41%–53% higher TCP throughput in comparison to the SampleRate [7] multirate adaptation algorithm on IEEE 802.11a-based ad hoc paths.

## 2. Insufficient Collision Awareness

In this section, we first investigate the existing multirate adaptation algorithms to specifically evaluate their collision awareness in multihop ad hoc networks. Using a simple experiment, we also observe that the use of RTS/CTS (Request To Send/Clear To Send) does not comprehensively deal with collisions, thereby the dependence on RTS/CTS for resolving collisions does not result in suitable rate selection. Finally, we discuss the fundamental limitations of IEEE 802.11 networks to measure collision loss in multihop ad hoc networks.

### 2.1 Limitations of Rate Selection

Consider an IEEE 802.11a-based 3-hop ad hoc path, as shown in Fig. 1. The circle around each node represents its transmission range and its clear channel assessment (CCA) range (also commonly called “physical carrier sensing” range). The CCA range determines how far a node can sense whether or not the channel is busy. In general, the CCA range is greater than or equal to the transmission range. For IEEE 802.11a, the CCA range is same as the transmission range, as observed in [9]. More specifically, the CCA range in the IEEE 802.11a PHY is at most the transmission range of data at 6 Mbps, regardless of the actual data rates. In contrast, the CCA range in the IEEE 802.11b PHY [2]

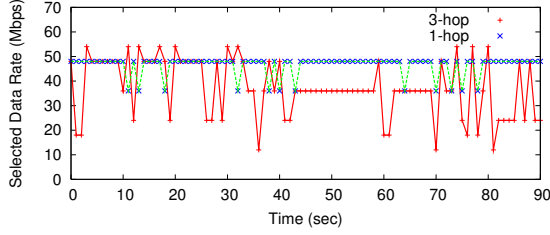


**Figure 1. A 3-hop ad hoc path in IEEE 802.11a.**

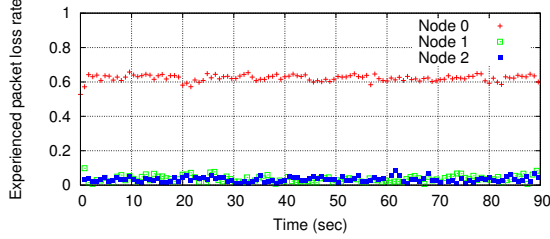
is approximately the double size of the transmission range. Therefore, in our case in Fig. 1, Node 0 is out of the CCA range of Node 2 and vice versa. Note that, when a wireless link is an IEEE 802.11b PHY link, with a 4-hop ad hoc path, a similar situation will arise with Node 0 and the fourth node in the ad hoc path being out of the CCA range of each other.

We first experiment with the 3-hop ad hoc path, shown in Fig. 1, on the Emulab wireless mesh network [11] to demonstrate the limitation of the existing SampleRate [7] algorithm in distinguishing between collision loss and channel-error loss. On the Emulab wireless mesh network, we use the Roofnet [15] software that implements Dynamic Source Routing (DSR) [10] using the Click toolkit [8]. Each wireless node has Netgear WAG311 802.11a/b/g cards that are based on the Atheros chipset [5] (AR5212). We have modified the MadWifi-0.9.3 [18] IEEE 802.11 driver on Linux-2.6.16.13 to make it compatible with Click. The MadWifi driver provides four different multirate adaptation algorithms. Among these, we focus on the default SampleRate [7] multirate adaptation algorithm (other algorithms in the MadWifi driver also show similar or worse behavior).

The SampleRate algorithm uses Expected Transmission Time (ETT) [7, 15] measurements to determine the most suitable data rate. ETT is expected to represent only channel-error loss. However, that is not true in reality. We conduct two separate sets of experiments. The first set of experiments uses an Iperf-2.0.2 TCP flow on a 1-hop path from Node 0 to 1. The second set of experiments use an Iperf-2.0.2 TCP flow on a 3-hop path from Node 0 through 3. In both cases, the TCP flows last for 90 seconds. Fig. 2 shows the selected data rates determined by the SampleRate algorithm as a function of time in both sets of experiments. As shown in Fig. 2, the selected data rates for the first link (from Node 0 to 1) in a 3-hop path fluctuate a lot, as opposed to the 1-hop case where there is no collision. Even though the first link channel-error is same for both the 1-hop path and the 3-hop path, we find that the measured ETT values for data rates at Node 0 are perturbed due to the additional loss caused by collisions in the 3-hop case. Thus, the SampleRate algorithm, oblivious to the cause of loss, misinterprets the channel conditions based on its ETT



**Figure 2. Selected data rates (Mbps) at the first link for a 1-hop path and a 3-hop path.**

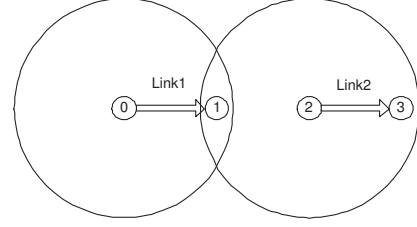


**Figure 3. Packet loss experienced at each node on a 3-hop path when a TCP flow is used.**

measurements, thereby resulting in a poor choice of data rates.

Robust Rate Adaptation Algorithm (RRAA) [23] has been shown to perform better than the SampleRate algorithm. We also use the same topology in *ns-2* [21] simulations to demonstrate similar limitations of the RRAA algorithm. Due to lack of per-packet RTS/CTS control in the Atheros chipset [5], RRAA that requires a dynamic on-off of RTS/CTS exchange per packet cannot be easily implemented. So we simulate RRAA in *ns-2* to observe how packet collisions can affect RRAA's performance in multihop ad hoc networks. We simulate TCP flows on the 3-hop ad hoc path shown in Fig. 1. Each of the three links in the ad hoc path are assumed to be free from channel-errors. Fig. 3 shows the packet loss experienced by each node. Interestingly, Node 0 experiences packet loss as high as  $\approx 60\%$  on the first link. This implies that all lost packets at Node 1 are caused by collisions because each link is free from channel-error. This happens because Node 0 has a hidden node, Node 2, in the 3-hop ad hoc path.

We observe that RRAA-BASIC (no use of RTS/CTS) always chooses 6 Mbps for the first link due to high packet loss caused by collisions in the 3-hop path. Recall that RRAA uses a short term packet loss measurement to determine the best data rate. Unfortunately, because Node 0 experiences high packet loss, RRAA also misinterprets those packet losses, considering them to be due to very high channel-error, and unnecessarily downgrades the data rate. Even for the lowered data rate, the packet loss would not decrease. Consequently, RRAA reduces the data rate even further. Very soon, the data rate at Node 0 becomes the lowest



**Figure 4. Two 1-hop links in IEEE 802.11a.**

data rate (6 Mbps). Therefore, the performance of RRAA is worse than SampleRate in our experiments. Although RRAA provides an adaptive use of RTS/CTS to avoid such collisions (i.e., RRAA-ARTS), in the next section, we observe that even the exchange of RTS/CTS cannot deal with collisions in a satisfactory manner in multihop ad hoc paths.

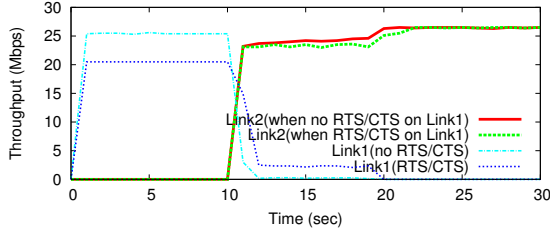
## 2.2 Limitations of RTS/CTS in Multihop Paths

As pointed out in MACAW [24], the use of RTS/CTS exchange does not resolve the hidden node problem in a satisfactory manner, in multihop wireless networks. Consider a simple scenario with two 1-hop ad hoc paths, as shown in Fig. 4. A fundamental problem is that, when Node 0 transmits an RTS or a data packet to Node 1 while Node 2 is also transmitting a data packet to Node 3, the packet sent by Node 2 will collide at Node 1 with the packet sent by Node 0. However, even with collisions, Node 2 is always able to successfully transmit the data packet to Node 3 (in the case of no channel-errors).

Here, we verify this hidden terminal problem using the topology in Fig. 4 on the Emulab wireless mesh network. To avoid any possible impact of multirate adaptation algorithm, we manually find the best data rate for Link1 and Link2 and use 36 Mbps as a fixed data rate for both links with almost zero channel-error. Initially at time  $t_0$ , Link1 starts a UDP flow that lasts for 20 seconds. At time  $t_0 + 10$  seconds, Link2 starts its UDP flow that also lasts for 20 seconds. As a result, there is a 10-second overlapping period between the two UDP flows. We experiment with two scenarios where Link1 does and does not use RTS/CTS. In both the scenarios, Link2 does not use RTS/CTS.

Fig. 5 shows the flow throughput as a function of time for different scenarios. We see that, when RTS/CTS is disabled on Link1, once Link2 starts its UDP flow, Link1 can never occupy the medium. Even when RTS/CTS is enabled on Link1, Link1 obtains only 11% of its achievable throughput when Link2 is active. Moreover, the throughput of Link2 does not change much regardless of the use of RTS/CTS on Link1. This phenomenon is essentially identical when RTS/CTS is enabled on Link2 (not shown here).

Thus, the exchange of RTS/CTS does not resolve the hidden node problem satisfactorily in multihop wireless net-



**Figure 5. Throughput of Link1 and Link2 on the Emulab wireless mesh network.**

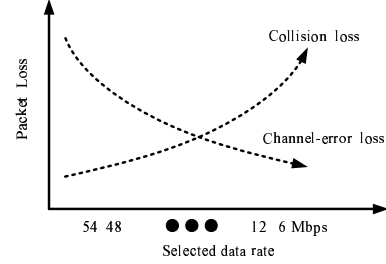
works when a hidden node is transmitting data. Now, going back to our 3-hop ad hoc path in Fig. 1, Node 2 becomes a hidden node whenever it has data to forward to Node 3. This hidden node problem also cannot be resolved with RTS/CTS. Using our *ns-2* simulations, we find that, when RRAA-ARTS is used, Node 0 still experiences packet loss as high as  $\approx 30\%$ . Such a high packet loss is still enough to take the data rate all the way down to 6 Mbps, similar to the case of RRAA-BASIC. Given that RRAA is difficult to implement on the Atheros chipset and noting that its performance is worse than SampleRate, we ignore RRAA in the rest of this paper.

Despite the two causes of packet loss (channel-error or collision), the IEEE 802.11 distributed coordination function (DCF) [3] considers all observed loss to be caused by collisions only. This is because there is no mechanism in DCF to distinguish loss due to channel-error from those due to collisions. As a result, DCF does not provide any explicit collision information.

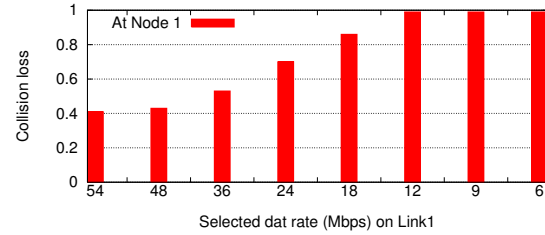
### 3 Packet Loss Anomaly

The relationship between channel-error loss and collision loss as a function of a selected data rate is shown in Fig. 6. Interestingly, the reduction in the data rate increases the frame transmission time and thus increases the medium occupancy time. This increase in the medium occupancy time actually increases the probability of collision loss. As a result, when a data rate decreases, the channel-error loss decreases whereas the collision loss increases. This *packet loss anomaly* between the channel-errors and the collisions makes the selection of data rates in multihop ad hoc wireless networks very challenging.

To investigate the packet loss anomaly problem in detail, consider two IEEE 802.11a-based 1-hop ad hoc paths, as shown in Fig. 4. The two paths are assumed to be free from channel-error in order to solely observe the impact of collision loss. The circle around each node corresponds to its transmission range and its clear channel assessment (CCA) range (also commonly called “physical carrier sensing” range). In our case in Fig. 4, Node 2 is a hidden node



**Figure 6. The problem of *packet loss anomaly* between channel-error loss and collision loss.**



**Figure 7. As the selected data rate in Link1 decreases, the collision loss increases at Node 1.**

for Node 0. Therefore, while Link2 is being used to transmit data packets, if Node 0 transmits its data packet to Node 1, the data packet will collide at Node 1. We simulate these ad hoc paths in *ns-2* and observe the collision loss as a function of data rate. In our simulation, Link2 has a moderate CBR (constant bit rate) load of 10 Mbps at 54 Mbps data rate.

Fig. 7 shows the collision loss at Node 1 as a function of the selected data rate on Link1. We find that the collision loss increases at Node 1 as the data rate on Link1 decreases. This result confirms that the increase in the medium occupancy time actually increases the probability of collision loss.

## 4. DESIGN

In order to address the packet loss anomaly problem in multirate adaptation schemes for multihop ad hoc wireless networks, we propose a new multirate adaptation scheme, Reduced Packet Probing (RPP), that effectively approximates channel-error loss in the presence of collisions for multihop ad hoc wireless networks.

### 4.1 Overview

In this section, we present an overview of our basic RPP scheme that consists of three components: a probing phase to probe different data rates, a state machine to handle the



dynamics of collisions, and a rate decision algorithm to determine a suitable data rate. In the subsequent sections, we described detailed algorithms for each component.

In our RPP scheme, a sender node invokes a *probing phase* when it starts to experience collisions.<sup>2</sup> In the probing phase, the sender node examines the delivery ratio for different data rates, starting from a higher data rate down to lower data rates to find a suitable data rate that maximizes the throughput. However, it is possible that many packets can be dropped when very high data rates for poor channel conditions are used in probing. Therefore, in our scheme, a sender node does not always start probing from the highest data rate. Instead, it starts to probe from a data rate that has a reasonable delivery ratio so as to avoid too aggressive and useless data rate probing.

While downgrading its data rate, the sender reduces the packet size to be transmitted, using link-level fragmentation that is already provided by the IEEE 802.11 standards, so that the actual transmission time of the reduced packet at the lowered data rate is same as the one of the packet at the previously higher data rate. The rationale behind this approach is the observation that the collision loss mainly depends on the actual medium occupancy time regardless of the selected data rate. By keeping the medium occupancy time same for different data rates (i.e., keeping the collision loss same), any changes in the packet loss at the sender node can be inferred as changes in the channel-error loss. The main idea in determining a suitable data rate from the collected information on packet loss for different data rates is to find a data rate at which the sender node does not observe any further increase in the measured packet delivery ratio while downgrading data rates. Because our RPP scheme guarantees that there is no change in collision loss, if there is no improvement in the measured packet delivery ratio for a specific lower data rate, we stop probing at lower data rates because no further reduction in channel-errors is likely. Thus, our approach enables a sender node to find a suitable data rate without any help from its receiver node by performing an accurate analysis of channel-error loss in the presence of collisions.

## 4.2 Packet Size

In order to keep the medium occupancy time same for different data rates in our probing phase, a fundamental question is what packet sizes should be used for lower data rates? In order to answer this question in a general way, we use Equation (29) from [1], that gives the packet transmission time on a perfect channel. Let  $T(R, S)$  denote the transmission time for a data rate  $R$  and a packet size  $S$  over an OFDM channel in the IEEE 802.11a PHY, as described

<sup>2</sup>We will shortly present a detailed method on what conditions are required to invoke the probing phase.

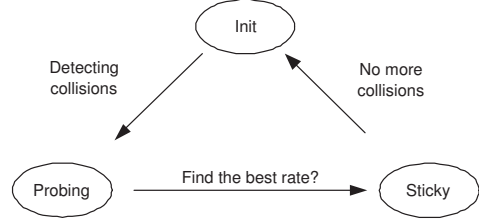


Figure 8. State machine of RPP.

in [1]. For a given value of  $T(R_i, S_i)$ , we can find a required smaller packet size  $S_{i-1}$  that satisfies the equivalence of Eq. (1) for a one level lower data rate  $R_{i-1}$ .

$$T(R_{i-1}, S_{i-1}) \equiv T(R_i, S_i) \quad (1)$$

An example of packet sizes<sup>3</sup> is as follows for other data rates when a packet size of 1500 bytes is used at the highest data rate, 54 Mbps: 1500-byte at 54 Mbps, 1327-byte at 48 Mbps, 981-byte at 36 Mbps, and so forth. However, using too small packet sizes for lower data rates could be overkill due to the increased number of too small packets. Therefore, we use a threshold of the minimum packet size (e.g., 400 bytes).

## 4.3 State Machine

When the number of collisions varies, effective use of our probing phase for finding a suitable data rate is challenging. To adaptively and efficiently invoke our probing phase, we develop a state machine that is able to dynamically model changes in collision situations. Fig. 8 shows our state machine that consists of three states: INIT, PROBING, and STICKY states. The transitions between two states are only in one direction. A node starts in the INIT state, transitions to the PROBING state, and then transitions to the STICKY state. Eventually, it gets back to the INIT state from the STICKY state.

### 4.3.1 INIT state

In this state, we can use any of the existing algorithms based on loss measurements to collect initial statistics on delivery ratio for each data rate. Typically, algorithms such as SampleRate [7] and Minstrel [18] manage a per-node measurement table that has packet loss information for all data rates. The Minstrel scheme is a simple variation of SampleRate but is more agile to channel conditions due to a smaller measurement window size than the one used in SampleRate (1 second). A node in the Minstrel scheme updates its measurement table every 100 ms. The MadWiFi driver also

<sup>3</sup>The actual packet size is rounded off in our implementation.

- 1: The Number of Packet Transmissions  $> MinPkt$  (packets) and
- 2: Packet Delivery Ratio of Selected Data Rate  $\leq MinInvProb$  (%) and
- 3:  $RSSI > MinRSSI$  (dBm)

**Figure 9. All three conditions should be satisfied for transitioning to the Probing state**

includes the Minstrel scheme as an optional multirate algorithm. In our current RPP implementation, we use the Minstrel scheme in the INIT state.

In order to effectively use our probing phase, we should be able to safely determine if a node is indeed experiencing collisions. Therefore, in our state machine, a node transitions from the INIT to the PROBING state only when all of the following three conditions, as outlined in Fig. 9, are met.

The first condition is needed as an aging stage for constructing initial stable statistics of delivery ratios across different data rates. We use the threshold  $MinPkt$  to count the number of initial transmissions including retransmissions at the physical layer. Second, when there are no collisions, the selected data rate is highly likely to have more than a 50% packet delivery ratio (practically more than 70%) unless the channel condition is very poor. As a result, when the data rate determined by a measurement-based algorithm has very low delivery ratio less than a threshold  $MinInvProb$ , a node can be safely considered to be experiencing severe collisions. Finally, to make sure that a node's decision to transition to the PROBING state is the right one, the node also examines the received signal strength indication (RSSI) provided by its wireless card. We use a threshold  $MinRSSI$  to check whether it is worth for a node to transition to the PROBING state.

### 4.3.2 PROBING state

When a node enters the PROBING state from the INIT state, it starts to examine the delivery ratio for different data rates starting from a higher data rate down to lower data rates until it finds a suitable data rate. Once a data rate is selected, the node transitions to the STICKY state where it uses only the selected data rate. The detailed description of how to determine the starting data rate and how to probe different data rates with reduced packet size in the PROBING state will be presented in the next section.

### 4.3.3 STICKY state

In the STICKY state, a node sticks to a specific data rate determined by the probing phase without referring to its packet delivery ratio table any more. However, when there are varying channel conditions as well as time-varying col-

- 1: Current Packet Delivery Ratio of Selected Data Rate  $> MinInvProb$  (%) or
- 2: Significant Change,  $SigDiff$  (%) Packet Delivery Ratio, at the current data rate or
- 3: Idle Time  $> MinIdle$  (second)

**Figure 10. Either one of the two conditions allows to transition to the INIT state**

lisions, it needs to transition back to the INIT state from the STICKY state to find a better data rate. Fig. 10 describes three possible conditions with which a node transitions to the INIT state from the STICKY state. When any of the three conditions is met, the node transitions back to the INIT state. The first condition is to check whether there is a reduction in channel-errors or collisions on its link. If the current packet delivery ratio is higher than a threshold  $MinInvProb$ , the node needs to transition to the INIT state to re-examine a suitable data rate. Second, the node also checks if there is a significant relative change in the current packet delivery ratio in comparison to the packet delivery ratio when the current data rate was selected in the PROBING state. If the relative change in packet delivery ratio is larger than a threshold  $SigDiff$  (e.g., 30%), the node transitions to the INIT state because the higher relative change indicates a clear change in channel-errors or collisions in the link. As the last condition, we consider the idle time of a node. When the node is idle and not transmitting any packets, its delivery ratio information could be obsolete. Therefore, the node must transition back to the INIT state when it is idle for a time period greater than  $MinIdle$ .

## 4.4 Probing Phase

In this section, we present our detailed probing algorithm to determine an initial data rate in the PROBING state. A node in the PROBING state first determines a starting data rate to probe different data rates with reduced packet size. Rather than blindly starting to probe the delivery ratio from the highest data rate, the node starts to probe from a data rate at which its associated delivery ratio is larger than a threshold of minimum required delivery ratio,  $MinDelivery$  (e.g., 25%–30%). After starting to probe at a suitable data rate, the node takes iterative steps for probing lower data rates.

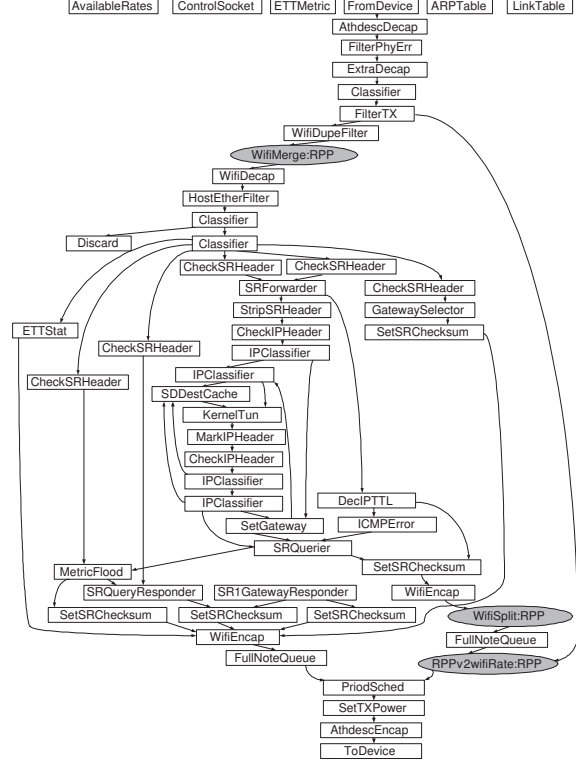
Before delving into the detailed explanation of our rate decision algorithm, we present the rationale behind our iterative steps. When a node downgrades its data rate with reduced packet size, there are three main factors to enhance its packet delivery ratio. First, downgrading of data rates itself provides a better delivery ratio due to more robust modulation schemes. Second, smaller packets typically have lower channel-error than larger packets for the same chan-

nel conditions. The last factor is related to physical capture that provides the capability to retrieve the packet with the stronger signal under certain conditions when two packets overlap in time at a receiver node (e.g., when the difference in the SNR value of the stronger packet and the weaker packet exceeds a threshold. More details can be found in [17]). In general, the packet with the stronger signal transmitted at a lower data rate has an equal or higher probability of being retrieved, compared to the one transmitted at a higher data rate. These three enhancement factors in the delivery ratio together provide a means to recognize reduced channel-error between two adjacent data rates in the presence of collisions. As a result, no further improvement in delivery ratios at a lower data rate implies that the node arrives at a data rate at which there is no more reduction in channel-error, compared to its previous probing data rate. At such data rates, most of the packet losses can be considered to be only due to collisions. Although we use several thresholds in our rate decision algorithm, the values of the thresholds are not sensitive to actual decisions as long as the values are within the right ranges. We provide a good, stable range for each threshold in our experimental environment.

In our rate decision scheme, a node examines the relative increases in delivery ratios between two adjacent data rates rather than the absolute quantities of delivery ratios. Since the node does not know its actual collision loss, it only observes a relative improvement in the delivery ratio between two adjacent data rates, that includes both channel-error and collision loss. When the current relative increase in two adjacent data rates is less than the previous one, the node stops probing. In its decision making, the node also uses a tolerance threshold of *TolGap* (e.g., 3%–5%) to make sure whether or not it can safely stop probing. When the node finds a suitable pair of two adjacent data rates by observing the relative increases in delivery ratio, it still needs to determine which of the two data rates would be a better choice. It compares the current relative difference in delivery ratio with a threshold of *MinUpperChoice*. When the difference is larger than *MinUpperChoice*, the node selects the higher data rate of the pair of data rates. Otherwise, the lower data rate is used. The *MinUpperChoice* threshold of 15%–20% results in a better selection of suitable data rates.

## 5. IMPLEMENTATION

We have implemented our RPP multirate adaptation scheme in the Emulab wireless mesh network [11]. The Roofnet [15] software that implements DSR [10] using the Click toolkit [8] is deployed on the Emulab wireless nodes in order to establish ad hoc paths. We use the values of parameters in our RPP scheme as follows: *MinPkt* = 1,000



**Figure 11. Roofnet Click configuration with our RPP multirate adaptation scheme.**

packets, *MinInvProb* = 50%, *MinRSSI* = 15 dBm, *MinIdle* = 2 seconds, *SigDiff* = 30%, *MinDelivery* = 25%, *TolGap* = 5%, and *MinUpperChoice* = 20%.

### 5.1 MadWifi Modification

Since the current MadWiFi driver (we use MadWiFi-0.9.3 [18]) is not able to recognize packets that are generated by Click (i.e., they are IP RAW packets), it simply passes the IP RAW packets to the wireless card without modifying fields of the packets that are relevant to multirate selection. We have modified the MadWiFi driver so that it can recognize Click packets, and hence set up necessary fields for multirate selection by communicating with RPP-related modules. All source code is available at [25].

### 5.2 Click Modules Implementation

Fig. 11 graphically depicts our Click configuration to implement our RPP multirate adaptation scheme. Each component (represented by a rectangle or an ellipse) in the graph is in charge of processing packets for each input port and spitting out the packets to the suitable output ports after necessary modifications. The modifications include encapsulating a new header, decapsulating the header, or chang-

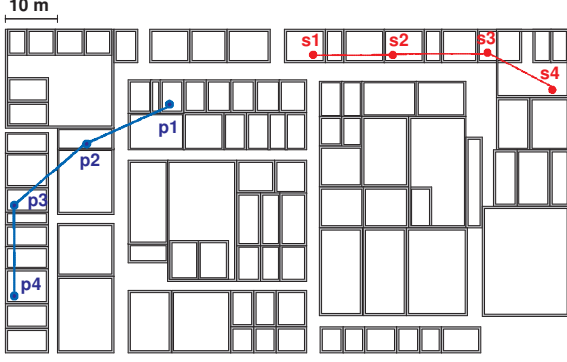


Figure 12. Two 3-hop ad hoc paths: p1-p2-p3-p4 and s1-s2-s3-s4.

ing the fields in the packets. An edge between two components represents a data flow path along which input packets are sent. In the graph, elliptical gray shapes annotated with “RPP” are modules that are newly added to implement RPP while rectangular shapes are original modules in Roofnet. The module *RPPv2wifiRate* implements most of the RPP algorithm including its packet delivery ratio measurement table and its state machine, as described in the Section 4. The two modules of *WifiSplit* and *WifiMerge* are responsible for fragmenting input packets at senders to reduce packet size and also reassembling fragments, respectively. In these modules, we use the existing fragmentation scheme provided by the 802.11 standards. The module *WifiSplit* is controlled by the the module *RPPv2wifiRate*. When the module *RPPv2wifiRate* enters the probing state, it sends a signal to *WifiSplit*. Upon receiving the signal, *WifiSplit* starts to fragment packets to a given packet size provided in the control signal. This is the way for *RPPv2wifiRate* to reduce packet size during the probing state. When *RPPv2wifiRate* is done with probing, it sends a signal to *WifiSplit* to stop fragmenting packets.

## 6. EVALUATION

In this section, we evaluate our RPP multirate adaptation scheme in the Emulab wireless mesh network [11].

### 6.1 Topology in the Emulab wireless mesh network

Fig. 12 shows two separate 3-hop ad hoc paths that we use in our evaluation. Other Emulab nodes that are not involved in the two paths, as well as the existing access points for the public use in the building are not shown in the figure. The two paths are from nodes *p1* to *p4* and from nodes *s1* to *s4*, as shown in Fig. 12. Each node on our

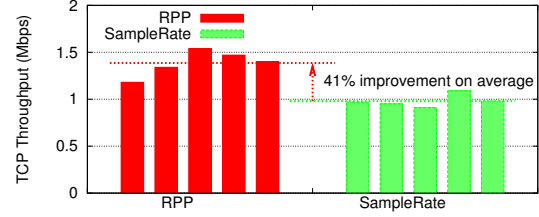


Figure 13. TCP performance comparison on the path from *s1* to *s4*.

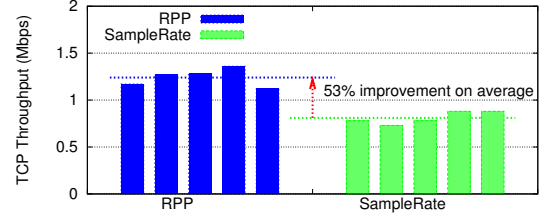


Figure 14. TCP performance comparison on the path of *p1* to *p4*.

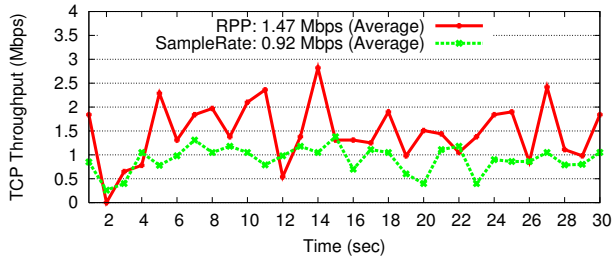
ad hoc paths uses the IEEE 802.11a PHY to communicate with other nodes. In our evaluation, we use 3-hop ad hoc paths with a significant number of collisions between adjacent nodes. To evade any possible interference from the existing public access points, we conduct our experiments during night.

### 6.2 Performance Comparison

We conduct five consecutive experiments for each multirate algorithm on the two ad hoc paths. In each experiment, we use an Iperf-2.0.2 TCP flow on a 3-hop path. Figs. 13 and 14 show the TCP performances during 30 seconds for the ad hoc paths from *s1* to *s4* and from *p1* to *p4*, respectively. In all cases, our RPP significantly outperforms SampleRate [7]. The average performance gains of the five experiments on each ad hoc path are 41% and 53% for ad hoc paths from *s1* to *s4* and from *p1* to *p4*, respectively.

To observe how RPP internally determines a suitable data rate on such 3-hop ad hoc paths, we consider the probing phase in RPP of a specific experiment on the ad hoc path from *s1* to *s4*. At the first node of the ad hoc path, measured delivery ratios for each probing data rate are as follows: 37% at 48 Mbps, 44% at 36 Mbps, and 48% at 24 Mbps. In this example, the starting probing data rate is 48 Mbps because its delivery ratio becomes greater than the threshold *MinDelivery* of 25%. The node *s1* starts probing 48 Mbps. It continues to probe lower data rates until the relative increase between two adjacent data rates stops increasing. The first relative increase in delivery ratio between 48 Mbps and 36 Mbps is 7%. The second relative





**Figure 15. An example of time-varying TCP performance on the ad hoc path of s1 to s4.**

increase between 36 Mbps and 24 Mbps is 4%, which is less than the previous relative increase of 7%. As a result, the node s1 stops probing, then selects a data rate between the previous pair of two adjacent data rates. Now, the actual relative increase is just 7%, which is less than the threshold of *MinUpperChoice* of 20%. The higher data rate from the pair (in this case, 48 Mbps) is selected. After this probing, the node enters the STICKY state with the selected data rate of 48 Mbps.

Next, we analyze how RPP performance changes with time. Fig. 15 shows instantaneous TCP throughput for every 1 second over 30 seconds. When a TCP flow starts at 0 second, the first node of the ad hoc path initially gathers delivery ratio information without invoking the probing phase. After about 1 second, it starts to probe because it suffers from a significant number of collisions. During the probing phase, the TCP performance is significantly reduced due to the overhead of the probing,<sup>4</sup> and possible packet losses due to aggressive selection of data rates. However, TCP performance soon stabilizes after the probing phase finds a suitable data rate. The performance of RPP immediately increases, and continues to achieve higher throughput than SampleRate. Due to TCP congestion control, the throughput under RPP and SampleRate fluctuates with time. However, the TCP throughput under RPP is much higher.

## 7. CONCLUSION

In this paper, we first experimentally identified the challenges in rate selection in multihop ad hoc networks. Next, we developed a novel multirate adaptation scheme, Reduced Packet Probing (RPP), that allows a sender node to effectively approximate channel-error loss in the presence of collisions. We showed that our RPP scheme, implemented in the Emulab wireless mesh network, achieves 41%–53% higher throughput in comparison to SampleRate on IEEE 802.11a-based ad hoc paths.

<sup>4</sup>The overhead of probing includes processing due to fragmentations.

## References

- [1] IEEE Std 802.11a, Supplement to Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: High-speed Physical Layer in the 5 GHz Band, 1999.
- [2] IEEE Std 802.11b, Supplement to Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed Physical Layer Extension in the 2.4 GHz Band, 1999.
- [3] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.
- [4] Ad Kamerman et al. WaveLAN-II: a High-Performance Wireless LAN for the Unlicensed Band. In *Bell Lab Tech Journal*, 1997.
- [5] Atheros Communications. <http://www.atheros.com/>.
- [6] B. Sadeghi et al. Opportunistic Media Access for Multirate Ad Hoc Networks. In *ACM MOBICOM*, 2002.
- [7] J. C. Bicket. Bit-rate Selection in Wireless Networks, Master Thesis, MIT, 2005.
- [8] Click. <http://www.read.cs.ucla.edu/click/>.
- [9] Daji Qiao et al. MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h. In *ACM MOBICOM*, San Diego, CA, 2003.
- [10] David B. Johnson et al. Dynamic Source Routing in Ad Hoc Wireless Networks. In *ACM SIGCOMM*, 1996.
- [11] Emulab. <http://www.emulab.net>.
- [12] Gavin Holland et al. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *ACM MOBICOM*, 2001.
- [13] Glenn Judd et al. Efficient Channel-aware Rate Adaptation in Dynamic Environments. In *ACM MobiSys*, 2008.
- [14] Ivaylo Haratcherev et al. Hybrid Rate Control for IEEE 802.11. In *ACM MobiWac*, 2004.
- [15] John Bicket et al. Architecture and Evaluation of an Unplanned 802.11b Mesh Network. In *ACM MOBICOM*, 2005.
- [16] Jongseok Kim et al. CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. In *IEEE INFOCOM*, 2006.
- [17] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi. An Experimental Study On the Capture Effect in 802.11a Networks. In *ACM WINTech*, 2007.
- [18] Madwifi: Multiband Atheros Driver for WiFi. <http://madwifi.org/>.
- [19] Mathieu Lacage et al. IEEE 802.11 Rate Adaptation: A Practical Approach. In *ACM MSWiM*, 2004.
- [20] Net-X at University of Illinois at Urbana-Champaign. <http://www.crhc.uiuc.edu/wireless/netx.html>.
- [21] Ns-2 simulator. <http://www.isi.edu/nsnam/ns>.
- [22] QuRiNet at University of California, Davis. <http://dave.schukin.com/spirit/quailridge/index.php>.
- [23] Starsky H.Y. Wong et al. Robust Rate Adaptation for 802.11 Wireless Networks. In *ACM MOBICOM*, 2006.
- [24] Vaduvur Bharghavan et al. MACAW: A Media Access Protocol for Wireless LAN's. In *ACM SIGCOMM*, 1994.
- [25] Wireless Mesh Networks for IEEE 802.11a/g. <http://www.cs.utah.edu/~jcpark/mywmn.html>.