Opportunistic Routing for Interactive Traffic in Wireless Networks

Tianji Li*†, Douglas Leith*
*Hamilton Institute
National University of Ireland Maynooth, Ireland
Email: {tianji.li, doug.leith}@nuim.ie

Lili Qiu†
†Computer Science Department
University of Texas at Austin, USA
Email: lili@cs.utexas.edu

***Abstract*—To take advantage of the broadcast nature of wireless communication, a number of opportunistic routing techniques have recently been proposed. In order to manage the extra signaling overhead associated with operation of the opportunistic routing, these schemes work in terms of 'batches' consisting of multiple packets. While these opportunistic techniques can dramatically improve the system performance, use of batches means that they are best suited to UDP traffic. In the Internet and wireless networks, however, the vast majority of the traffic is interactive[1] (e.g., up to 80-90% is TCP). To support interactive traffic opportunistically and efficiently, we introduce a novel scheme called RIPPLE. In the RIPPLE scheme, an expedited multi-hop transmission opportunity mechanism ensures low signaling overhead and eliminates re-ordering, and a two-way packet aggregation technique further reduces overhead. We implement the RIPPLE and related schemes[2] in NS-2 and compare their performance for long- and short-lived TCP transfers and VoIP traffic over a wide range of network conditions, including varied wireless channel states, levels of regular and hidden collisions, and geographic locations of stations derived from measurement studies (i.e., the Wigle and Roofnet topologies), etc. Our results show that the RIPPLE scheme consistently delivers 100% – 300% performance gains over other approaches.**

***Index Terms*—Medium access control (MAC), Opportunistic Routing, IEEE 802.11, Wireless Multi-hop Networks, Wireless LANs (WLANs), Wireless Mesh Networks.**

I. Introduction

Wireless communication is inherently broadcast in nature. A unicast transmission can be heard not only by the target receiver, but also by every other station in the neighborhood of the transmitter. Indeed, these stations (called forwarders hereafter) typically decode all transmissions they hear and then drop transmissions for which they are not the intended recipients. To take advantage of this broadcast property, it is appealing to let forwarders help relay overheard traffic. This can be expected to yield significant performance gains when, for example, the link between the sender and the receiver is poor, but the links between the forwarders and the sender, and the links between the forwarders and the receiver are good. This idea is often referred to as opportunistic routing in the literature (e.g., [14] [7] [8] [25]).

A key issue in opportunistic routing schemes is the signaling overhead associated with the routing of each packet. In classical predetermined routing, once the routing tables have been constructed there is no additional per packet signaling overhead. However, in opportunistic schemes, multiple forwarders typically overhear a packet transmission and, due to the stochastic nature of channel noise, this set of receivers varies from packet to packet. It is thus necessary for the forwarders to acknowledge whether they hear a particular packet. One straightforward acknowledging approach is for the forwarders (and the receiver) to transmit a MAC acknowledgment (ACK) on receipt of a packet and for these MAC ACKs to be scheduled sequentially in order to avoid collisions between the ACK transmissions. This approach is used in the early version of ExOR [6], which we refer to as preExOR to distinguish it from the later work in [7]. Clearly, the sequential acknowledging of the preExOR scheme can be inefficient if there are many forwarders. For efficient use of network resources it is important to minimize this per packet signaling overhead. ExOR [7] mitigates the overhead caused by sequential one-hop ACKs by working in terms of batches and using end-to-end ACKs. In MORE [8], the work in [15] and CodeOR [19], ExOR is extended with network coding and efficient coordinating ideas. The MCExOR scheme [25] proposes an approach whereby forwarders can prematurely stop waiting for MAC ACKs from the receiver and high ranked forwarders, and send their MAC ACKs.

While these opportunistic techniques can dramatically improve the system performance, none of them considers supporting interactive traffic such as TCP and VoIP. Consideration of this type of traffic is however important. In fact, the vast majority (up to 80%-90% [24] [23]) of network traffic is TCP, and VoIP is becoming more and more popular. Interactive traffic is different from UDP. In particular, TCP flows are two-way in nature and in each direction the number of in-flight packets, which is controlled by the congestion control algorithm of TCP, varies over time; VoIP is used by at least two simultaneous callers. Existing opportunistic schemes which make use of a fixed batch size to manage the per packet signaling overhead are not suited to carrying such traffic (where the number of packets in flight is frequently much smaller than the typical batch sizes). This is acknowledged by the authors of [7]. Since MORE [8] and the work in [15]

[1]By interactive traffic, we mean traffic such as TCP/VoIP in which there is a feedback process between the involved sender(s) and receiver(s).

[2]Namely predetermined routing, shortest path routing, the early version of ExOR for supporting interactive traffic, MCExOR and a 802.11n-like single-hop packet aggregation scheme called AFR.

focus on network coding extensions to [7], they inherit similar issues. Approaches using per-packet ACKs (i.e., preExOR and MCExOR) are not effective due to the high signaling overhead and also re-ordering issues (see Section II). Forwarding interactive traffic opportunistically is thus challenging.

To tackle the challenge of supporting interactive traffic opportunistically we design a novel scheme called RIPPLE in this paper. In the RIPPLE scheme, an expedited multi-hop transmission opportunity (mTXOP) mechanism ensures low signaling overhead and eliminates re-ordering; a two-way packet aggregation technique further reduces overhead (see Section III for details). We implement the RIPPLE and related schemes (namely predetermined routing, shortest path routing, preExOR, MCExOR and a 802.11n-like single-hop packet aggregation scheme called AFR[17]) in NS-2 and compare their performance for long- and short-lived TCP transfers and VoIP traffic over a wide range of network conditions, including varied wireless channel states, levels of regular and hidden collisions, and geographic locations of stations derived from measurement studies (i.e., the Wigle and Roofnet topologies), etc. Our results show that the RIPPLE scheme consistently delivers significant performance gains over other approaches, i.e., 100% – 300% throughput improvement is achieved.

We envision that the proposed scheme will most likely be useful in infrastructure WLANs (e.g., those considered in [4]) where clients can help each other communicate with the AP efficiently, and in multi-hop networks which provide end-users Internet like services.

II. Motivation

A routing protocol normally consists of three parts: route discovery, packet forwarding and route maintenance. Denote the sender, the receiver and the set of forwarders by *S*, *R* and $F = F_1, \ldots, F_n$ (where n is the number of forwarders) in a wireless multi-hop network. For an opportunistic routing protocol, the task of route discovery involves determining a selection of the members of the forwarder set F. Existing routing schemes (e.g., ExOR [7]) leverage heuristic methods such as ETX [12].

In this paper we focus on the second task, i.e., packet forwarding. Current forwarding techniques can be classified into two sub categories: predetermined and opportunistic forwarding.

A. Predetermined Forwarding

Forwarding sequences in this category are predetermined in nature, i.e, when the sender *S* transmits a packet, only the first forwarder F_1 is permitted to receive it (other stations simply drop the packet). When the first forwarder F_1 forwards, only the second forwarder F_2 receives the packet, and so on. The forwarding path followed by a packet is thus fixed and is updated periodically.

B. Opportunistic Forwarding

Opportunistic forwarding methods allow the receiver and all the forwarders in F to receive transmissions from the sender S, and all of the forwarders help relay overheard transmissions with the aim of improving system efficiency. We consider two of the existing methods (preExOR and MCExOR) in this section which do not use batches and so are potentially suitable for supporting interactive traffic (as discussed in Section I).

In the preExOR scheme, forwarders send MAC ACKs sequentially to avoid collisions amongst the ACKs after receiving a packet from S. Thus, after the sender transmits a data packet it then defers for a period that is sufficient to allow the receiver and all the forwarders to transmit their ACKs, see details in [6].

The MCExOR scheme uses a compressed acknowledging mechanism, in which a forwarder of rank i waits for i SIFS intervals before transmitting a MAC ACK. If it detects an ACK transmission during its waiting period it will not transmit its ACK since the ACK reception indicates that a higher ranked forwarder has received the packet.

C. Comparison

In the context of opportunistic routing, the receiver R is able to hear from the sender S but the link quality between them is normally low. The link quality between the sender S and the forwarders in F and that between the forwarders in F and the receiver R is normally high. Thus, a properly designed opportunistic routing scheme should be able to seize the opportunities when a transmission from S is directly heard by R, or by high priority forwarders that are close to R, so as to reduce the number of transmissions needed to forward packets, and thereby improving system efficiency.

However, this performance gain is often not achieved when using the preExOR and MCExOR schemes for supporting interactive traffic such as TCP and VoIP. There are two reasons for this inefficiency: signaling overhead and packet reordering.

1) Signaling Overhead: The preExOR and MCExOR schemes frequently incur higher signaling overhead than predetermined schemes. Specifically, transmissions from the sender S are received with the highest probability by the first forwarder F_1, and forwarding from F_1 is most frequently received by the second forwarder F_2, and so forth, then both the opportunistic and predetermined routing schemes tend to use this same route (i.e., $S \rightarrow F_1 \rightarrow \ldots \rightarrow F_n \rightarrow R$). That is, even if opportunistic routing methods are used, the most probable route is the same as that used by predetermined routing. Opportunistic transmissions in this case have a higher signaling overhead than predetermined routing (due to the need to identify the set of forwarders that have received each packet transmission) and so network throughput can be significantly lower than when predetermined routing is used.

In more details, let the time to perform random backoff be $T_{backoff}$, to transmit MAC ACKs be T_{ACK}, to send the physical layer header be T_{phyhdr}, to defer SIFS be T_{SIFS} and to defer DIFS be T_{DIFS}, respectively. Using predetermined routing methods (with 802.11 DCF[3] at the MAC layer), for a

[3]Note that the preExOR and MCExOR schemes are extensions based on 802.11 DCF.

packet that is relayed by $n-1$ forwarder(s) to be received by the receiver, it takes $n(T_{backoff}+T_{DATA}+T_{DIFS}+T_{SIFS}+T_{ACK}+2T_{phyhdr})$ where one T_{phyhdr} is used for sending the data packet, another for the ACK. While due to the use of sequential ACKs (and the compressed sequential ACKs), it takes $n(T_{backoff}+T_{DATA}+T_{DIFS}+T_{phyhdr})+\sum_1^n(T_{ACK}+T_{SIFS}+T_{phyhdr})$ and $n(T_{backoff}+T_{DATA}+T_{DIFS}+T_{ACK}+2T_{phyhdr})+\sum_1^n T_{SIFS}$ in the preExOR and MCExOR schemes respectively. Using preExOR and MCExOR therefore lead to longer channel usage time. For example, in Fig. 2 we illustrate transmission timeline of two packets for flow 1 in the topology in Fig. 1. The predetermined route used for flow 1 is called PRR for ease of explanation (in which packets of flow 1 follow the route $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$). Comparing PRR with preExOR and MCExOR, we can see that in this example the overhead incurred by the preExOR scheme is $6*(T_{ACK}+T_{SIFS})$ longer than with PRR. Due to the use of compressed slots, MCExOR takes $6*T_{ACK}$ less time than preExOR, but still $6{*}T_{SIFS}$ intervals longer than PRR. That is, for the most probable transmission sequence the preExOR and MCExOR schemes incur extra signaling overhead over PRR due to the signaling requirements associated with operation of the opportunistic routing.

2) Packet Re-ordering: The foregoing analysis relates to the most probable transmission sequence. Of course, there are many other possible transmission sequences which would be shorter and therefore incur lower overhead. To compare performance, we implemented the preExOR and MCExOR schemes and compared them with predetermined methods.

We observed that even if other transmission sequences are also possible, the performance of preExOR and MCExOR forwarding is still not comparable with that of predetermined routing. For example, consider a TCP flow from station *0* to station *3* which lasts for 10 seconds with the parameters in Table I. The measured throughput is 0.001, 6.7, 5.9 and 5.85 Mbps with the Shortest Path Routing (SPR, i.e., directly from station *0* to *3*), PRR ($0 \rightarrow 1 \rightarrow 2 \rightarrow 3$), preExOR and MCExOR schemes, respectively. Please refer to our technical report [18] for more details.

On closer inspection of the simulation data, in addition to the signaling overhead discussed above, the degradation in throughput of preExOR and MCExOR relative to predetermined routing is affected by a second factor. Namely, we find that packet re-ordering happens frequently with both the preExOR and MCExOR schemes. For example, with the preExOR scheme, of the 10766 TCP packets received by the receiver 2862 are out of order, i.e., 26.58% of packets are re-ordered. With the MCExOR scheme, 27.9% packets are out of order (3122 packets out of 11191 packets). The congestion control algorithm in TCP treats re-ordering in a similar way to packet losses, reducing the send rate and so reducing the connection throughput. Re-ordering in preExOR and MCExOR occurs due to the random backoff mechanism of 802.11 and the unpredictable channel state. To see this, consider a situation where the sender has two packets i and $i+1$ to send. Suppose it sends packet i first which is received by forwarder j but not by the receiver. Both the sender and forwarder j then initiate a random backoff to win the channel access, but the sender will sometimes choose a shorter random backoff time than forwarder j and so transmit packet $i+1$ before forwarder j transmits packet i. If packet $i+1$ is heard by the receiver, then re-ordering will occur.

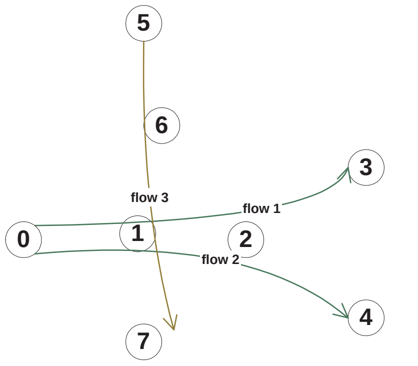


Fig. 1. A multiple-flow topology. There are three flows altogether in this topology. Flows 1 and 2 share stations 0, 1 and 2, and flow 3 intersections with the other flows at station 1.

T_{SIFS} (μs)	16
Idle slot duration (μs)	9
Packet size (bytes)	1000
PHY data rate (Mbps)	216
PHY basic rate (Mbps)	54
Interface queue (packets)	50
T_{phyhdr} (μs)	20

TABLE I
MAC/PHY PARAMETERS USED IN THIS PAPER.

III. THE RIPPLE SCHEME

A. The Main Idea

For opportunistic routing protocols (as discussed in the previous section), performance for supporting interactive flows is mainly affected by two key issues: packet re-ordering and the per packet signaling overhead. In the RIPPLE scheme, these two issues are resolved in the following manner.

1) Resolving Re-ordering: The cause of packet re-ordering, as introduced in the last section, is the time difference between transmissions of new packets by the sender, and transmissions of old packets by the forwarders. To solve this issue, we do not let the forwarders cache any heard frames while still letting them help forward transmissions (This is thus an idea similar to that proposed in [22] for next generation mobile ad-hoc networks). That is, we design an atomic operation between the sender and the receiver within which re-ordering can be completely eliminated. We call this kind of operation a multi-hop transmission opportunity[4] (mTXOP) and describe the details in the following steps.

[4]Recall that a transmission opportunity in 802.11 consists of a DIFS interval, a backoff period, a data transmission, a SIFS interval and a MAC ACK transmission

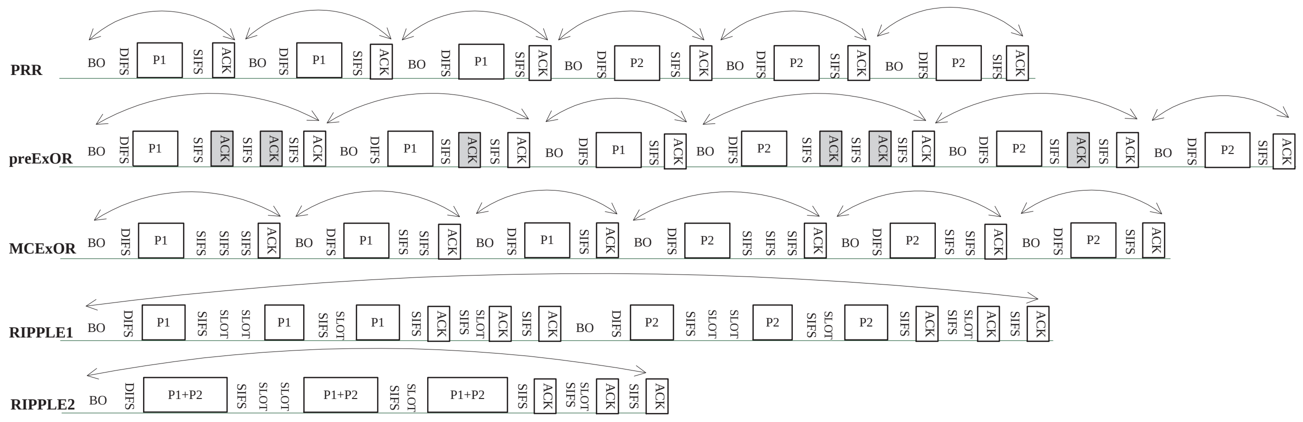


Fig. 2. Transmissions of two packets (P1 and P2) with predetermined route (i.e., $0 \rightarrow 1 \rightarrow 2 \rightarrow 3$), preExOR, MCExOR and RIPPLE. BO is the abbreviation of backoff. In the preExOR scheme, shadowed ACKs indicate that the sender is waiting for an ACK which is not transmitted. Each arc line indicates one transmission opportunity.

- *Multi-hop Transmission Opportunity*. Denote the highest priority forwarder to be forwarder 1, the next highest priority forwarder be forwarder 2, and so on. In the RIPPLE scheme, the receiver acknowledges reception of a frame after a $T = T_{SIFS}$ time, where T_{SIFS} is the time for a SIFS duration [1]. Forwarder i $(i \geq 1)$ relays a received data frame only after detecting the channel to be idle for a $T = i \times T_{Slot} + T_{SIFS}$ time, where T_{Slot} is the time for a slot duration [1]. This results in a prioritized opportunistic acknowledging scheme whereby the highest priority forwarder that receives a data frame relays the packet while lower priority forwarders defer and make no transmission (See Sections III-B1 and III-B2 for details about priority assignments.). Therefore, high priority forwarders can help relay whenever they overhear the transmissions, thus improving performance. Note that in MCExOR, a similar premature waiting mechanism is used, after which the MAC ACK will be sent to the sender. Whilst in our scheme, we forward the overheard data frame towards to destination.
- *Two-way Opportunistic Forwarding*. On receiving a data frame the receiver replies with a MAC ACK. Since MAC ACKs are important for ensuring multi-hop performance, we let the forwarders help relay MAC ACKs in a similar manner to data frames, so that the MAC ACKs will be received by the sender with high probabilities. That is, forwarder i $(i \geq 1)$ relays a received MAC ACK frame after detecting the channel to be idle for a $T = (i-1) \times T_{Slot} + T_{SIFS}$ time. Since there is no acknowledgment for the MAC ACKs, forwarders defers one slot less time for relaying a MAC ACK than for relaying a data frame.
- *Multi-hop Retransmission*. Forwarders do not cache any, and only relay heard transmissions at most once, i.e., if a forwarder hears a data (or a MAC ACK) frame but does not hear the due transmissions from higher priority stations, it will start relaying, otherwise it discards the heard frame. Retransmission of lost frames is thus performed on a multi-hop basis, with the sender retransmitting when it does not receive a MAC ACK for a transmitted frame. Thus, re-ordering caused by relaying from forwarders will never happen.

We revisit the example in Section II using the mTXOP mechanism. As before, assuming for illustrative purposes the same transmission order as in the PRR scheme, the transmission timeline for packets *P1* and *P2* is shown in Fig. 2 (see RIPPLE1). When packet $P1$ is transmitted by station *0* it is received by station *1* but not by *2* or *3*. Station *1* waits for one SIFS and 2 slot intervals (the SIFS for possible ACK transmissions from station *3*, the first slot for from station *2*, and the second slot to provide time to turn itself from receiving to sending state) before forwarding $P1$. After hearing *1*'s transmission, station *2* relays in a similar way but defers one SIFS and one slot as it is only one hop from the destination. Finally, *P1* arrives at station *3*. The MAC ACK is then sent and forwarded in a similar way and then the same sequence repeats for Packet *P2*.

2) Mitigating Overhead: Although we can guarantee re-ordering free using the mTXOP mechanism, a similar to the preExOR and MCExOR schemes overhead is incurred. To mitigate the overhead, we propose a two-way packet aggregation mechanism which works as follows.

- When the sender (re)transmits, we allow multiple packets (each protected with a separate CRC of its own) to be aggregated in the (re)transmitted frame[5]. Thus, overhead is incurred only once for the large frame, while without aggregation, overhead has to be repeated for at least the number of aggregated packets times. For the above example, using the packet aggregation (RIPPLE2 in Fig. 2) leads to approximately 50% overhead reduction in comparison to the non-aggregated version (i.e., RIPPLE1 in Fig. 2). As per [2], [17], we select 16 as the maximum

[5]In the RIPPLE scheme, multiple packets can be transmitted in a single frame. To distinguish, we define a packet as what the MAC receives from the upper layer, a frame as what the MAC transfers to the PHY layer.

number of packets that can be aggregated into a frame.

- Aggregation can be performed in a bi-directional manner, i.e. if there are data packets waiting to be transmitted from the receiver to the sender, the receiver also aggregates packets into large frames. This seemingly simple mechanism can lead to significant efficiency gains for two-way flows such as TCP, where TCP ACKs in the reverse direction have to be sent.
- If there is local traffic at forwarders, a forwarder can aggregate local packets and relayed packets in order to save bandwidth.

B. Remarks

1) Forwarder Lists: The selection of forwarder lists (and the priority assignments based upon it) belongs to the route discovery task of a routing protocol. Existing routing schemes (e.g., ExOR [7]) leverage the forwarder selection method of ETX [12]. Since this paper focuses on packet forwarding, selection of forwarders is out of the scope of this current work. However, the design of the RIPPLE frames allows any forwarder lists (and thus priority assignments) to be supported. As we will show in Section IV, given a pre-selected path, using the RIPPLE scheme can always achieve improved performance.

2) Priority Assignments: A station, on hearing a frame, checks the forwarding list to decide whether it is a forwarder. For forwarders, we use an implicit rule to assign their priorities. In particular, all stations know that the forwarding list is located between the MAC header and the frame body. We mandate that the identities (and thus the forwarders) that are closer to the MAC header have higher priorities. The destination is always the highest priority forwarder, and thus the closest one to the MAC header.

3) How mTXOPs Break: The mTXOPs used in the RIPPLE scheme are potentially longer than transmission opportunities in other schemes. If such mTXOPs are stopped prematurely and frequently, performance will be negatively impacted. We argue however that while this situation can happen, the impact is likely to be minor.

Broken mTXOPs can be due to channel noise and collisions. Opportunistic routing schemes are mainly designed for mitigating the former issue, i.e., when the link between the sender and the receiver is error prone, and links between the sender and forwarders and the links between the forwarders and the receiver are in good states.

As for the latter issue, collisions may be classified into intra-path or inter-path collisions. By intra-path collisions we mean collisions between stations (including the sender and the receiver) that are on the path from the sender to the receiver, while by inter-path collisions we mean the collisions between stations on and off the path.

For intra-path collisions, there are two issues to consider. First, two on-path stations' forwarding transmissions can overlap when stations can not overhear the due transmissions (for example a lower priority forwarder does not hear a relay transmission by a higher priority forwarder). This is essentially a form of hidden terminal effect acting between on-path stations. To mitigate such effects we can use a small number of forwarders (see remark (4) below), and our results indicate that using up to 5 forwarders can ensure that over a wide range of network conditions, this type of collision does not have a major impact. Second, there may be local traffic at a forwarder waiting to be transmitted. To reduce collisions between relayed and local traffic, when relaying a heard frame, a forwarder aggregates local packets (if the frame is not large enough) so that both multi-hop and local packets are sent in one transmission.

Inter-path collisions can be regular or hidden collisions: a regular collision happens when two senders that can hear each other start sending at the same time, whilst a hidden collision occurs when two senders that are not able to hear each other start simultaneously. Very frequent regular collisions can lead to low performance for all considered schemes in this paper. But one can expect that this will happen infrequently for interactive flows such as TCP due to its self-adaptability. Hidden traffic can also have major effects on all the schemes when for example the hidden traffic is nearly saturated. Fortunately, recently measurement studies in real networks show that hidden collisions only account for less than 5% of all losses (Fig. 9 in [11]). In both regular and hidden collision cases, as shown in Section IV-B, performance achieved by the RIPPLE scheme is always the best of the schemes considered.

4) Maximum Number of Forwarders: Using a small number of forwarders can restrict the potential cost of intra- and inter-path collisions. If the number of forwarders is large, collision can become frequent in a manner which is similar to the effect observed in [13] (i.e., if there are too many forwarders, collisions can be so frequent that final performance is worse than that of using single path routing approaches.).

We comment that the number of forwarders refers to the actual number of forwarders used on a given path, not to the number of potential forwarders in the network. The forwarder selection method used (e.g., [7] [12]) ensures that a short path is selected, i.e., not all potential forwarders will be chosen as forwarders.

We leave the maximum number of forwarders as an open design parameter. In this paper, we mostly use 5 as the maximum forwarders since in a wide range of network conditions we did not observe that the system performance was impacted by this value. In Section IV-C we also introduce results for up to 7 hops.

5) Self-Adaptability to Traffic Demands: Packet aggregation requires the availability of packets to form into a large frame. While some form of suitable wait mechanisms before transmissions seems a good idea at first glance, we note that real world traffic can exhibit complex bursty behaviors which make the design of an effective waiting scheme difficult. Therefore, in this paper, we leverage the zero waiting mechanism proposed in our previous work for single hop packet aggregation [17]. With zero waiting, the sender simply aggregates and transmits the available packets (if it is not exceeding the maximum allowed number) waiting in the sending queue, with no additional waiting time introduced. This simple scheme turns out to adapt automatically to changing network conditions. Specifically, when the current level of efficiency is too low for the offered load, a queue backlog will develop which in turn induces larger frames to be transmitted and so increases efficiency. If the incoming traffic subsides, smaller frame sizes will be automatically selected as queue backlogs tend to disappear too.

6) MAC Layer Queues: Two queues are maintained at the MAC layer: a sending queue (Sq) at the sender and a receiving queue (Rq) at the receiver[6]. With the Sq, we can i) aggregate packets into large frames, ii) keep packets until they are acknowledged. With the Rq, we can keep incoming packets if they arrive out of order, i.e., we only pass in-order packets to the upper layer. Note that a new type of packet re-ordering (i.e., it is not the same as in the preExOR and MCExOR schemes) may happen without the Rq due to the use of packet aggregation. In particular, with packet aggregation, multiple packets can be transmitted in a large frame. Due to channel noise, some packets in a same frame may be corrupted but others are correctly received by the receiver. If the corrupted packets are those that arrive at the sender earlier than the correct ones, the Rq is required to cache the correct packets temporarily, waiting for the corrupted packets to be retransmitted.

7) Piggy-back on MAC ACKs: After receiving a data transmission from the sender, the destination first replies with a MAC ACK, then sends a data packet back to the sender if there are any in the Sq of the receiver. Potentially, it is possible to aggregate the MAC ACK and new data packet into a same frame in order to reduce the transmission overhead. However, such aggregation creates signaling difficulties. Namely, if we piggy-back data packets on MAC ACKs, then the sender station does not know in advance what timeout period to use before triggering a retransmission.

8) Co-existing With 802.11: Similarly to 802.11n, the RIPPLE scheme basically uses long transmissions if possible while keeping other aspects of 802.11 (e.g., backoff, DIFS, etc.) unchanged. It thus can co-exist safely with 802.11.

IV. Evaluation

We implemented the RIPPLE and related schemes (namely predetermined routing, shortest path routing, preExOR, MCExOR and a 802.11n-like single-hop packet aggregation scheme called AFR[17]) in NS-2. All results presented are averages over multiple runs. Due to packet re-ordering and signaling issues introduced in Section II, the performance of the preExOR and MCExOR schemes is always worse that with predetermined routing schemes, and so we do not introduce their results in this section.

To evaluate performance when the channel is error prone and both intra-path and inter-path collisions happen (see Section III Remark (3)). We use a combination of frame and bit error models.

[6]No queue is used at the forwarders.

	Flow 1	Flow 2	Flow 3
ROUTE0	0 1 2 3	0 1 2 4	5 6 1 7
ROUTE1	0 1 3	0 1 4	5 6 7
ROUTE2	0 2 3	0 2 4	5 1 7

TABLE II
THE USED PATHS FOR THE TOPOLOGY IN FIG. 1.

For the frame models, the Shadowing model of NS-2 is used in which frame losses are proportional to the distance between stations. Note that this model assumes that losses between the sender and different forwarders are independent. Therefore, intra-path and inter-path collisions occur in a random manner. The chosen Shadowing model parameters are: path loss exponent 5, shadowing deviation 8, transmission power 281 mW.

Using bit error models is important (and fair when comparing with other schemes) for studying the performance of packet aggregation schemes, since partial retransmissions are normally used for them. For this aim, we use the independent and identically distributed (i.i.d.) BER model as in [17]. Note that replacing i.i.d. with other BER models (e.g. Gilbert-Elliot model) is straightforward and dose not impact the comparison in this paper (see [17] for details on this). Since TCP congestion control views packet losses as an indicator of congestion, TCP throughput is strongly dependent on the link loss rate (e.g., [9] [10]) and too high a loss rate may then prevent high utilization of the wireless channel. To study the impact of channel noise therefore, we use a BER of 10^{-5} and 10^{-6} to simulate a 'noisy' and a 'clear' channel state, respectively.

We first present the performance measured on the topology in Fig. 1 (in Section IV-F we present results for two larger topologies). There are initially three flows in this topology: the sources of flows 1, 2 and 3 are station 0, 0 and 5, while the destinations are 3, 4 and 7. We consider three sets of predetermined routes for these flows and call them ROUTE0, ROUTE1 and ROUTE2. In Table II we list these routes. For example, if we use ROUTE0, flow 1 would have a route from 0 to 3 via stations 1 and 2. In the results figures (Figs. 3 and 4) we report results when only flow 1, both flows 1 and 2, and all flows 1, 2 and 3 are activated at the same time, respectively.

For predetermined routing methods (i.e., using routes ROUTE0, ROUTE1 and ROUTE2), we use at the MAC layer the standard 802.11 DCF and the AFR scheme [17] for packet aggregation (which is similar to 802.11n proposals). The AFR scheme is a packet aggregation extension of 802.11 DCF, and the maximum number of packets that can be aggregated into large frames is 16 which is the same as that used in the RIPPLE scheme. In this way, we can compare AFR and RIPPLE on a fair basis and distinguish clearly the source of observed throughput improvements, i.e., distinguish improvements due to packet aggregation and due to mTXOPs.

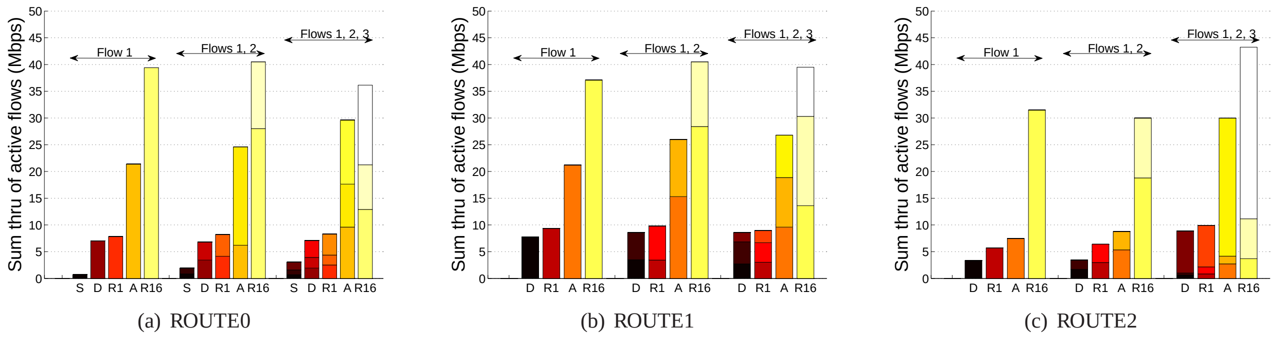


Fig. 3. Measured throughput in Mbps for the topology in Fig. 1. 'S', 'D', 'R1', 'A' and 'R16' represent the SPR (directly from station *1* to station *3*) with DCF, 802.11 DCF, RIPPLE without packet aggregation, AFR and RIPPLE (with packet aggregation) schemes respectively. BER is 10^{-6} and other parameters listed in Table I.

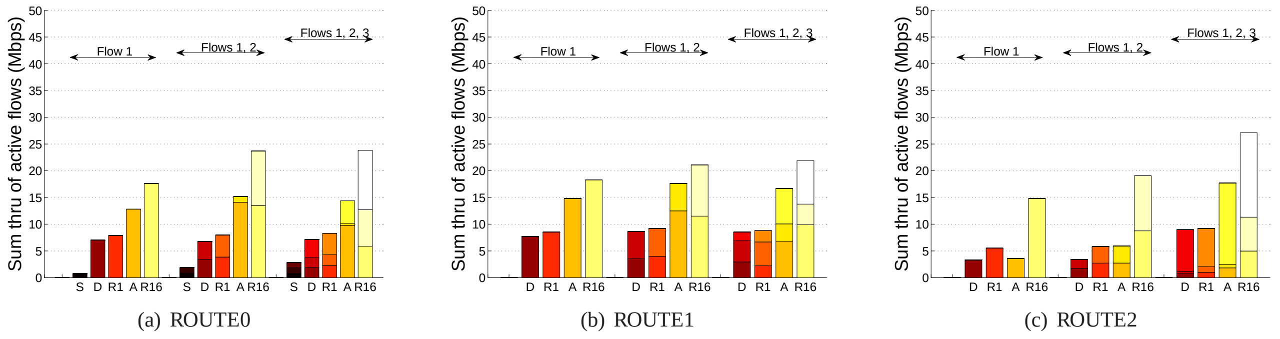


Fig. 4. Measured throughput in Mbps for the topology in Fig. 1. 'S', 'D', 'R1', 'A' and 'R16' represent the SPR (directly from station *1* to station *3*) with DCF, 802.11 DCF, RIPPLE without packet aggregation, AFR and RIPPLE (with packet aggregation) schemes respectively. BER is 10^{-5} and other parameters listed in Table I.

A. Long-lived TCP Transfers

We first consider long-lived TCP transfers, which persistently send traffic during the simulation time. In order to simulate a situation where mTXOPs are necessary, we locate the stations and tune carrier/receiving ranges in such a way that one-hop routing is inefficient. That is, a single TCP flow's throughput with SPR (directly from station *0* to station *3*) is 0.76 Mbps when the BER is 10^{-6}, using the parameters in Table I. But, when ROUTE0 is used, throughput increases to 7.04 Mbps (see Fig. 3 for results of SPR and ROUTE0 running multiple flows).

With this network configuration, both intra- and inter-flow collisions can happen. Namely, we can simulate the performance when collisions due to activities from both hidden and non-hidden stations. Further, use of the BER model allows us to simulate channel noise which is independent to collisions.

We first show the results of the RIPPLE scheme when packet aggregation is turned off, and of the AFR scheme from [17]. Namely, we would like to see the performance of pure mTXOP without aggregation, and pure aggregation without mTXOP. The results when the ROUTE0 route is used are shown in Fig. 3(a). It can be seen that the pure mTXOP (marked as R1) and pure aggregation (marked as A) schemes achieve slightly higher and around twice more throughput than the 802.11 DCF (marked as D).

We then turn on the packet aggregation of the RIPPLE scheme. As seen in Fig. 3(a), the RIPPLE scheme (marked as R16) is able to take advantage of both mTXOP and packet aggregation and the throughput achieved is approximately the sum of both. That is, the effectiveness of the RIPPLE scheme is due to both mTXOPs and packet aggregation. But, with either of these individually, high performance can not be achieved.

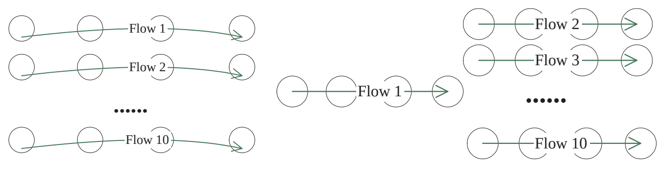
To show that the RIPPLE scheme is able to support various selection of forwarder lists (and thus priority assignments), in Figs. 3(b) and 3(c) we illustrate results when the ROUTE1 and ROUTE2 routes are used. We observe that regardless of the routes used, the RIPPLE scheme consistently outperforms the other approaches.

Note that the performance of RIPPLE is similar on both the ROUTE0 and ROUTE1 routes, while a significantly lower throughput is achieved on ROUTE2. Thus, either ROUTE0 or ROUTE1 will likely be preferred if heuristic routing metrics such as ETX [12] are used.

The performance when the BER 10^{-5} is illustrated in Fig. 4 where a similar trend is observed when comparing the RIPPLE scheme with others.

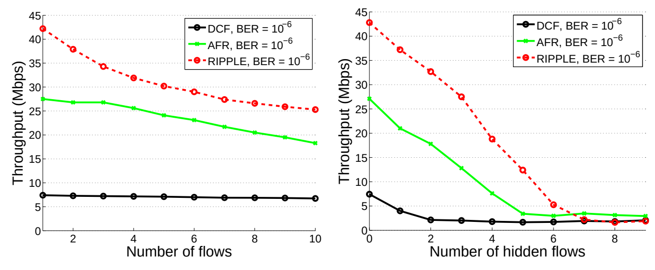
B. Effects of Regular and Hidden Collisions

Excessive regular/hidden collisions can have a large impact on the performance of any CSMA/CA based protocols. To



(a) For regular collisions (b) For hidden collisions

Fig. 5. Topologies used for illustrating the impact of regular and hidden collisions.



(a) Regular collisions (b) Hidden collisions

Fig. 6. Measured throughput in Mbps for regular and hidden collisions. BER is 10^{-6} and other parameters listed in Table I.

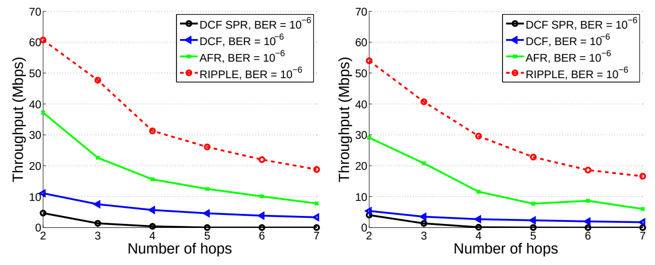
show the effect of regular collisions, we use the topology shown in Fig. 5(a) where all the flows are in hearing and receiving ranges of each other (but note that due to the use of the Shadow model, infrequent inter- and intra-path hidden collisions can still happen). The results (in Fig. 6(a)) show that the sum throughput of all flows drops as expected when the number of flows increases, with the RIPPLE scheme outperforming the DCF and AFR schemes.

As for the hidden collisions, we use the topology shown in Fig. 5(b) where the senders of flows 2–10 are completely hidden to the sender of flow 1 (but not so to the forwarders). Therefore, when the hidden traffic load becomes heavy, flow 1 can be throttled which is what we see in Fig. 6(b) where the throughput of flow 1 is plotted as the number of hidden flows (flows 2–10, each sending 5×10^6 packets during the simulations) is increased from 0 to 9. Again, the RIPPLE scheme behaves better for less than 7 hidden flows. When there are 7, 8 and 9 hidden flows, the performance of RIPPLE is slightly worse than that of the other two. This is because the hidden flows of RIPPLE use mTXOPs which can cause longer hidden collisions than when using DCF and AFR. Note that in this extreme region, no scheme can achieve more than 3 Mbps throughput even though the physical layer rate is 216 Mbps.

C. Maximum Hops with Cross Traffic

In this paper, we mostly use 5 as the maximum number of forwarders. In this section, we briefly introduce results when up to 7 hops are used since the maximum reported hops that we can find in the literature is 7 (see [7]).

For this aim, we use a line topology and increase the line length from 2 to 7 hops with and without a 3-hop flow (sending 5×10^6 packets) intersecting it. As expected (see Figs. 7(a) and



(a) Without Cross Traffic (b) With Cross Traffic

Fig. 7. Throughput in Mbps for up to 7 hops. BER is 10^{-6} and other parameters listed in Table I.

7(b)), the throughput drops with increased distance with again the RIPPLE scheme achieving the best performance. Note that when the sender/receiver are 6 and 7 hops apart, they are not able to hear each other. This means that in the RIPPLE scheme, direct sending/receiving between the two ends of the path is not possible, and thus its performance depends entirely on the forwarders help. Results show interestingly that the forwarders do well.

D. Short-lived TCP Transfers: Web Traffic

In this section, we present results for short TCP transfers which mimic realistic web traffic ([21]). Web traffic consists of ON/OFF periods. During the ON time, a web user visits some web pages, whilst in the OFF time, the user is reading what he/she just downloaded. To run the simulations in a realistic manner, traffic generated in ON periods should be Long-Range Dependent, i.e., resembles the aggregation of many ON-OFF senders with heavy-tailed ON periods. In this paper, a transfer sized with a Pareto distribution with mean 80KB and shape parameter 1.5, is used when the traffic is in ON time. While during the OFF periods, no traffic is generated. The length of the OFF periods follows an exponential distribution with mean duration of one second.

The topology used for web traffic is the same as that used for long-lived TCP transfers (i.e., Fig. 1). However, there are now 10 short transfers between each sender/receiver pair. Namely, between stations 0 and 3, 0 and 4, and 5 and 7, are flows 1–10, 11–20, and 21–30 respectively. In Fig. 8, we show the sum throughput of all active flows. As can be seen the RIPPLE scheme outperforms the other two approaches when supporting web traffic.

E. VoIP

To investigate RIPPLE's ability of supporting interactive traffic, we further test it with VoIP traffic. VoIP is sensitive to both losses and delay/jitter, and thus is more challenging than TCP traffic. The standard evaluation metric for VoIP is Mean Opinion Score (MoS) which ranges from 1–5, where 1, 2, 3, 4, 5 corresponds to that the perceived VoIP quality is *impossible*, *very annoying*, *annoying*, *fair* and *perfect* respectively. MoS is commonly estimated from an R-factor as: 1, if $R < 0$; 4.5, if $R > 100$; and $1 + 0.035 \times R + 7 10^{-6} R(R-60)(100-R)$, otherwise. The R-factor is obtained (as per [5]) from the

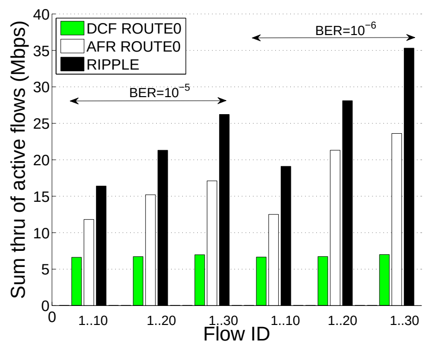


Fig. 8. Web traffic results in Mbps (sum throughput of all active flows) for the topology in Fig. 1. Parameters used are listed in Table I.

	BER=10^{-5}			BER=10^{-6}		
Flows	1..10	1..20	1..30	1..10	1..20	1..30
DCF ROUTE0	3.82	1.19	1.18	4.13	1.56	1.20
AFR ROUTE0	4.11	1.21	1.00	4.12	1.42	1.01
RIPPLE	4.11	2.49	1.75	4.14	2.82	2.09

TABLE III
MOS FOR VOIP TRAFFIC IN FIG. 1. THE PHYSICAL LAYER DATA AND BASIC RATES USED ARE BOTH 6MBPS, AND OTHER PARAMETERS LISTED IN TABLE I.

expression $94.2 - 0.024d - 0.11(d - 177.3)H(d - 177.3) - 11 - 40\log(1 + 10e)$, where d is the mouth-to-ear delay including coding/network/buffering delays, e is the total loss rate including losses in the network and those due to late arrivals, and $H(x) = 1$ if $x > 0$; 0 otherwise.

For simulating VoIP traffic, we model a 96kbs on-off traffic stream with on and off periods exponentially distributed with mean 1.5 seconds. Similarly to [5], we aim a 177 ms mouth-to-ear delay, and a 52 ms delay for the wireless part. That is, packets arrived but with a >52 ms delay in wireless part are considered as losses.

We use the same topology as for the web traffic, and in Table III we list the results, where it can be seen that the MoS achieved by the RIPPLE scheme is consistently higher than that with the DCF and AFR schemes.

F. Large Scale Topologies with Low Rates

We now consider a typical Wigle topology, shown in Fig. 9, obtained from the Wigle database which contains measurements of real AP locations (the topology used corresponds to the connected part of Fig 3. in [20]). The main network consists of 8 wireless stations, and we added two additional stations *S* and *R* in order to simulate the impact of hidden collisions. The hidden traffic used is a TCP flow from *S* to *R* sending 1×10^6 packets.

Figs. 10(a) 10(c) and 10(b) 10(d) present throughput measurements for TCP flows between eight randomly picked pairs of stations. Results are shown with and without hidden stations (i.e. with and without traffic between stations *S* and *R*). Due to the small diameter of the network topology, most of the flows traverse 1–3 hop(s). The results show that the RIPPLE scheme consistently outperforms predetermined routing with

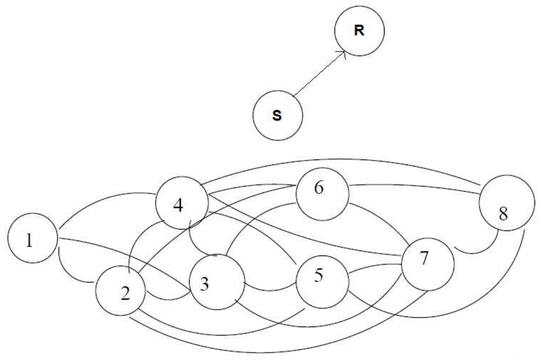


Fig. 9. A typical topology from the Wigle database, adapted from Fig. 3 of [20]

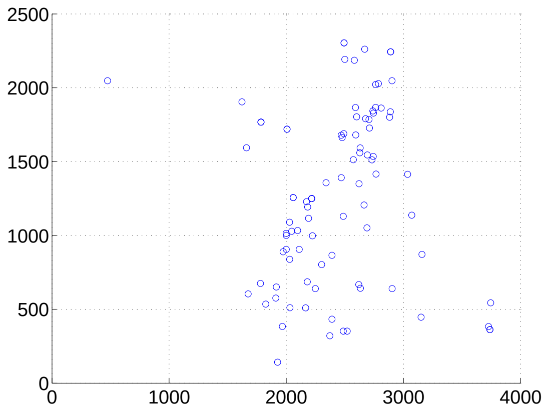


Fig. 11. The Roofnet topology. The unit for both x-axis and y-axis is meter.

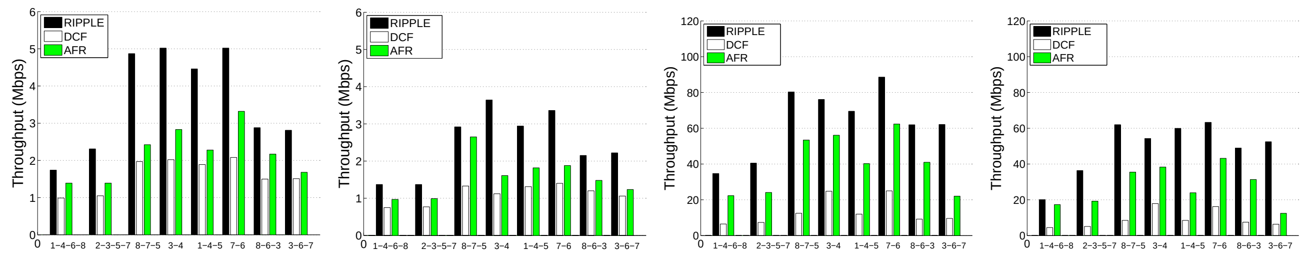
the DCF and AFR schemes. Improvements of up to 200% are observed (e.g.flow 8-7-5 in Fig. 10(a)).

Lastly we consider the topology (see Fig. 11) of the MIT Roofnet, derived from the GPS coordinates file at http://www.pdos.lcs.mit.edu/roofnet/roofnet-coords. This topology is relatively large so we focus on transmissions between stations that are 4 or 5 hops apart. After picking station pairs to use as senders and receivers, two more nearby stations are selected to act as the hidden terminals.

The results obtained are shown in Fig. 12. Again, the results show that the RIPPLE scheme consistently outperforms the other two schemes with up to 300% improvement observed (e.g., flow 5(1) in Fig. 12(a)).

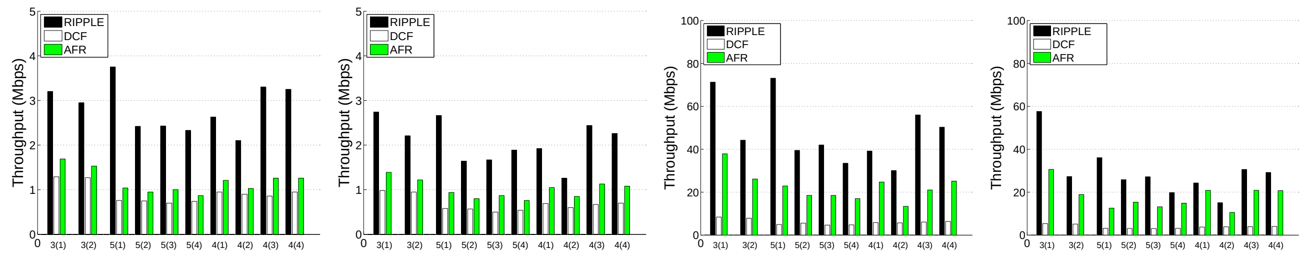
V. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a novel scheme called RIPPLE. In the RIPPLE scheme, an expedited multi-hop transmission opportunity mechanism ensures low signaling overhead and eliminates re-ordering, and a two-way packet aggregation technique further reduces overhead. We implement the RIPPLE and related schemes in NS-2 and compare their performance for long/short TCP transfers and VoIP over a wide range of network conditions, including varied wireless channel states, levels of regular/hidden collisions, and geographic locations of stations derived from measurement studies (i.e., the Wigle and



(a) 6Mbps (b) 6Mbps with hidden terminals (c) 216Mbps (d) 216Mbps with hidden terminals

Fig. 10. Throughput measurements for the topology in Fig. 9. The labels on the x-axis indicate the flow path concerned. For example, '1-4-6-8' means that the flow is from station 1 to 4 with stations 4 and 6 as forwarders.



(a) 6Mbps (b) 6Mbps with hidden terminals (c) 216Mbps (d) 216Mbps with hidden terminals

Fig. 12. Results for the Roofnet topology in Fig. 11. The labels on the x-axis indicate the number of hops and the number of each test. For example, '3(1)' means there are 3 hops between the source and the destination and this is the 1st 3-hop example, with '3(2)' meaning the 2nd 3-hop example.

Roofnet topologies), etc. Our results show that the RIPPLE scheme consistently delivers 100% – 300% performance gains over other approaches.

In future work, we plan to design a theoretical analysis for the RIPPLE scheme and will consider if it is possible to combine the advantages of opportunistic routing and rate adaptation.

References

[1] *Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications*, IEEE 802.11.

[2] *Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Amendment 4: Enhancements for Higher Throughput*, IEEE 802.11n.

[3] http://pdos.csail.mit.edu/roofnet

[4] V. Bahl, et. al., "Opportunistic Use of Client Repeaters to Improve Performance of WLANs," Conext 2008.

[5] A. Balasubramanian, R. Mahajan, A. Venkataraman, B. N. Levine, and J. Zahorjan, "Interactive WiFi Connectivity For Moving Vehicles," Sigcomm 2008.

[6] S. Biswas and R. Morris, "Opportunistic Routing in Multi-Hop Wireless Networks," Hotnets 2003.

[7] S. Biswas and R. Morris, "ExOR: Opportunistic MultiHop Routing for Wireless Networks," Sigcomm 2005.

[8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading Structure for Randomness in Wireless Opportunistic Routing," Sigcomm 2007.

[9] N. Celandroni, "Comparison of FEC types with regard to the efficiency of TCP connections over AWGN satellite channels," *IEEE Trans. on Wireless Commun.* vol. 5, no. 7, pp. 1735-1745, Jul. 2006.

[10] N. Celandroni, F. Davoli, E. Ferro, and A. Gotta, "Long-Lived TCP Connections Via Satellite: Cross-Layer Bandwidth Allocation, Pricing, and Adaptive Control," *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 1019-1030, Oct. 2006.

[11] Y.-C. Cheng, et. al., "Jigsaw: Solving the Puzzle of Enterprise 802.11 Analysis," Sigcomm 2006.

[12] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. "A high-throughput path metric for multi-hop wireless routing," MOBICOM 2003.

[13] Y. Ganjali, and A. Keshavarzian, "Load Balancing in Ad hoc Networks: Single-path Routing vs. Multi-path Routing", Infocom 2004.

[14] Y. Ganjali, and A. Keshavarzian, "Selection Diversity Forwarding in a Multihop Packet Radio Network with Fading Channel and Capture," ACM MCCR 2001.

[15] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level Network Coding for Wireless Mesh Networks," Sigcomm 2008.

[16] M. Kurth, A. Zubow, and J. P. Redlich, "Cooperative Opportunistic Routing using Transmit Diversity in Wireless Mesh Networks," Infocom 2008.

[17] T. Li, Q. Ni, D. Malone, D. Leith, T. Turletti, and Y. Xiao, "Aggregation with Fragment Retransmission for Very High-Speed WLANs," *IEEE/ACM Transactions on Networking*, Apr. 2009.

[18] T. Li, D. Leith, and L. Qiu, "Opportunistic Forwarding for Interactive Traffic in Multi-hop Wireless Networks," *Technical Report*, 2009, www.hamilton.ie/tianji_li/ripple.tr.pdf.

[19] Y. Lin, B. Li, and B. Liang, "CodeOR: Opportunistic routing in wireless mesh networks with segmented network coding," ICNP 2008.

[20] A. Mishra, V. Shrivastava, D. Agarwal, S. Banerjee, and S. Ganguly, "Distributed Channel Management in Uncoordinated Wireless Environments," Mobicom 2006.

[21] R. S. Prasad, C. Dovrolis, and M. Thottan, "Router Buffer Sizing Revisited: The Role of the Output/Input Capacity Ratio," CoNEXT 2007.

[22] R. Ramanathan, "Challenges: a radically new architecture for next generation mobile ad hoc networks," MobiCom 2005.

[23] D. Tang and M. Baker, "Analysis of A Local-Area Wireless Network," in *Proc. of ACM MobiCom*, Aug. 2000.

[24] Z. Zhao, S. Darbha, and A. L. N. Reddy, "A Method for Estimating the Proportion of Nonresponsive Traffic At a Router," *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 708–718, Aug. 2004.

[25] A. Zubow, M. Kurth, and J. P. Redlich, "Multi-Channel Opportunistic Routing," European Wireless Conference 2007.