Poster: 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

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.

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, which we refer to as preExOR to distinguish it from the the later work from the same authors. 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 mitigates the overhead caused by sequential one-hop ACKs by working in terms of batches and using end-to-end 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%) 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 ExOR.

Approaches using per-packet ACKs (i.e., preExOR and MCExOR) are not effective due to the high signaling overhead and also re-ordering issues. For example, in Fig. 1 we illustrate transmission timeline of two packets. The predetermined route used is called PRR for ease of explanation. 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.

Re-ordering can also happen in the preExOR and MCExOR schemes 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.

Forwarding interactive traffic opportunistically is thus challenging.

To tackle the challenge of supporting interactive traffic opportunistically we design a novel scheme called RIPPLE. 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.

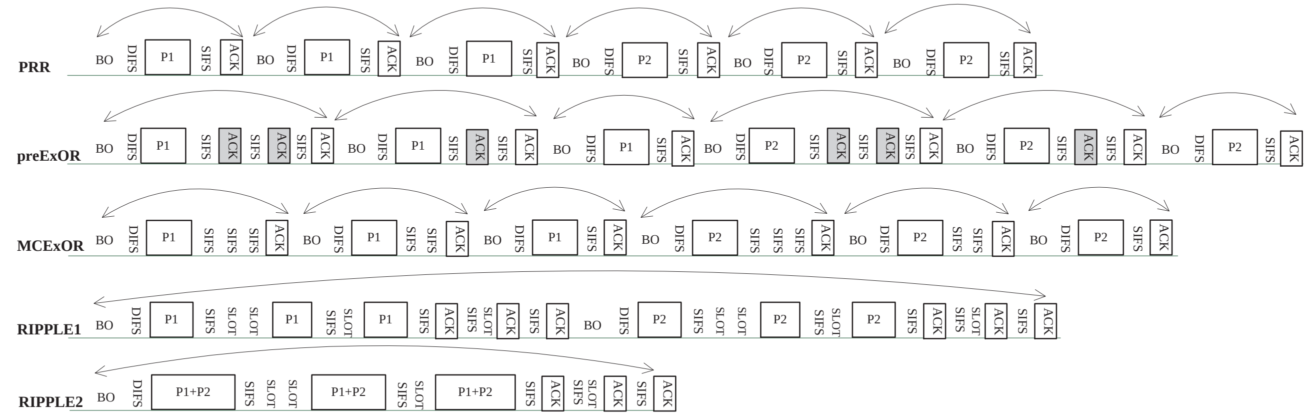

Fig. 1. Transmissions of two packets (P1 and P2) with PRR, preExOR, MCExOR and RIPPLE. BO is the abbreviation of backoff. In the preExOR scheme, shadowed ACKs indicate that the sender is waiting for an ACK which is not transmitted. Each arc line indicates one transmission opportunity.

1) Resolving Re-ordering: To solve the re-ordering issue, we do not let the forwarders cache any heard frames while still letting them help forward transmissions. 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[1] (mTXOP) and describe the details in the following steps.

- *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. 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. 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. Therefore, high priority forwarders can help relay whenever they overhear the transmissions, thus improving performance.
- *End-to-end 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 an end-to-end 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.

Using the mTXOP mechanism, the transmission timeline for packets *P1* and *P2* is shown in Fig. 1 (see RIPPLE1).

2) Mitigating Overhead: Although we can guarantee reordering free using the mTXOP mechanism, a similar to the preExOR and MCExOR schemes 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 to be aggregated in the (re)transmitted frame[2]. 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. 1) leads to approximately 50% overhead reduction in comparison to the non-aggregated version (i.e., RIPPLE1 in Fig. 1).
- Owing to the broadcast nature of the wireless medium, 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.

We implement the RIPPLE and related schemes 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.

---

[1] 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

[2] 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.