Multi-hop MAC Implementations for Affordable SDR Hardware

J. Colman O'Sullivan*, Paolo di Francesco†, Uchenna K. Anyanwu‡, Luiz A. DaSilva* and Allen B. MacKenzie‡

*CTVR / The Telecommunications Research Centre, University of Dublin, Trinity College, Ireland
†University of Bologna, Bologna, Italy
‡Virginia Tech, USA

Email: cosull13@tcd.ie, paolo.difrancesco@studio.unibo.it, uchevt@vt.edu, dasilval@tcd.ie, mackenab@vt.edu

***Abstract*—This paper presents the implementation and experimental evaluation of a MAC protocol designed to overcome some of the limitations of affordable and widely-deployed software defined radio hardware. We propose a modified Aloha-based MAC protocol with implicit acknowledgements to mitigate the impact of intra-flow collisions in multi-hop wireless communications. We experimentally observe a significant improvement in throughput and delay through simple modifications to the MAC protocol, tailoring it to the timing constraints of the USRP1.**

I. Introduction & Motivation

The evaluation of multi-hop networks of software defined radios (SDRs) and cognitive radios (CRs) has focused primarily on simulation. This has long has been the case for Mobile Ad-hoc Network (MANET) research, and its shortcomings are well documented [1]. Implementation and experimentation present both a steeper learning-curve and their own myriad difficulties, particularly with regard to replicability [2]. In our experience [3], though, they also bring to light new protocol issues that may not be apparent in simulations, whether due to assumptions and simplifications made in modeling or otherwise unforeseen effects of physical hardware. Simulation serves an important role, of course, allowing a scale of experimentation which is usually impractical for implementation, especially at the early stages of research. Research is best served by a combination of methods, with results from both serving to refine protocol and system design.

Thus far, implementation and experimentation with SDRs and CRs have largely focused on physical-layer communications, mostly ignoring higher layer issues. In large part, this may be because it has proved difficult to implement reasonable medium access control (MAC) on inexpensive SDR platforms. A key problem, identified by some of us in [4], is long (and sometimes variable) latencies in the transmit and receive chains and in reconfiguration from receive mode to transmit mode. This issue has been further explored in [5] with similar conclusions — implementing MAC protocols with acceptable performance on widely available, inexpensive SDR hardware platforms is challenging.

Clearly, better SDR hardware is needed and will be developed over time. In particular, we anticipate that the arrival of new hardware (such as the new USRP E-series [6]) that enables hybrid FPGA/GPP radio implementation may make the implementation of CSMA-based MAC protocols more tractable. In the meantime, though, there are dozens of inexpensive SDR hardware platforms already deployed in cognitive radio testbeds such as [7]. The goal of this paper is to explore and implement random-access MAC protocols appropriate for use in multi-hop networks of nodes whose bus latencies make CSMA-based protocols untenable.

II. MAC Protocols for Limited SDR Hardware Platforms

A. Platform and Protocol Choices

Extending wireless communications from single-hop to multi-hop is a significant challenge in SDR research. Although other hardware platforms would allow us to develop high performance MACs, our objective was to make use of widely available and affordable experimental equipment. In particular, we have selected the widely-used USRP1 from Ettus Research [6].

Common MAC layer protocols include carrier sense multiple access (CSMA), time-division multiple access (TDMA), slotted Aloha, and Aloha. When implemented with USRP1 hardware, host-to-device latency and jitter (introduced by the USB bus), as well as latency at the GPP will cause problems for MAC protocols with tight timing constraints. Scheduling a transmission at a precise point in time, such as for TDMA or slotted Aloha, for example, or rapid turnaround between reception and subsequent transmission, such as for CSMA, cannot be performed without hardware modification [5]. TDMA and slotted Aloha also require tight inter-node synchronization, which is also challenging on the USRP1 hardware platform.

As a result of its simplicity and lack of timing requirements, we have selected and implemented unslotted Aloha with stop-and-wait ARQ as the foundation of our MAC protocol.

This material is partly based upon work supported by Science Foundation Ireland under Grant No. 08/CE/I523 as part of CTVR / The Telecommunications Research Centre, and under the Stokes Professorship programme.

This research was supported by a Bradley Fellowship from Virginia Tech's Bradley Department of Electrical and Computer Engineering and made possible by an endowment from the Harry Lynde Bradley Foundation.

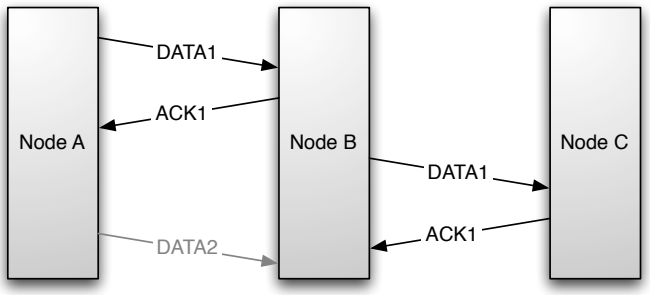

Fig. 1. Order of DATA and ACK packets for two-hop scenario. For the *Explicit ACK with delay* protocol, the delay corresponds to the position of **DATA2**. Given insufficient delay, it might be sent at the same time as **DATA1** is forwarded by Node B or as **ACK1** is sent by Node C, and hence cause interference.

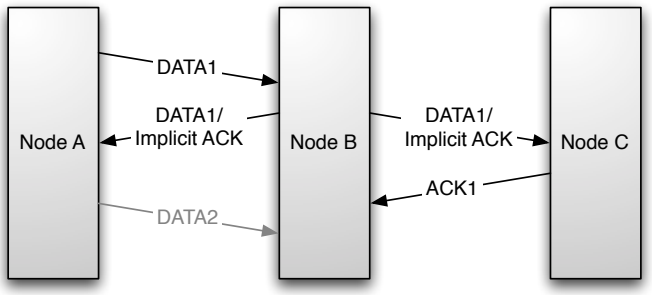

Fig. 2. Order of DATA and ACK packets for an *Implicit ACK* protocol. As in Fig. 1, delay corresponds to the position of DATA2.

In a multi-hop scenario with per link ARQ, however, we expect there to be a significant risk of intra-flow interference. Referring to Figure 1, it seems likely that if node A is sourcing more than one packet (as in any significant data transfer) for node C, then the second DATA packet sent by Node A will collide with Node B's attempt to forward the first DATA packet to node C. This is the primary problem that we address in this work.

Among the SDR software platforms freely available for research purposes, GNU Radio is undoubtedly the most popular [8]. However, in the implementation of MAC protocols, its stream-based data-flow model may not be the most intuitive or efficient choice. The Iris SDR platform uses a block-based data flow model and, as a result, is highly suited to processing packets [9]. Since packets are the natural processing unit in Iris, the radio design does not need to be artificially constrained as in GNU Radio. As with GNU Radio, Iris can be easily paired with Ettus USRP1 RF frontends, to allow for affordable experimentation over-the-air [6].

B. Proposed Solutions

We have first implemented a pure Aloha, stop-and-wait MAC protocol with a DATA-ACK handshake as shown in Fig. 1. Routing and forwarding is performed at the network layer, and the MAC layer requires no knowledge of the network's topology.

Naming the source, relay and destination nodes A, B and C respectively, it is easy to see that this configuration is vulnerable to intra-flow collisions, particularly under saturation conditions, in which node A has multiple packets to send. In particular, retransmission of the data packet by node B will likely occur at the same time that node A is sending the next data packet, resulting in a collision at node B. As it turns out, this problem is not quite as bad as we might expect *due to the long turnaround time at node A*. In particular, as we will see in the results in the next section, in many cases by the time node A has processed the received ACK packet and prepared to transmit the next DATA packet, node B has already completed its forwarding operation.

If the turnaround time at node A were *consistently* longer than the transmit latency for the forwarded packet at node B, then this would avoid much of the intra-flow interference. Experimentally, though, we find that retransmissions at the source node are still quite common. The long turnaround latency, though, inspires our first enhancement: After receiving an ACK, a node delays the transmission of the next DATA packet for a short period of time to allow the relay to complete packet forwarding, reducing the risk of a collision. We call this modified Aloha protocol *Explicit ACK with Delay*.

A further enhancement to the Aloha protocol for multihop use is the addition of Implicit ACKs to decrease collisions and increase throughput. In standard Aloha, as shown in 1, the relay node first sends an ACK and then retransmits the DATA packet. In the implicit ACK scheme, the relay node embeds the ACK in the forwarded DATA packet. When the source node deframes the packet, it finds the ACK and can then proceed to transmit the next DATA packet. The destination node, upon deframing the same packet, ignores the embedded ACK and simply processes the received packet as a DATA packet. The destination node still sends an explicit ACK, as it has no need to forward the DATA packet. Note that in this case the MAC protocol must be able to determine whether it is the final destination of the received packet in order to be sure that the ACK is embedded in the proper outgoing packet. We refer to this scheme as the *Implicit ACK* protocol. A protocol diagram is shown in Fig. 2 and a state diagram is shown in Fig. 3.

There remains a benefit to imposing a delay on transmissions at the source node, as the destination node still sends an explicit ACK to inform the relay of its successful receipt of the forwarded DATA and implicit ACK packet. As the relay is the intended target of both this ACK and the source's DATA packets, interference may occur if there is insufficient delay.

III. Implementation & Results

1) Explicit ACK Protocol with Delay: Our experimental setup was of a two-hop network, consisting of 3 Iris nodes each using a USRP1 RF front-end. Each experimental run involved successfully transmitting 500 data packets from the source node to the destination node via a relay node. The parameters used for the radio setup can be seen in Table III-1.

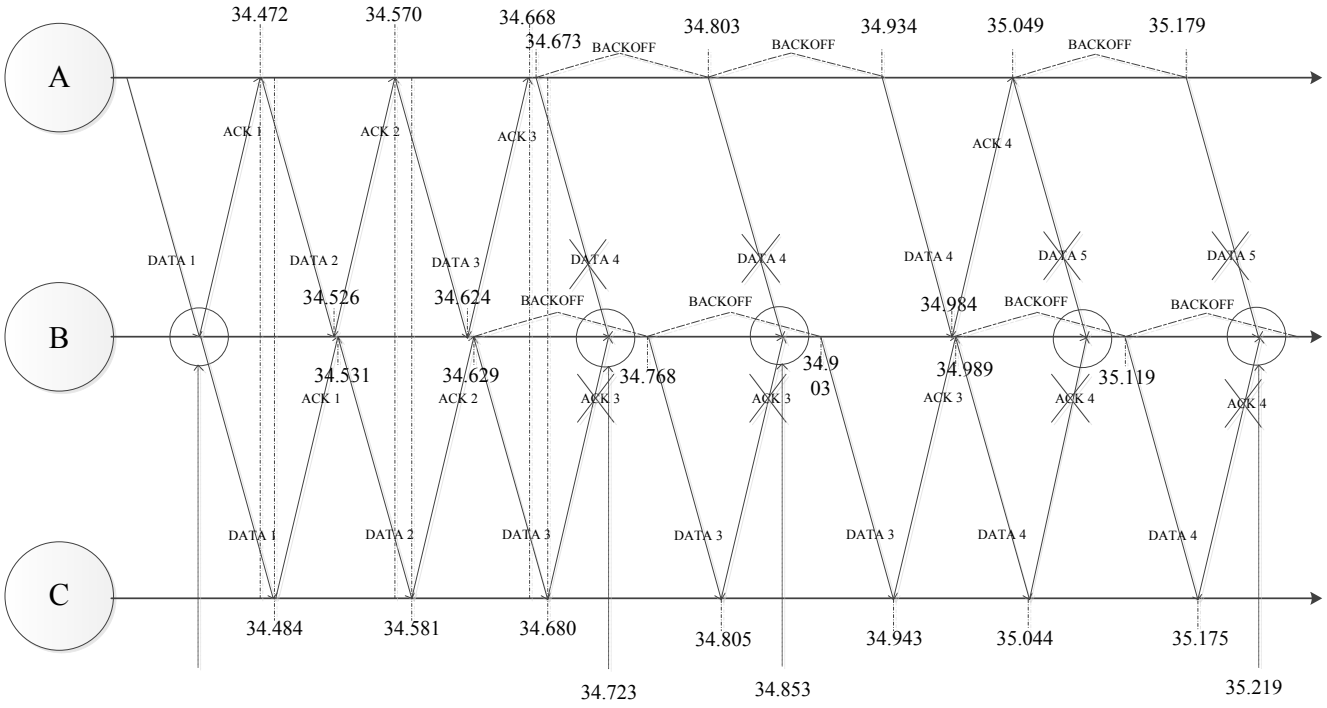

Fig. 4. Timeline of events for a typical experiment. Nodes A, B & C are the source, relay and destination respectively.

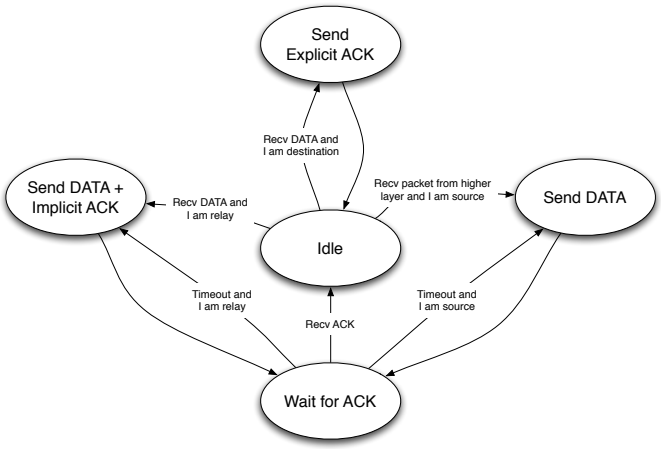

Fig. 3. State diagram of the proposed *Implicit ACK protocol*. It modifies basic Aloha operation to add an implicit ACK header to forwarded DATA packets at the relay node, while still using conventional explicit ACKs at the destination node.

Packet size at MAC Layer	600 B
Center Frequency	2.39 GHz
Bandwidth	1 Mhz
Modulation Scheme	OFDM
Active data subcarriers	192 of 256
Subcarrier Modulation	BPSK
ARQ timeout	50 ms

TABLE I
TABLE OF RF PARAMETERS USED

Fig. 4 is a partial timeline of events from one of the many experiments run. It shows, as expected, that the variable latency associated with the Tx chain, Rx chain, and turnaround time [10] results in intra-flow collisions at the relay for some, but not all, packets.

In order to test the modified *Explicit ACK with Delay*, we performed 50 experimental trials for each value of delay, increasing the delay from 0ms to 40ms in increments of 10ms. The results can be seen in Fig. 5. Each bar represents the average number of retransmissions for the source node and relay node respectively. The best compromise between forced delay and number of retransmission for this configuration is given by 20ms.

2) Implicit ACK Protocol: Performing a similar set of 50 trials for each delay value, we experimentally we found a 10 ms delay on the first node to be the best-performing delay for the two-hop environment using a MAC modified to employ Implicit ACKs. (See Fig. 6.) Specifically, we found that there is almost 25 percent improvement in end-to-end throughput, as compared to the throughput observed with Explicit ACK with Delay alone.

3) Effect of Buffer Size on Transmission/Reception Times: When performing experiments using a USRP1 SDR, it is important to be aware of the presence of a receive buffer, of a variable size of up to 32KB and a fixed 8KB on the host machine and the USRP1, respectively. These buffers can influence the processing time and the end-to-end delay for the whole system, so the choice of how to fill these transmit and receive buffers is of great relevance. In order to understand how this buffering affects the system, we measured the Round

Trip Time (RTT), here defined as the time elapsed from sending a DATA packet (at the MAC layer) to receiving its ACK. This includes the time required to frame the DATA packet, modulate and scale it in Iris, send it over the USB, pass through the USRP buffers and send it over the air; as well as the time required to perform the reverse at the receiver, and to send and demodulate the corresponding ACK.

We used an Explicit ACK strategy (with no delay) at the MAC layer, and an OFDM modulation made up of BPSK modulated subcarriers at the PHY layer.

A DATA packet at the last component of the Iris transmit chain comprises 15232 samples, including OFDM-preamble and necessary padding bits. An ACK packet at the last component of the TX chain comprises 2176 samples. Initializing the USRP1's Rx buffer size at 16384 samples for one experiment, and at 8192 samples for another, we observed the behavior seen in Fig. 7.

It may be expected that the use of a larger Rx buffer would be beneficial if its size is greater than the largest possible number of useful samples (15232, as stated above), as this could mean that the demodulator would need to be called only once. The reality is different, though. A single call to the demodulator for a frame is unlikely, even with the larger buffer, as the entire frame will be contained within the buffer only when the OFDM preamble is located near the beginning of the buffer. In the majority of cases the preamble is not at the beginning of the buffer, and so demodulation of the OFDM frame is split over two (or more) calls. Should an OFDM frame spread over multiple calls to the demodulator, there is a good chance the frame will be found in its entirety before all the samples passed to the demodulator for the final call have been processed. The demodulator will however continue to process these remaining samples, and not pass the demodulated data to the Iris modules above until it has completed. This behavior avoids a backlog of samples in the demodulator, but causes a delay when the buffer size is large relative to the average number of samples subsequent to the end of a frame. As a result, a smaller buffer size leads to a faster turnaround time. The use of a smaller Rx buffer results in a RTT decrease of 30-40 percent and a corresponding improvement in end-to-end throughput.

IV. Conclusions & Future work

In this paper we discuss the implementation of and experimentation with MAC protocols for affordable SDR testbed hardware. We introduced a simple Aloha-based MAC protocol and discussed its implementation on the Iris software radio platform using USRP1 hardware. Extending this MAC protocol to a multi-hop scenario, we recognized potential problems with intra-flow interference. We modified the Aloha-based protocol to incorporate a forced delay, improving performance by reducing intra-flow contention. We further modified the protocol to employ an implicit ACK. Both modifications were experimentally verified to improve throughput and end-to-end delay, demonstrating that in spite of the difficulties associated

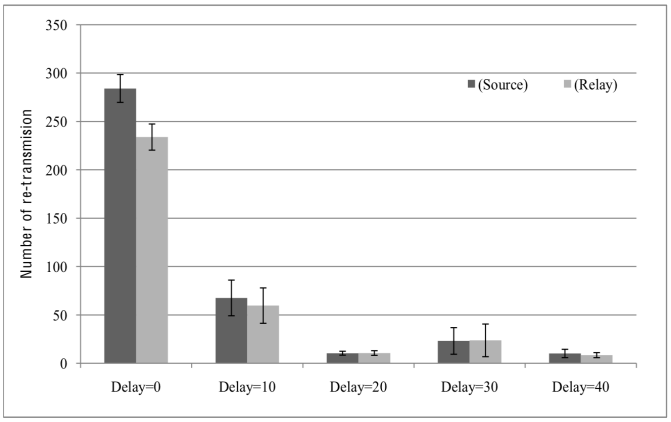

Fig. 5. Number of retransmissions versus delay for the Explicit ACK with Delay protocol. Bars represent the 95% confidence interval.

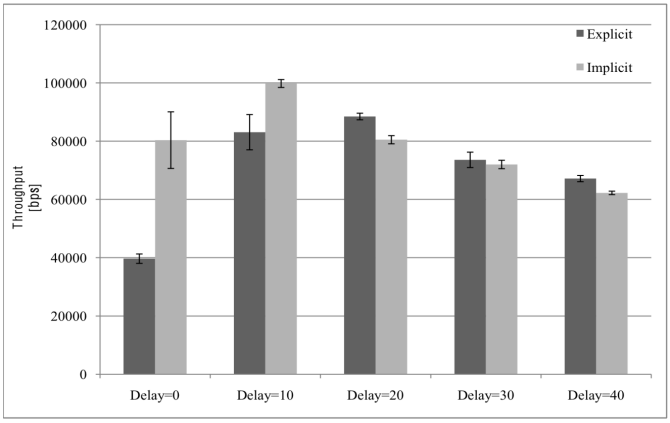

Fig. 6. Throughput for variations in delay. Bars represent the 95% confidence interval.

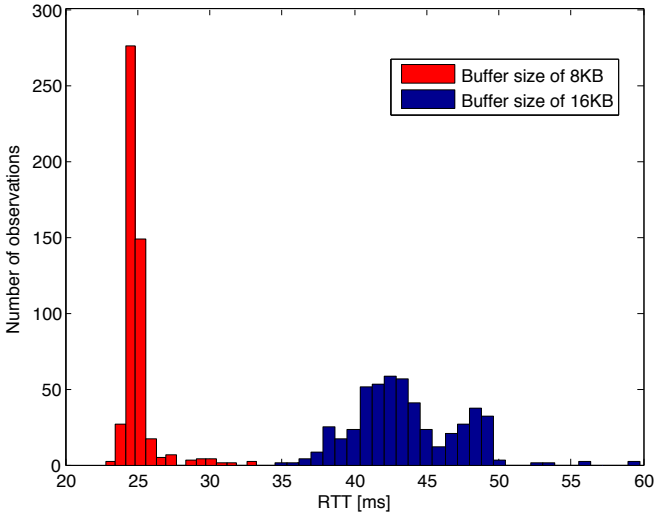

Fig. 7. Round-trip-time for variations in delay

with using affordable SDR hardware, MAC layer strategies can be tailored to these limitations.

Future challenges include an expansion in the number of hops and traffic flows considered in experiments and the development and validation of simulation models to allow even larger experiments. Further improvements will also include distributed control of MAC features to improve neighborhood or network-wide performance, and the incorporation of hardware allowing scheduled transmission-time (such as the USRP N210 and E100).

REFERENCES

[1] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET Simulation Studies: The Incredibles," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 4, pp. 50–61, 10 2005.

[2] K. Mandke, R. C. Daniels, S.-H. Choi, S. M. Nettles, and R. W. Heath, "Physical concerns for cross-layer prototyping and wireless network experimentation," in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, ser. WinTECH '07. New York, NY, USA: ACM, September 2007, pp. 11–18. [Online]. Available: http://doi.acm.org/10.1145/1287767.1287771

[3] M. S. Thompson, A. E. Hilal, A. S. Abdallah, L. A. DaSilva, and A. B. MacKenzie, "The MANIAC challenge: Exploring MANETs through competition," in *Proc. of the International Workshop on Wireless Networks: Communication, Cooperation, and Competition (WNC3)*, 2010.

[4] L. A. DaSilva, A. B. MacKenzie, C. da Silva, and R. W. Thomas, "Requirements of an Open Platform for Cognitive Networks Experiments," *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, pp. 1–8, October 2008.

[5] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste, "Enabling MAC protocol implementations on software-defined radios," in *NSDI'09: Proceedings of the 6th USENIX symposium on Networked systems design and implementation*. Berkeley, CA, USA: USENIX Association, 04 2009, pp. 91–105.

[6] "Ettus Research LLC." [Online]. Available: http://www.ettus.com/

[7] T. R. Newman, S. M. S. Hasan, D. DePoy, T. Bose, and J. H. Reed, "Designing and deploying a building-wide cognitive radio network testbed," *IEEE Communications Magazine*, vol. 48, no. 9, pp. 106–112, September 2010.

[8] Free Software Foundation, Inc., "GNU Radio — The GNU Software Radio." [Online]. Available: http://www.gnu.org/software/gnuradio/

[9] P. D. Sutton, J. Lötze, H. Lahlou, S. A. Fahmy, K. E. Nolan, B. Özgül, T. W. Rondeau, J. Noguera, and L. E. Doyle, "Iris: An Architecture for Cognitive Radio Networking Testbeds," *Communications Magazine, IEEE*, vol. 48, no. 9, pp. 114–122, 09 2010.

[10] T. Schmid, O. Sekkat, and M. B. Srivastava, "An experimental study of network performance impact of increased latency in software defined radios," in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, ser. WinTECH '07. New York, NY, USA: ACM, September 2007, pp. 59–66. [Online]. Available: http://doi.acm.org/10.1145/1287767.1287779