Piggybacked-Ack-aided Concurrent Transmissions in Wireless Networks

Naveen Santhapuri Junling Wang Zifei Zhong (students) Srihari Nelakuditi (faculty)
Department of Computer Science & Engineering
University of South Carolina
Columbia, SC, 29208
{santhapu,wang257,zhongz,srihari}@cse.sc.edu

1. INTRODUCTION AND MOTIVATION

It is well-known that MAC protocols for multihop wireless networks face the problems of *hidden* and *exposed* nodes due to the limited transmission range of wireless nodes. The Distributed Coordination Function (DCF) mechanism of the IEEE 802.11 standard [2] addresses the issue of hidden node problem by employing a four-way handshake of RTS-CTS-DATA-ACK frames. In this exchange, the transmitter emits RTS and DATA frames and the receiver responds with CTS and ACK frames. During this whole period, all nodes that are in the range of either the transmitter or the receiver has to remain silent so as to not interfere with the reception of DATA frame at the receiver and CTS/ACK frames at the transmitter. While this approach alleviates the hidden node problem, it does not address the exposed node problem. During a transmission $A \rightarrow B$, it does not permit a feasible simultaneous transmission $C \rightarrow D$ by a node C which is in the range of A, even if B and D are outside the range of C and A respectively. For example, in Fig. 1, when 2→1 transmission is on-going, no other transmission is allowed even though three transmissions 2→1, 5→6, and 3→4 can happen concurrently.

There have been some proposals [1,4] for enabling concurrent transmissions and thus increasing the overall throughput of a wireless network. We briefly mention two such proposals that are quite relevant to our work in the following. The MACA-P scheme proposed in [1] uses an extended control phase to synchronize the DATA and ACK phases of all concurrent transmissions to avoid the problem of DATA of a transmission and ACK of another transmission interfering with each other. To achieve this alignment, RTS and CTS frames carry the times of the scheduled DATA and ACK phases. Another scheme proposed in [4] tries to squeeze in a secondary transmission (without RTS and CTS) concurrently with the primary transmission (with RTS and CTS). This scheme requires that the size of DATA of secondary transmission is smaller than that of primary transmission, and also that the secondary transmission is deferred such that both primary and secondary transmissions finish exactly at the same time, in order to avoid any interference between DATA and ACK frames.

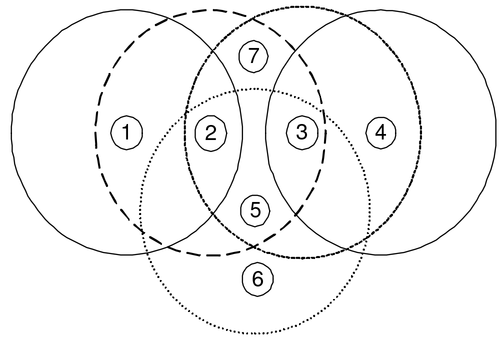

Figure 1: An example wireless network

The fundamental limitation of IEEE 802.11 DCF, that prevents concurrent transmissions, is that transmitter and receiver switch their roles during RTS-CTS-DATA-ACK exchange [1]. The transmitter is the recipient for CTS/ACK frames whereas the receiver is the recipient for RTS/DATA frames. The above approaches for concurrent transmissions address this limitation by synchronizing the actions of all transmitter-receiver pairs such that role reversals happen in lock-step. In this work, we investigate an alternate approach that explores the possibility of obviating the need for role reversals, thus enabling concurrent transmissions without such synchronization among transmitter-receiver pairs.

2. OUR APPROACH

We propose to address the exposed node problem by eliminating ACK frames and instead *piggybacking* the ACK information on other frames. Our piggybacked ACK is specifically desgined to replace the functionality of the explicit ACK in 802.11. In addition, we suggest making the reception of CTS optional for transmitting the corresponding DATA frame, based on the states of neighbors. With these changes, a transmitter is not required to be a recipient during the transmission, which in turn enables concurrent transmissions. We refer to our approach as *Piggybacked-ACK-aided Concurrent Transmissions* (PACT) [3]. In the following we provide details on piggybacking of ACKs and how it facilitates concurrent transmissions.

ICNP'05, November 6-9, 2005, Boston, Massachusetts, USA.
.

2.1 Piggybacking Acknowledgments

A key idea of our approach is to replace explicit ACK frames in 802.11 with piggybacked ACKs in other frames. Several opportunities for piggybacking ACKs exist in a multihop wireless network. It is likely that the receiving node is not the final destination and so forwards the packet soon after. The sender may have more packets to the same receiver and elicit a CTS from it. The receiver is likely to have data for the sender or other neighbors. A TCP-like sliding window is maintained by the sender since the packets are not ACKed immediately. A packet is retransmitted if a duplicate ACK is received or if there is no ACK within a timeout period. The proposed piggybacked-ACK approach makes ACK delivery robust by carrying ACKs to multiple senders in every frame.

The piggybacked-ACK mechanism can be implemented by including a PACK field in the header of each frame emitted by a node. The PACK field carries ACKs for several neighbors. An ACK for a neighbor is represented by <`nibid`,`seqno`> where `nibid` identifies the neighbor uniquely using a certain number of lower order bits of its MAC address, and `seqno` is the sequence number of the most recently received packet in-order from that neighbor. It is assumed that a node is aware of all its neighbors and can determine if a given number of bits are sufficient to uniquely identify a neighbor. An example encoding of PACK field using 5 bytes can be as follows. The `nibid` may be represented using 5, 8, 14 or 32 bits to ensure uniqueness among the neighbors whereas seqno for each ACK may be coded using 4 bits. Effectively, the size of each ACK would be 9, 12, 18, or 36 bits which means that PACK may carry ACKs for 4, 3, 2, or 1 neighbors depending on the number of bits needed to represent `nibid`. The number of ACKs can be coded using 2 bits leaving 2 bits for future use.

2.2 Piggybacked-ACK-aided Concurrent Transmissions

We now describe how piggybacked-ACKs facilitate concurrent transmissions. Consider a transmission 2→1 in Fig. 1. Suppose node 3 has a packet to be sent to 4. Under 802.11 DCF, when 3 hears 2→1 RTS, it defers its own transmission until the corresponding CTS-DATA-ACK exchange is complete. On the other hand, under PACT, since there is no explicit ACK frame, 3 can begin its transmission to 4 after a time equivalent to CTS as the RTS and DATA frames corresponding to 3→4 do not interfere with DATA of 2→1. Here it is assumed that node 3 can determine whether DATA frames of 3→4 and 2→1 interfere with each other. In other words, PACT requires that a node has 2-hop knowledge, i.e., it is aware of the neighbors of each of its neighbors, which is quite feasible, particularly in case of fixed wireless networks. In addition, each node has to maintain the state, i.e., a separate contention window and separate NAV for each neighbor, which we believe is worth paying for the increased throughput.

The number of concurrent transmissions under PACT can be controlled by making CTS optional as follows.

2.2.1 RTS-CTS-DATA

This is a conservative scheme which requires that a node receive a CTS before it sends a DATA frame. However, when a node is not in a position to receive CTS, due to an ongoing transmission in its range, it proceeds with DATA frame after the CTS timeout.

2.2.2 RTS-[CTS]-DATA

Under this aggressive scheme, a node always proceeds with DATA frame regardless of the reception of the CTS. The purpose of CTS here is to reserve the channel around the receiver and also to piggyback ACKs whenever possible.

The above two variants have their merits and demerits. They behave similarly in some cases and differ in others. For example, both permit three concurrent transmissions 2→1, 3→4 and 5→6 or two concurrent transmissions 2→7 and 5→6. On the other hand, with RTS-[CTS]-DATA, 3 concurrent transmissions 1→2, 4→3 and 6→5 are possible whereas only one such transmission is feasible with RTS-CTS-DATA.

2.2.3 Discussion

The proposed scheme tries to mitigate the exposed node problem, but at the same time runs the risk of aggravating the hidden node problem. In the scenario shown in Fig. 1, if 6→5 is ongoing and if 2 receives a RTS from 1, no CTS will be sent by 2. Under the aggressive scheme 1→2 is permitted. If 7 wants to transmit a packet to 3, there will be a collision at 2 since 7 is not aware of 1→2. However, the conservative scheme avoids such scenarios to a large extent (1→2 won't be permitted because 2 didn't send a CTS to avoid interrupting the reception of 5 and 1 requires a CTS). Additionally, overhead is incurred for piggybacked cumulative ACKs, per-neighbor sliding windows for acks and per-neighbor contention windows. The throughput gains of this scheme are expected to offset the additional overheads and increased instances of hidden node problems. We need to verify the usefulness of these tradeoffs by conducting a thorough investigation of these schemes under various scenarios.

3. ON-GOING AND FUTURE WORK

We implemented the piggybacked ACK mechanism by modifying the NS2 simulator and verified that its performance is similar to that of 802.11. The additional overhead for piggybacking the ACKs in the RTS/CTS/DATA frames is compensated by the lack of ACK frame. A preliminary evaluation of scenarios with single hop UDP flows yeilded a relatively high throughput compared to 802.11 due to the concurrent transmissions. We are now in the process of evaluating the effectiveness of conservative and aggressive versions of PACT. We intend to compare the performance of 802.11 MAC and PACT in terms of overall throughput and fairness under varying node densities and traffic characteristics.

4. REFERENCES

[1] A. Acharya, A. Misra, and S. Bansal. MACA-P: A MAC protocol to improve parallelism in multi-hop wireless networks. In *PERCOM*, 2003.

[2] IEEE Computer Society. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1997.

[3] Piggybacked-Ack-aided Concurrent Transmissions. http://arena.cse.sc.edu/pact.html.

[4] D. Shukla, L. Chandran-Wadia, and S. Iyer. Mitigating the exposed node problem in IEEE 802.11 ad hoc networks. 2003.