International Journal of Research in Computer Engineering and Electronics. 1 ISSN 2319-376X
VOl : 2 ISSUE :1 (Feb 2013)

Design Aspects of Multi-hop Wireless Networks using Frame Aggregation

Bharat Bhushan Naib[1], Dr. V.R. Singh[2]

Abstract— As data rates supported by the physical layer increase, PHY and especially MAC overheads increasingly dominate the throughput achievable by wireless networks. A promising approach for reducing these overheads is to aggregate a number of frames together into a single transmission. The 802.11n standard uses such an approach for unicast frames. We present the design of a system that can aggregate both unicast and broadcast frames. Further, the system can classify TCP ACK segments so that they can be aggregated with TCP data flowing in the opposite direction. A novel aspect of our work is that we implement and validate our design not through simulation, but rather using our wireless node prototype, Hydra, which supports a high performance PHY based on 802.11n. Design aspects of multi-hop wireless networks using frame aggregation are given in this paper.

Index Terms— Transmission control protocol (TCP), frame aggregation, 802.11n, wireless multi-hop networks.

1 Introduction

As the demand for high data rates increases, wireless networking systems are deploying broadband communication high-throughput technologies such as orthogonal frequency division multiplexing (OFDM) and multiinput multi-output (MIMO).

These technologies allow the data portions of frames to be transmitted at high data rates, which decrease the time, spent transmitting data, but does not generally decrease the time spent on a variety of overheads. These include the time spent waiting to gain access to the transmission floor, exchanging control frames required by the MAC protocol and physical layer (PHY) headers. The result is that these overheads begin to dominate performance even when the PHY is capable of high data rates. In general, this problem becomes more severe as rate increases because the time to transmit the data decreases, but the transmit time for most of the overheads does not.

Similarly, the effect of these overheads is more dominant for short frames, such as those typically used for control, than for longer ones. [1-21]

One approach to reducing these overheads and thus achieving the potential performance gains offered by modern PHYs is to group (or aggregate) several frames together into one transmission. This has two benefits : one, it reduces the total number of transmissions, resulting in less time waiting for the floor and transmitting control frames, and two, it reduces header overhead by allowing several frames to share headers. As an example of this approach, the IEEE 802.11n standard includes several frame aggregation schemes to support high data rates as part of its high throughput MAC design [8].

Most frame aggregation schemes require that frames that are aggregated all be destined to the same receiver. This approach neglects the fact that transmissions are broadcast and a single transmission will potentially be received by many receivers. A simple way of taking advantage of this is to aggregate broadcast frames along with a group of unicast frames all destined for one receiver. Because broadcast frames do not require acknowledgement, this can be done while still having a single ACK for the unicast frames.

We expect TCP traffic to be important for wireless networks and it can also benefit from frame aggregation. The key observation is that TCP ACKs are small packets that flow in the opposite direction from the (typically) larger TCP data packets. Since TCP ACKs are cumulative and thus carry redundant information, they have lower reliability requirements than the data packets. We take advantage of this by treating TCP ACKs as if they were broadcast frames and do not require link-level acknowledgements.

This allows the ACKs to be aggregated with TCP data traveling in the opposite direction, significantly reducing the cost of the TCP ACKs. By allowing the MAC to investigate TCP headers, our design breaks layering abstractions and thus is a cross-layer algorithm.

Design aspects of three aggregation techniques: unicast aggregation, broadcast aggregation, and TCP ACK aggregation. Although our design is a general modification of 802.11, our implementation is specifically for our Hydra wireless node prototype [6], which supports an 802.11n-based PHY, with OFDM and MIMO. Thus our performance evaluatoin is based on a real operational wireless network, rather than simulation. In this paper, design aspects of Hydra are given.

- *[1] Bharat Bhushan Naib is Research Scholar, Mewar University, Rajasthan.India.*
- *[2] Prof. (Dr.) V.R. Singh, Director, PDMCE, Bahadurgarh, Haryana .India*

2 Related Work

Despite the wide variety of physical layer approaches to increasing wireless network throughput, MAC overhead ultimately limits maximum achievable throughput [21]. Further, these limits have more impact for frames with small data payloads and the impact increases as the rate used for data increases.

Frame aggregation can help to address both the problem of short frames and of overhead becoming dominant at high data rates. Here we describe previous efforts to improve throughput using frame aggregation.

To improve throughput, the IEEE 802.11n [8] high throughput standard adopts two approaches to frame aggregation: the aggregated MAC service data unit (A-MSDU), and the aggregated MAC protocol data unit (A-MPDU) [1]. A-MSDU aggregates packets from the upper layer and adds a single MAC header and check-sum. This scheme is effective when the MAC aggregates many small user packets such as TCP ACKs or other control oriented data. A-MPDU concatenates normal 802.11 MAC frames each having its own MAC header and checksum. Each of these subframes is separated by a MAC delimiter, which includes a length, checksum, and delimiter signature. The MAC delimiter allows a receiver to robustly separate each subframe, even in the case where some errors occur in the individual subframe.

This approach has more overhead than the first, but supports a block ACK scheme. Block ACKs allow each subframe to be acknowledged separately, thus allowing retransmission of only the subframes in error. This approach will have an advantage with high error rates. Kim et al. [12] evaluate the throughput of an early variant of 802.11n frame aggregation as a function of payload size and physical data rate.

The 802.11n MAC also specifies a bi-directional data transfer method that can reduce floor acquisition overhead [1]. It is particularly useful for reducing the overhead of a bi-directional stream of TCP data and ACKs. When a node transmits a frame, instead of relinquishing the floor when the transmission completes, the node can grant the receiver permission for a reverse direction transmission destined for the original transmitter. This approach allows both TCP data and ACKs to be transmitted in turn. This saves a floor acquisition time and the time to exchange a request-to-send (RTS) and clear-to-send (CTS) if they are being used. However, this method does not reduce the MAC and PHY header overheads or the cost of link-level ACKs.

Skordoulis et al. [18] proposed a two-level frame aggregation scheme that mixes 802.11n's two aggregation methods. In the first stage, the MAC aggregates user packets from the upper layer into an A-MSDU with a MAC header and checksum. Then, a series of these A-MSDUs are concatenated into an A-MPDU with each A-MSDU separated by a MAC delimiter. This scheme increases the maximum aggregation size compared to using A-MSDUs and reduces MAC header overheads compared to using A-MPDUs. It allows the block ACK scheme to be applied to the A-MSDUs.

Kim et al. [11] proposed a multi-layer scheme that provides aggregation at both the MAC and PHY layers. The MAC aggregates multiple MAC frames into an A-MPDU, and then the PHY aggregates a series of A-MPDUs into a single physical frame. Within the physical frame, an additional physical delimiter precedes each of the A-MPDUs. The physical delimiter contains modulation and coding scheme information for each A-MPDU, and thus allows each A-MPDU to be transmitted at a different rate. Unlike the other existing approaches, this scheme also supports multi-destination aggregation because each A-MPDU can be addressed to a different destination. To facilitate this, the protocol employs a polling scheme so that frames sent to different destinations can be acknowledged.

Sadeghi et al. [16] proposed the opportunistic autorate (OAR) method, which uses frame aggregation to take advantage of favorable channel conditions. When the underlaying rate adaptation algorithm shows that a frame can be sent at higher than base-rate, the MAC attempts to aggregate frames so that the time spent sending the frame at the higher rate equals the time that would be the same as the time to send a single frame at base-rate. This preserves the basic fairness capabilities of the 802.11 MAC while taking advantage of higher rates and the overhead reduction of frame aggregation.

There have been cross-layer approaches to improve TCP performance by piggybacking small TCP ACKs with link-level frames [14, 19, 20, 17]. Parsa et al. [14] proposed the transport unaware link improvement protocol (TULIP) that provides a piggyback method which transfers a TCP ACK with a link-level frame. Tourrilhes [19] proposed PiggyData of which idea is to transmit PiggyData ACK, a link-level acknowledgement with a flag indicating status of transmit queue, as a response of TCP data. Setting the flag to one means that a TCP ACK is ollowed after SIFS interval. Xiao [20] suggested a piggyback mechanism which allows the MAC to piggyback a link-level acknowledgement with a TCP ACK in a single frame immediately after the node receives TCP data. Our scheme differs from these approaches in that these do not reduce the MAC and PHY header overheads or the cost of link-level ACKs.

Scalia et al. [17] suggested PiggyCode which allows the MAC to combine TCP data with TCP ACK by using network coding technology. This approach can reduce the MAC and PHY header overheads for TCP ACKs, similar to our approach. However, the PiggyCode requires additional header to encode and decode packets, and it allows only packets of different types to be coded together, which restricts transmitting a single TCP data and TCP ACK at a time.

3 Design Aspects

Our design incrementally extends the familiar IEEE 802.11 distributed coordination function (DCF) MAC protocol [7] to support frame aggregation. This addition requires modest changes to the PHY frame format, demonstrating one of the cross-layer aspects of our design.

3.1 Unicast Aggregation

The 802.11 standard includes significant overheads when transmitting a single frame. These overheads include gaining access to the floor, PHY and MAC header overheads, and control frame overheads for RTS, CTS, and ACK frames. Unicast aggregation reduces the impact of this overhead by combining frames being transmitted to the same destination, similar to what the IEEE 802.11 standard describes.

In addition to reducing header overheads, aggregating frames allows us to reduce the total number of transmissions, and thus amortizes much of the perframe overhead over several frames..As demonstrated in this document, the numbering for sections upper case Arabic numerals, then upper case Arabic numerals, separated by periods. Initial paragraphs after the section title are not indented. Only the initial, introductory paragraph has a drop cap.

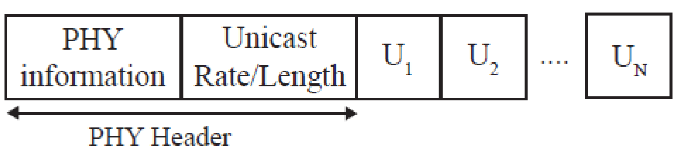

Figure 1: Unicast Aggregation Format

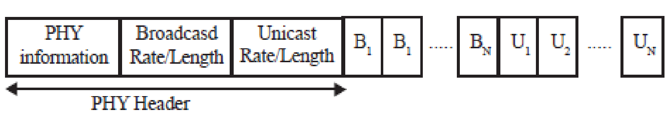

Figure 2: Broadcast Aggregation Format

Figure 1 is the format for supporting unicast aggregation, where UN denotes the N-th unicast subframe. The aggregated frame consists of some PHY-oriented information, such as training sequences, the rate and length fields for the frame and then a series of unicast sub-frames, all bound for the same destination. Because all the unicast subframes are destined for the same node, a single ACK can be used to acknowledge all the sub-frames. Here, no changes are needed to the PHY.

3.2 Broadcast Aggregation

Broadcast frames are likely to be important in a multihop wireless network, especially for control protocols. For example, the dynamic source routing and ad-hoc on-demand distance vector routing protocols use broadcast frames for route discovery and maintenance [9, 15]. The broadcast nature of radio frequency transmissions means broadcast frames can not only be aggregated with each other, but also with unicast frames. This promises to significantly lower the impact of flooding based control protocols on data transport.

Figure 2 shows how our broadcast aggregation format extends the basic unicast format, where BN stands for the N-th broadcast subframe. Our design modifies the PHY header to add a rate and length field for the broadcast subframes. The rate information enables us to support different data rates for broadcast and unicast subframes. The length information allows our protocol to prepend a variable number of broadcast subframes to a variable number of unicast subframes, all within the same physical frame. Our design requires modifying the PHY header to include rate and length information to allow the receiving PHY to decode incoming streams. The broadcast subframes are not acknowledged and thus a single link-level ACK is still sufficient. In addition, depending on queue status, the broadcast aggregation scheme allows the MAC to aggregate broadcast or unicast frames only.

3.3 Treating TCP ACKs as Broadcasts

Many of the Internet's most important applications use TCP as the transport protocol. Thus, breaking layer abstractions and optimizing for TCP becomes important. Furthermore, TCP represents a general class of protocols that support reliable transmission.

TCP relies upon a bidirectional traffic flow of TCP data and TCP ACKs. Because the ACKs are small, the impact of the fixed overhead becomes even more significant, making them especially good candidates for aggregation.

TCP employs a cumulative acknowledgement mechanism. In this scheme, receipt of an ACK for packet Pi, where i is the sequence number of the packet, implies acknowledgement of all previous packets Pj, for j i. Because of this redundancy, ACKs have less need for reliable transmission than data. Indeed, some TCP implementations intentionally drop some fraction of the ACKs to reduce protocol overhead [2].

This suggests that TCP ACKs could be transmitted without link-level acknowledgement, in the same manner as broadcast frames.

Our design breaks layer boundaries in a novel way by classifying TCP ACKs as broadcast frames and then aggregating them in the same manner as frames with broadcast addresses. This can potentially cut the number of transmissions and thus floor acquisitions needed by a TCP flow in half as well as save the significant other overheads associated with transmitting the small TCP ACK frames.

TCP ACK aggregation does not require a new frame format. Instead, TCP ACKs are categorized as link level broadcasts and transmitted in the broadcast portion of the frame. Although the TCP ACKs are transmitted as a broadcast sub-frame and thus do not generate link-level ACKs, they still have unicast MAC addresses. When a node receives a TCP ACK not addressed to it, it drops the packet, rather than passing it up the stack. This behavior is significant; if the packet was passed up the stack to the IP layer, it would attempt to deliver the packet, resulting in improper duplication of the TCP ACK. Thus, instead of broadcasting TCP ACKs to the entire network, as a typical broadcast packet, the ACKs are just broadcast within the range of the nodes along the TCP stream's path, just as they would be if the ACKs were sent as unicast packets.

4 HYDRA TECHNICAL LAYOUT

In this section, we overview Hydra and present the details of it.

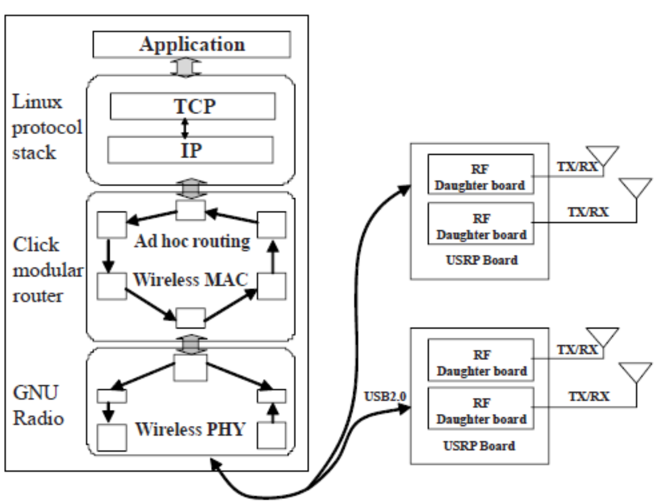

Figure 3: Block Diagram of a Hydra Node

4.1 Hydra Background

Hydra is a wireless network prototype being developed at the University of Texas at Austin [6]. Hydra was designed to allow both the PHY and the MAC to be flexible and easy to modify. This design allows us to experiment with cross-layer designs, such as the one presented here, on realistic hardware and real RF channels rather than just in simulations.

Figure 3 presents a block diagram of the main components of the Hydra including the RF frontend, the PHY, and the MAC. The programmable RF front-end is the universal software radio peripheral (USRP) [3], which interfaces to the general purpose host through a USB 2.0 connection. Figure 3 shows several USRPs with multiple antennas. Using multiple antennas enables us to exploit the MIMO capabilities of the protocols we implement. All other aspects of Hydra, the PHY, MAC, and higher layers, run on a general purpose computer running Linux.

In addition to the MAC, we implement ad-hoc routing using Click, which interfaces with the Linux TCP stack [13]. This allows us to use standard network software for experiments.

4.1.1 PHY

The Hydra physical layer essentially follows the IEEE 802.11n standard [8] and is implemented in the GNU Radio framework [4]. This open-source software allows developers to implement signal processing blocks in C++ and then flexibly connect them together using Python as a glue language. The PHY uses BPSK, QPSK, 16-QAM and 64-QAM as its modulation schemes. It supports various MIMO transmission modes including beamforming, spatial multiplexing, and cyclic delay diversity. For the experiments described here, we only use cyclic delay diversity. In addition to the standard 802.11n features, the PHY includes a link adaptation algorithm using explicit feedback. Hydra's data rates are limited due to the bandwidth of the USB and processing delay created by the software implementation of the PHY.

Thus the prototype supports physical layer data rates 10 times less than the actual data rates defined in the IEEE 802.11n standard. Table 1 summarizes the features of the Hydra's physical layer.

Table 1: Hydra PHY Details

System Bandwidth	1 *MHz**
Center Frequency	2:4 - 2:5 *GHz*
Maximum TX Power	10 *mW*
Modulation	BPSK, QPSK, 16-QAM, 64-QAM
Coding	Bit-Interleaved Binary Convolutional
SISO Data Rates	0:65*, 1:30*, 1:95*, 2:60*, 3:90*, 5:20*, 5:85*, 6:50* *Mbps*
MIMO Data Rates	2X, 3X$^+$, and 4X$^+$ SISO Data Rates
Diversity Schemes	Cyclic Delay Diversity, Space-time Coding, Spatial Mapping, Beamforming

* indicates non-standard values

+ indicates capabilities in development

4.1.2 MAC

The MAC is written in C++ using the Click modular router framework [13]. This software framework, developed at MIT, runs on a general purpose processor and was originally created for building flexible and high performance routers. Similar to

GNU radio, Click allows users to build packet processing elements in C++ and connect them using its own glue language.

The Hydra MAC follows the IEEE 802.11 MAC standard for DCF with a RTS/CTS exchange.

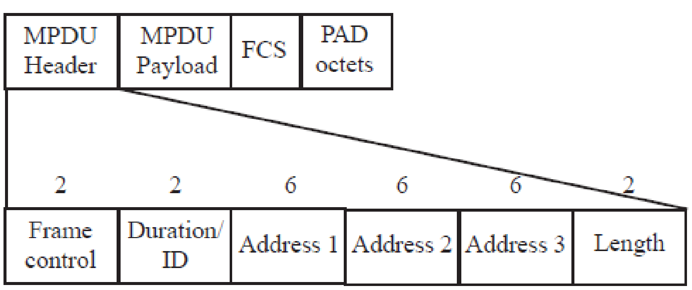

Figure 4: The MAC Subframe Format

In addition, the Hydra supports an explicit feedback scheme using the RTS/CTS exchange and rate adaptation schemes including receiver based auto rate (RBAR) and auto rate fallback (ARF) [5, 10]. The present design and experiments do not use the rate adaptation schemes.

4.2 Aggregation

We enhanced the Hydra MAC and PHY to support unicast, broadcast, and TCP ACK aggregation. We describe the MAC subframe format used by our aggregation schemes, the receive and transmit processes, and finally details of the classification of

TCP ACKs as broadcasts.

4.2.1 Frame

A single physical frame includes a series of MAC subframes. These subframes are embedded in the aggregated frame shown in Figure 2. Figure 4 shows the format of each MAC subframe. This follows the standard 802.11 MAC format with the exception that we eliminated the address 4 field because we do not support infrastructure networking. Each subframe includes a MAC header containing general information: duration, source and destination addresses, and length. Our aggregation protocol only uses the duration field of the first unicast subframe for virtual carrier sensing. However, for the purpose of easy prototyping, all of the subframes have the duration field. Each subframe includes a 2-byte length field. Finally all of the subframes contain frame check sequence (FCS) and PAD octets.

Frames transmitted in the broadcast portion of the frame can have a broadcast or unicast address but are not acknowledged. On the other hand, the subframes transmitted in the unicast portion require an ACK and thus all must be addressed to the same destination.

4.2.2 The Receive Process

When receiving a frame, the PHY uses the broadcast length and rate information to decode the broadcast subframes and then the unicast length and rate information to decode the unicast subframes. Once the PHY completes decoding all the subframes, it sends the subframes up to the MAC. When the MAC receives an aggregated frame, it first processes the broadcast subframes and then processes the unicast subframes. For the broadcast portion, as soon as each subframe passes the cyclic redundancy check (CRC), the MAC sends the subframe to the next layer. Thus the broadcast subframes do not suffer from higher loss probability though they are aggregated with unicast subframes. For the unicast subframes, the MAC checks destination address and all of the CRCs, and, if they all pass, then the MAC sends them up to the next layer and sends a link-level ACK. Otherwise, all of the unicast subframes are discarded. We could optimize by storing and applying CRCs to aggregates instead of individual subframes. However, the current scheme has only a small overhead and will allow us to extend our design to a block ACK scheme like that of 802.11n.

4.2.3 The Transmit Process

On the transmit side, the MAC must assemble the aggregated frames into the correct format. To achieve this, we have two queues: One for broadcasts and one for unicasts. The MAC first searches the broadcast queue and assembles all the broadcast frames. Then the MAC searches the unicast queue and gathers the unicast frames being transmitted to the same destination as the first frame in the unicast queue. Once completed, the MAC aggregates the broadcast subframes followed by unicast subframes up to a parameterized maximum aggregation size.

Putting the broadcasts ahead of the unicasts enables the broadcasts to be less sensitive to changes in the wireless channel. This is because the channel might change during transmission and the subframes close to the PHY training sequences are less likely to be corrupted by these changes. Once the frames are assembled, the MAC hands the aggregated frame down to the PHY along with the rate and length information for the broadcast and unicast parts of the frame. The entire transmit process triggers when the DCF of the MAC acquires the floor.

4.2.4 TCP ACKs

The process above neglects TCP ACKs, which are specially handled when assigning packets to the unicast or broadcast queues. We assign "pure" TCP ACKs to the broadcast queue. We define "pure" TCP ACK segments to be those that do not contain any data and are not part of connection set-up. Click provides a packet classification mechanism, and our implementation uses these classifiers to sort pure TCP ACKs from other unicast frames and place them in the broadcast queue.

5. TOOLS USED FOR SIMULATION

NS-2 will be used for simulation in this research work.

6. CRITICAL ANALYSIS

In this paper, the design aspects of multi-hop wirless networks using frame aggregation are cirtically analysed.

7. CONCLUSION

In this paper, design aspects of multi-hop wireless networks using frame aggregation are presented.

REFERENCES

[1] Enhanced Wireless Consortium. HT MAC Specification v1.24, 2006.
[2] S. Floyd and T. Henderson. The NewReno Modification to TCP's Fast Recovery Algorithm. IETF RFC 2582, Apr. 1999.
[3] GNU Radio: Universal Software Radio Peripheral. http://www.gnuradio.org/trac/wiki/USRP.
[4] GNU Software Radio. http://www.gnu.org/software/gnuradio/.
[5] G. Holland, N. H. Vaidya, and P. Bahl. A Rate-Adaptive MAC protocol for Multi-Hop Wireless Networks. In Proceedings of ACM MOBICOM, July 2001.
[6] Hydra { A Wireless Multihop Testbed. http://hydra.ece.utexas.edu/.
[7] IEEE 802.11 Working Group, Piscataway, NJ. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, 1997.
[8] IEEE 802.11n Working Group. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification-Draft 2.0: Enhancements for Higher Throughput, Part 11 Standard ed., 2007.
[9] D. B. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Proceedings of ACM SIGCOMM, Aug. 1996.
[10] A. Kamerman and L. Monteban. WaveLAN II: A High-Performance Wireless LAN for the Unlicensed Band. Bell Labs Technical Journal, 2(3):118{133, 1997.
[11] S. Kim, S. Choi, Y. Kim, and K. Jang. MCCA: A High-Throughput MAC Strategy for Next-Generation WLANs. IEEE Wireless Communications, 15(1):32{39, Feb 2008.
[12] Y. Kim, S. Choi, K. Jang, and H. Hwang. Throughput Enhancement of IEEE 802.11 9 WLAN via Frame Aggregation. In Proceedings of IEEE VTC, Sept. 2004.
[13] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek. The Click Modular Router. ACM Transactions on Computer Systems, 18(3):263{297, 2000.
[14] C. Parsa and J. J. Garcia-Luna-Aceves. Improving TCP Performance over Wireless Networks at the Link Layer. Mobile Networks and Applications, 5(1):57{71, Mar. 2000.
[15] C. Perkins, E. Royer, and S. Das. Ad-hoc Ondemand Distance Vector (AODV) Routing. In Proceedings of IEEE WMCSA, Feb. 1999.
[16] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic Media Access for Multirate Ad hoc Networks. In Proceedings of ACM MOBICOM, Sept. 2002.
[17] L. Scalia, F. Soldo, and M. Gerla. PiggyCode: a MAC Layer Network Coding Scheme to improve TCP Performance over Wireless Networks. In Proceedings of IEEE GLOBECOM, Nov. 2007.
[18] D. Skordoulis, Q. Ni, H.-H. Chen, A. P. Stephens, C. Liu, and A. Jamalipour. IEEE 802.11n MAC Frame Aggregation Mechanisms for Next-Generation High-Throughput WLANs. IEEE Wireless Communications, 15(1):40{47, Feb 2008.
[19] J. Tourrilhes. PiggyData: Reducing CSMA/ CA Collisions for Multimedia and TCP Connections. In Proceedings of IEEE VTC, Sept. 1999.
[20] Y. Xiao. Concatenation and Piggyback Mechanisms for the IEEE 802.11 MAC. In Proceedings of IEEE WCNC, Mar. 2004.
[21] Y. Xiao. IEEE 802.11n: Enhancements for Higher Throughput in Wireless LANs. IEEE Wireless Communications, 12(6):82{91, Dec. 2005.