

Towards Reliable Broadcasting using ACKs

Mathilde Durvy
EPFL

Email: mathilde.durvy@epfl.ch

Christina Fragouli
EPFL

Email: christina.fragouli@epfl.ch

Patrick Thiran
EPFL

Email: patrick.thiran@epfl.ch

Abstract— We propose a mechanism for reliable broadcasting in wireless networks, that consists of two components: a method for bandwidth efficient acknowledgment collection, and a coding scheme that uses acknowledgments. Our approach combines ideas from network coding and distributed space time coding.

I. INTRODUCTION

Most operating networking protocols today use the wireless links to create point-to-point connections without taking advantage of the broadcasting capabilities of wireless channels. In contrast, information theoretical work illustrates the significant benefits in terms of physical resources that can be gained by exploiting broadcasting combined with tailored coding schemes to offer reliability.

In this paper we take as an example the widely used IEEE 802.11 Medium Access Control (MAC) protocol. The 802.11 protocol perceives the random fluctuations at the physical layer as packet erasures. In the case of a unicast (point-to-point) transmission the 802.11 protocol uses the feedback provided by acknowledgment packets to decide on data packet retransmissions. In fact it is well known, that the capacity of a discrete memoryless erasure channel is achieved by employing symbol by symbol perfect feedback and repetition coding. This is the basic behavior emulated by the 802.11 protocol to provide reliability to unicast transmissions. In the case of a basestation broadcasting to N nodes, applying the same strategy would require the basestation to collect up to N separate acknowledgment packets. This would lead to significant overhead. Moreover, once the basestation receives these acknowledgments, the use of repetition coding is no longer optimal: more sophisticated coding schemes are required to achieve the capacity. As a result, the 802.11 protocol only supports unreliable broadcasting.

In this paper we examine the two aspects that need to come together for reliable broadcasting using ACKs: efficient acknowledgment collection, and design of low complexity coding schemes.

Our first contribution (Section III) is a very simple scheme to reduce overhead when collecting acknowledgments at the MAC layer. The main idea is to have nodes transmit their acknowledgment packets towards the basestation simultaneously. The basestation thus receives only one packet that is the superposition of all the acknowledgments sent. The acknowledgments packets are designed in such a way that the

basestation, by examining the single received packet, can infer which nodes have successfully received its broadcast packet.

Once the MAC layer collects the broadcast packet acknowledgment, it informs a higher layer of its contents. This layer can then decide the contents of a next information packet to transmit using coding techniques. Our second contribution (Section V) is to identify properties and discuss the design of coding schemes that exploit the use of acknowledgments. In particular, we discuss benefits that the use of ACKs can offer as compared to other approaches such as Forward Error Correction (FEC).

II. MODEL AND PREVIOUS RESULTS

A. Model

We assume a 802.11 local area network with a basestation and N nodes. The number N of nodes might vary in time. A node joining the network associates itself to the basestation and terminates this association upon leaving the network.

The 802.11 MAC protocol alternates between contention free periods where the basestation regulates the access to the channel and contention periods where the nodes compete to access the channel using carrier sense multiple access with collision avoidance (CSMA/CA). An optional handshake might be used prior to the data packet transmission to further reduce the probability of collisions in the contention period. Yet, the delivery of packets remains essentially unreliable due to the variability of the channel's quality and to the interference from other networks (or within the same network in the contention period). Broadcast packets are especially vulnerable since contrarily to unicast packets they are not acknowledged.

At the physical layer, the 802.11 protocol adds a preamble and a header to the MAC packet. It then modulates the packet. The type of modulation used depends on the version of the 802.11 protocol and on the transmission rate.

B. Related Work

MAC protocols that do not incorporate network coding use N successive acknowledgements (one ACK per receiving node) to provide reliable broadcast. In [1] the receivers compete for the channel to send their ACKs. In [2], the basestation sends Request for ACK packets successively to each node to trigger the ACK. This approach reduces collisions between ACK packets and thus the time needed to send the ACKs.

Our work falls in the framework of applying network coding to control traffic. In this vein, use of network coding techniques

This work was supported (in part) by the NetRefound FP6 FET Open Project of the European Union under grant number IST-034413, and by the Swiss National Science Foundation.

towards network monitoring was proposed and investigated in [4], [5], [6]. The work in [7] also applies network coding to ACK packets, to optimize resource utilization in systems that provide reliable transmission over wireline networks.

In the network coding literature, the idea of superimposing physical signals at a receiver to implement algebraic superposition was recently proposed in [20]; the superposition was however applied to information packets. This idea has been well-explored in the context of cooperative communication over wireless networks, see for example [8]. Coding schemes and benefits of exploiting broadcast transmissions and network coding have been examined in the literature, the closer to this work being [9], [10], [11], [19].

III. SCHEME FOR ACK COLLECTION

Our proposed scheme operates as follows. Each node within the basestation transmission radius is allocated an identity, from 1 to N . The basestation broadcasts an information packet. If Receiver i did not receive the broadcast transmission it remains silent. However, if Receiver i successfully received the broadcast transmission, it sends an acknowledgment packet whose payload is the concatenation of $N + 1$ bit sequences $[\mathbf{x}_1^i, \mathbf{x}_2^i, \dots, \mathbf{x}_{N+1}^i]$ with $\mathbf{x}_{i+1} = \alpha$ and $\mathbf{x}_j = \beta$ for $j \neq i+1$. α and β depend on the modulation scheme used at the physical layer. The modulation scheme associates to each bit sequence \mathbf{x}_j^i a corresponding symbol sequence \mathbf{s}_j^i . It is of primary importance that the symbol sequences α' and β' corresponding to the bit sequences α and β be different.

The basestation observes the superposition of the transmitted symbols sent by all the receiving nodes. Denote by P_i the transmitted power of Node i and by γ_i the channel attenuation from Node i to the basestation which is assumed to be a complex Gaussian with zero mean ¹. The received symbols $[\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{N+1}]$ can be expressed as,

$$[\mathbf{y}_1 \dots \mathbf{y}_{N+1}] = [\gamma_1 \sqrt{P_1} \dots \gamma_{N_s} \sqrt{P_{N_s}}] \underbrace{\begin{bmatrix} \mathbf{s}_1^1 & \mathbf{s}_2^1 & \mathbf{s}_3^1 & \dots & \mathbf{s}_{N+1}^1 \\ \mathbf{s}_1^2 & \mathbf{s}_2^2 & \mathbf{s}_3^2 & \dots & \mathbf{s}_{N+1}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_1^{N_s} & \mathbf{s}_2^{N_s} & \mathbf{s}_3^{N_s} & \dots & \mathbf{s}_{N+1}^{N_s} \end{bmatrix}}_{\mathbb{S}} + [\mathbf{n}_1 \dots \mathbf{n}_{N+1}] \quad (1)$$

where the \mathbf{n}_i are noise vectors with each element a Gaussian process of zero mean and unit variance. The matrix \mathbb{S} has dimension $N_s \times (N + 1)$ where N_s is the number of ACK packets sent. Note that although only N_s receivers send ACKs, the row dimension of matrix \mathbb{S} does not affect the basestation operation.

Example III.1 (BPSK). Consider a basestation transmitting information to $N = 3$ receivers, and assume that only Receivers 1 and 2 successfully received the broadcast transmission. For BPSK modulation $\alpha = 1$ and $\beta = 0$. Since BPSK

associates symbol -1 to bit 0 and 1 to bit 1, this choice of α and β satisfies the requirement $\alpha' \neq \beta'$. The ACK payload of Receiver 1 and 2 are $[0 \ 1 \ 0 \ 0]$ and $[0 \ 0 \ 1 \ 0]$ respectively. After modulation and transmission the basestation receives,

$$[\mathbf{y}_1 \dots \mathbf{y}_4] = [\gamma_1 \sqrt{P_1} \ \gamma_2 \sqrt{P_2}] \begin{bmatrix} -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \end{bmatrix} + [\mathbf{n}_1 \dots \mathbf{n}_4].$$

After sending a broadcast packet the basestation waits for the combined ACK. If it receives nothing, it concludes that none of the receivers has received the packet. Otherwise, it knows that at least one and at most N nodes acknowledge reception. Although it does not know how many of the Receivers have acknowledged, it does know, that only Receiver i may have transmitted a symbol α' at position $i + 1$. Thus, to decide whether Receiver i has transmitted, it can proceed as follows. It compares its first received symbol \mathbf{y}_1 with \mathbf{y}_{i+1} . If we ignore the noise, \mathbf{y}_1 is different from \mathbf{y}_{i+1} if and only if Receiver i has transmitted. Assuming that the noise levels are sufficiently low for successful signal reception, the basestation can successfully decide whether the symbols \mathbf{y}_1 with \mathbf{y}_{i+1} differ or not. In Example III.1, ignoring the noise, we had $y_1 \neq y_2$, $y_1 \neq y_3$, but $y_1 = y_4$ which indeed meant that Receivers 1 and 2 had received the broadcast packet but not Receiver 3.

Example III.2 (DBPSK). Consider the exact same situation as in Example III.1 but now assume that we use DBPSK instead of BPSK modulation.

The DBPSK modulator operates as follows. The first symbol of the frame equals -1 if it corresponds to bit 0 and 1 if it corresponds to bit 1. To create the symbol at position i it multiplies the signal at position $i - 1$ by 1 if the bit i is 0 and by -1 if it is 1. That is, the signal is shifted by π to indicate an input bit of 1.

The proposed scheme can be implemented using $\alpha = 11$ and $\beta = 00$. The first bit in α creates a shift of π in the corresponding signal while the second bit allows to cancel this shift. We thus have $\alpha' = 1 - 1$ and $\beta' = -1 - 1$.

At the basestation, the received baseband signal equals

$$[\mathbf{y}_1 \dots \mathbf{y}_4] = [\gamma_1 \sqrt{P_1} \ \gamma_2 \sqrt{P_2}] \begin{bmatrix} -1 - 1 & 1 - 1 & -1 - 1 & -1 - 1 \\ -1 - 1 & -1 - 1 & 1 - 1 & -1 - 1 \end{bmatrix} + [\mathbf{n}_1 \dots \mathbf{n}_4].$$

We can check that the received signal at the basestation differs from \mathbf{y}_1 only at \mathbf{y}_2 and \mathbf{y}_3 as desired.

Our scheme thus naturally extends to different modulation schemes. Moreover, note that it does not require Channel State Information (CSI) at the basestation. We discuss in more detail issues related to practical considerations in Section IV.

IV. PRACTICAL CONSIDERATIONS

A. Physical Layer

1) *Synchronization:* Ideally we would like the receivers to send their acknowledgments at the same time to guarantee

¹We consider block fading, in the sense that the channel coefficients remain constant for the duration of a packet transmission.

a correct superposition of the bits at the basestation. In practice, propagation delays, clock skews, and variations in the processing time at the receivers might impact the transmission time of the ACKs. However, one can use repetition coding to protect against limited time shifts. Denote by k the repetition factor; α_k and β_k have k additional bits compared to α and β . The basestation knows that node i received the broadcast packet if the symbol $k + 1$ of \mathbf{y}_{i+1} and \mathbf{y}_1 differs. This code allows to protect against possible k bit shifts.

Example IV.1 (BPSK with repetition factor $k=2$). Consider the exact same situation as in Example III.1, with a repetition factor $k = 2$. We have simply $\alpha_2 = 111$ and $\beta_2 = 000$.

Perfect synchronization would require both receivers to start transmitting simultaneously. Assume, however, that Receiver 2 is faster and send its ACK before Receiver 1 (the opposite is also possible). At the basestation, we have

$$\begin{aligned} [\mathbf{y}_1 \dots \mathbf{y}_4] &= [\gamma_1 \sqrt{P_1} \quad \gamma_2 \sqrt{P_2}] \cdot \\ &\begin{bmatrix} * & * & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 \end{bmatrix} \\ &+ [\mathbf{n}_1 \dots \mathbf{n}_4], \end{aligned}$$

where “*” reflects a unit time shift and the last two symbols sent by Receiver 1 are neglected. The basestation compares the last symbol of \mathbf{y}_1 , \mathbf{y}_2 , \mathbf{y}_3 and \mathbf{y}_4 to infer which nodes received its packet.

2) *Relative Power*: One of the practical challenges of our proposed approach, that is common to all systems that employ cooperative communication via distributed space time coding, is that the complex channel fading coefficients γ_i may have quite different magnitudes. As a result, quantization at the reception end of the basestation might lead to significant errors. There are two approaches employed in the distributed space time coding literature to address this problem, and both can also be applied here:

I) *Power adaptation*, where the basestation directs the mobile nodes to increase or reduce their transmission power. This ensures that all receivers contribute a comparable power to a received symbol.

II) *Analog domain operations*, that is, perform the required comparisons on the analog received signals. In this case, the basestation needs to use a delay line, to store the analog signal \mathbf{y}_1 and be able to compare it with signals \mathbf{y}_i . The comparison can be performed by subtracting the signals in the analog domain, and using a threshold to decide if the remaining analog component consists only of noise or not.

B. MAC Layer

At the MAC layer, our approach requires the following four main modifications. First, when a node (that supports acknowledged broadcasts) joins the network, the basestation should give this node an identity from 1 to N . This identity will remain valid as long as the node is associated with the basestation. Second, there should be a way to distinguish between broadcast packets that require ACKs and broadcast

packets that do not require ACKs (e.g., this distinction could be made based on the type and subtype of the packet). Third, we need to create a new acknowledgment frame. This ACK frame would be used only to acknowledge broadcast packets. It would have essentially the same format as the current ACK frames but would include, in addition, a data body. And finally, the MAC layer should be aware of the modulation used at the physical layer as well as of the repetition factor that needs to be used. Note that these modifications allow the coexistence of reliable and unreliable broadcast so that our scheme is backward compatible.

V. CODES FOR MULTICASTING WITH ACKS

Again we consider the simple scenario of a basestation that has k packets to transmit to all the N nodes within its transmission radius. The nodes use ACKs to signal successful reception. This section examines the second component of providing reliability, that is, coding schemes. Such schemes are effectively algorithms run at the basestation that use the feedback information from all received ACKs to decide what to broadcast next.

In linear coding, each broadcast transmission is a linear combination of the information packets over a finite field \mathbb{F}_q . The size of the field q is called the *alphabet size*, and affects the encoding and decoding complexity². In addition to alphabet size, coding schemes can also be evaluated in terms of algorithmic complexity, achievable rate and experienced delay.

We say that a coding scheme is *rate optimal*, if every node that successfully receives exactly k broadcast transmissions can retrieve the k information packets. Each coding scheme also has an associated *delay*, which measures how fast a receiver can decode information packets. Delay is affected by the rate, i.e., how fast the node receives useful information, and also, since in general received packets are linear combinations of information packets, the time the node needs to wait before it is able to actually decode the information packets. For example, assuming equal rates, a coding scheme where each received coded packet allows to immediately decode one information packet, has smaller delay than a coding scheme where the receiver needs to wait to receive all k packets before decoding. The later is the case of packet-level FEC schemes. Use of ACKs, as compared to FEC, may allow to achieve different performance trade-offs.

We start this section by connecting our code design to code design over appropriately defined random graphs; as a side result we provide an alternative proof (to the one originally proposed in [17]) of the benefits network coding offers in the coupons collector problem. We then identify properties of the optimal schemes, and propose new algorithms for code design.

A. Random Graphs and Coding

We model the downlink channels between the basestation and each of the N receivers (as perceived at higher layers)

²Operations over a field \mathbb{F}_q of size $q = 2^n$ require with typical algorithms $\mathcal{O}(n^2)$ binary operations and with the faster implemented algorithms $\mathcal{O}(n^{1.59})$ binary operations.

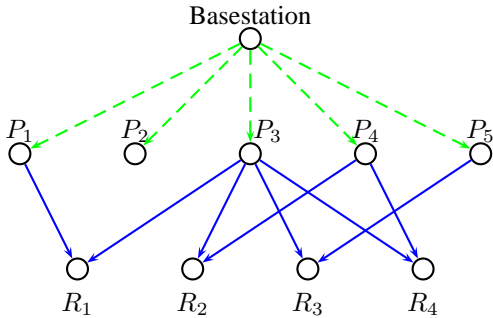


Fig. 1. A random graph realization where at every packet transmission, the channel realizations keep or erase edges. The first packet P_1 is received only by R_1 , the second packet P_2 by none, the third packet P_3 by all receivers, etc. The packets are generated by the basestation.

by iid erasure channels. This random process gives rise to the random graph depicted in Fig. 1, that connects a vertex corresponding to the basestation, to N vertices corresponding to the receivers, through vertices corresponding to time-slots. The basestation, using the k information packets, broadcasts at time slot i the coded packet P_i . An edge exists between time-slot i and receiver R_j if and only if R_j successfully receives the packet P_i . We distinguish three cases:

(a) *No knowledge*: If the basestation has no knowledge of which packets get successfully received, that is, no knowledge of the random graph structure, it has to employ FEC. In general, FEC schemes achieve the optimal rate asymptotically in the number of information packets k that are linearly combined, which may result in large delay. Moreover, FEC schemes require higher complexity decoders than ACK schemes.

(b) *Perfect knowledge*: Assuming that the basestation has perfect a-priori knowledge of the graph structure, the problem reduces to that of multicasting over a fixed lossless network, and all the related results for network code design in the literature apply.

(c) *Causal knowledge*: This is our case of interest. The basestation does not have channel state information (CSI), but does receive ACKs, for example using the mechanism we propose at the MAC layer. For simplicity, we assume that the basestation has perfect feedback, i.e., for each broadcasted packet, it knows which nodes have successfully received it. Having perfect feedback at time t implies that, when we need to decide the coding operation at time t , we know the graph realization up to time $t - 1$. We investigate coding schemes for this case in the following sections. We will show that to guarantee rate optimality, we may need to use an alphabet of size N . We will then discuss rate-optimal and delay-optimal schemes for two receivers, and possible extensions to N receivers. Before that, we use the random graph representation of our model to provide an alternative proof for the coupons collector problem.

B. Random Graphs and the Coupons Collector Problem

In the coupons collector problem, coupons of k types are placed uniformly at random inside boxes of some commodity. A collector needs to buy on the average $O(k \log k)$ boxes

to collect a coupon of each type [16]. Deb and Médard [17] proved that allowing random linear combinations of the coupons to be placed inside each box reduces this number to $O(k)$. Here we show that this result can be retrieved as a corollary of the main theorem in network coding [12].

Alternative Proof of Coupons Collector Problem

(i) We first note that the coupons collector problem can be reduced to a flow problem, using the random graph representation in Fig. 1 and assuming no knowledge of the graph realization as in case (a). The source, that has the information packets (coupons), places random combinations of them inside each coded packet (box) P_i . Each node R_i receives (buys) a (not-determined in advance) set of P_i 's.

(ii) In the information theoretic proof of the main theorem in network coding [12], it was shown that if all nodes in the network randomly combine their incoming flows and transmit them to their outgoing edges, no matter what the network topology between the source and the destination is, provided that the transmitted rate equals the min-cut to each receiver, we can achieve this rate.

In the coupons collector problem we do not know the network structure, but we do know that the min-cut to each receiver (number of packets each receiver collects) equals k . Thus the result follows. \square

C. Alphabet Size for Rate-Optimal Codes using ACKs

Theorem V.1. *Assuming perfect feedback and no CSI, an optimal code may require encoding operations over a finite field of size N , where N is the number of receivers.*

Proof. Assume that we have two packets x_1 and x_2 to transmit to N receivers. As discussed in [14], we can restrict our choice of coding vectors to the following set of $q+1$ vectors in general position over the field \mathbb{F}_q , with α is a primitive element of \mathbb{F}_q :

$$\mathcal{A} = \{[01], [10], \text{ and } [1\alpha^i] \text{ for } 0 < i \leq q-1\}.$$

Any two vectors in this set form a basis of the two-dimensional space \mathbb{F}_q^2 . If the basestation had a-priori the bipartite graph realization it would need $q+1 = O(\sqrt{N})$ such vectors [14]. In our case, however, at transmission t , we only know the graph realization up to transmission $t-1$. We will show that in this case an optimal code may need to use an alphabet of size N . Indeed, consider the following sequence of channel realizations:

- The basestation uses the coding vector $[01]$. Only R_1 successfully receives it.
- The basestation now needs to use a different vector in the set \mathcal{A} , for example the vector $[10]$, to ensure that R_1 does not receive the same vector twice (otherwise the code is not optimal). Assume that only R_2 successfully receives it.
- Continuing along these lines, at transmission k , with $3 \leq k \leq N+1$, the basestation needs to use a different vector in the set \mathcal{A} from the $k-1$ previously employed, to ensure that none of the receivers R_1, R_2, \dots, R_{k-1} receive the same packet twice (once a receiver decoded both packets, the corresponding vector can be reused). Thus the set \mathcal{A} needs to have size $q+1 \geq N+1$ and the result follows. \square

Theorem V.2. *Assuming perfect feedback and no CSI, there exist polynomial algorithms to construct codes that use an alphabet of size N .*

Proof. We can simply use the linear information flow algorithms proposed in [15]. This algorithm maintains for each receiver R_j a set of coding vectors, that is updated as the algorithm evolves. To assign a coding vector at transmission t , we can assume conservatively that all receivers (that have not yet received their mincut value) are going to successfully receive the transmitted packet. Using the feedback information, we can then simply update the sets of the receivers that indeed received this transmission. \square

D. Rate-Optimal Codes for Two Receivers

For the case of two receivers there exist rate-optimal binary deterministic codes, which also guarantee to result in the minimum delay. Such are the codes in [18], which we describe in a slightly different language in the following.

For each receiver R_i the basestation keeps a list S_i , $i = 1, 2$. Initially both lists are empty. If R_i successfully receives a packet x_j , then x_j is added to S_i . A packet x_j appearing in both lists S_1 and S_2 is removed from both lists. Thus S_i contains the list of packets that R_i has successfully received, but the other receiver has not. In other words, all packets in the list S_1 are useful for R_2 , while all packets in list S_2 are useful for receiver R_1 . The basestation can then operate as follows.

- If S_1 and S_2 are not empty, the basestation broadcasts the xor of two packets, one from S_1 and another from S_2 . Note that if any of the receivers successfully receives the broadcasted transmission, it will be able to directly decode one packet. This packet will be added to its list.
- If either S_1 or S_2 or both are empty, the basestation transmits a packet that has never been transmitted. If such a packet does not exist, then at least one of the receivers has successfully received all packets. The basestation can transmit the packets missing from the other receiver using repetition coding.

E. Rate-Optimal Codes for N Receivers

A rate-optimal coding solution at time t can be thought as a feasible solution to a set-cover problem. This problem takes as input a universe of N elements (the N receivers), k sets of elements of the universe (where set i contains the receivers that have not received³ packet i at time t) and selects, according to some criterion to be discussed in the following, a *cover*, a set of packets such that each receiver has not received at least one of them.

The criterion we would like to optimize, in our choice of cover, is delay. However, unlike the case of $N = 2$ receivers, for $N \geq 3$, and depending on the channel realizations, there might not exist a coding choice such that each successfully received transmission allows to decode one information packet.

³If a receiver receives a linear combination of two or more packets, we can remove the receiver from the list of one of them. Which one does not affect the rate-optimality, but might affect the delay.

Moreover, there might not exist a choice that will clearly lead to the smallest delay, as the following example argues.

Example V.1. *Assume that at time t , R_1 has not received the packets $\{P_1, P_2\}$, and R_2, R_3 have not received the packets $\{P_1, P_3\}$ and $\{P_2, P_3\}$ respectively. For a rate-optimal code with minimum delay, we should transmit one of the combinations $\{P_1 + P_2, P_1 + P_3, P_2 + P_3\}$. Which one will actually result in the minimum delay depends on the unknown future channel realization.*

Given this, we can choose between several possible criteria to optimize delay. For example, let $m_i \geq 1$ denote the number of times that receiver R_i appears in the cover, we can select the cover to $\min \max_i m_i$, or $\min \sum m_i$, etc. Comparing these criteria through simulations is part of future work.

VI. CONCLUSIONS

In this paper we proposed a mechanism to efficiently collect ACKs when broadcasting in a wireless environment, and discussed properties of companion coding schemes.

REFERENCES

- [1] K. Tang, M. Gerla, "MAC reliable broadcast in ad hoc networks", *MILCOM*, 2001.
- [2] M. Sun, L. Huang, A. Arora, T. Lai, "Reliable MAC layer multicast in IEEE 802.11 wireless networks", *ICPP*, 2002.
- [3] "IEEE802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", p. 42, 1999.
- [4] C. Fragouli and A. Markopoulou, "A network coding approach to network monitoring", *Allerton*, 2005.
- [5] T. Ho, B. Leong, Y. Chang, Y. Wen and R. Koetter, "Network monitoring in multicast networks using network coding", *ISIT*, 2005.
- [6] C. Fragouli and A. Markopoulou, S. Diggavi, "Topology inference using network coding", *Allerton*, 2006.
- [7] C. Fragouli, D. Lun, M. Médard, P. Pakzad, "On feedback for network coding", submitted to *CISS*, 2007.
- [8] F. Fitzek, M. Katz, "Cooperation in Wireless Networks: Principles and Applications", Springer 2006.
- [9] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, J. Crowcroft, "XORs in the air: practical wireless network coding", *ACM SIGCOMM* vol. 36, iss. 4, pp. 243-254, 2006.
- [10] C. Fragouli, J. Widmer, J.Y. LeBoudec, "A network coding approach to energy efficient broadcasting: from theory to practice", *Infocom*, 2006.
- [11] A. Eryilmaz, A. Ozdaglar, M. Médard, "On delay performance gains from network coding", *CISS* 2006.
- [12] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow", *IEEE Trans. Inform. Theory*, vol. 46, iss. 4, pp. 1204-1216, July 2000.
- [13] T. Ho, R. Koetter, M. Médard, M. Effros, J. Shi, and D. Karger, "A random linear network coding approach to multicast", *IEEE Trans. on Inform. Theory*, vol. 52, iss. 10, pp. 4413-4430, October 2006.
- [14] C. Fragouli and E. Soljanin, "Information flow decomposition for network coding", *IEEE Trans. Inform. Theory*, vol. 52, iss. 3, pp. 829-848, March 2006.
- [15] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction", *IEEE Trans. Inform. Theory*, vol. 51, iss. 6, pp. 1973-1982, 2005.
- [16] M. Mitzenmacher and E. Upfal, "Probability and Computing", Cambridge University Press, 2005.
- [17] S. Deb and M. Médard, "Algebraic gossip: a network coding approach to optimal multiple rumor mongering", *Allerton*, 2004.
- [18] Y. Wu, P. A. Chou, S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast", *CISS* 2005.
- [19] Y. Wu, "A trellis connectivity analysis of random linear network coding with buffering", *ISIT*, 2006.
- [20] S. L. Zhang, S. C. Liew, and P. P. Lam, "Physical-Layer Network Coding", *ACM Mobicom'06*, Sept 2006.