XORs in the past and future

Muriel Médard
MIT
medard@mit.edu

Sachin Katti
Stanford
skatti@stanford.edu

Dina Katabi
MIT
dina@csail.mit.edu

Wenjun Hu
Yale
wenjun.hu@yale.edu

Hariharan Rahul
MIT
rahul@csail.mit.edu

Jon Crowcroft
Cambridge University
jon.crowcroft@cl.cam.ac.uk

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

While placing the paper "XORs in the Air" in the context of the theoretical and practical understanding of network coding, we present a view of the progress of the field of network coding, In particular, we examine the interplay of theory and practice in the field.

CCS CONCEPTS

• **Mathematics of computing** → **Coding theory**; • **Networks** → **Error detection and error correction**;

KEYWORDS

Network coding

1 THE CHRYSALIS.

When our paper "XORs in the air" [16], [17] appeared, network coding officially emerged from its information theoretic chrysalis stage to, well, take to the air. The paper [3], of the late Ahlswede, and of Cai, Li and Yeung, had first presented the concept of coding in the network and had put forth the now famous butterfly example, that we show in the figure below. That example, like its eponymous insect, is both beautiful and fragile. Beautiful because it is simple, memorable and accessible. Fragile because the very simplicity beguiled many researchers to believe that it represented the essence of network coding. A considerable number of works undertook hunting for butterflies in networks (see Myth #1 in [27]). More sophisticated ones, noticing that the solution to the butterfly network was the coding together of two trees, sought to construct trees and code them together, but rapidly wandered into the forest of Steiner tree problems that have so stumped uncoded multicast communications.

The *engouement* for butterflies was no doubt also rooted in the fact that explicit constructions for arbitrary networks were not available in the original work of Ahlswede et al. Such constructions required network coding to be framed in a way that was tractable, at least to the community that already for half of a century had been constructing codes. The language of that community is largely algebraic, with some probabilistic elements. The 2001 ISIT and 2002 INFOCOM paper by the late Ralf Kötter and one of the authors of this paper [18, 19], that later appeared in IEEE/ACM Transactions

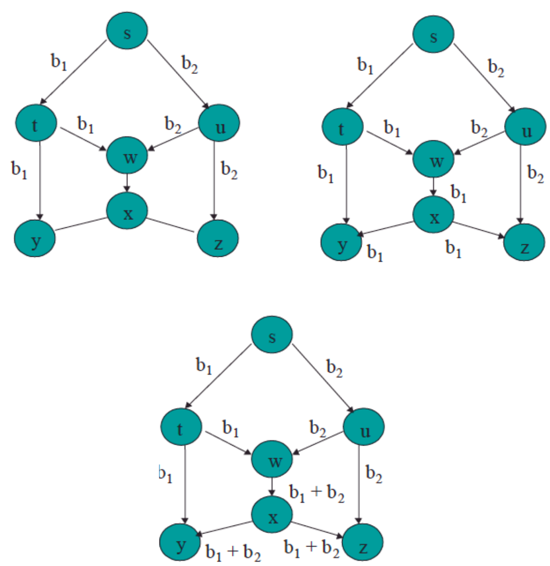

Figure 1: The butterfly example

on Networking in 2003 [20], framed network coding as an algebraic problem.

2 AN ALGEBRAIC COCOON

A network has edges with directions, so that bidirectional links are represented by two directed edges. The effect of capacity is also modeled by network topology, with higher capacity links being mapped to multiple parallel links. For linear network coding, nodes in a network perform linear operations, defined over some appropriate finite field, of which there will be more anon. The nodes map their inputs, which are data vectors, $\underline{X}$ represented over whatever finite field $\mathbb{F}_q$ is appropriate. If our operations are over bits, we are in $\mathbb{F}_2$, as bytes when we are in $\mathbb{F}_{2^8}$. Linearity being endlessly composable, the effect, over an entire network, is itself linear. The crux of the network coding problem is in characterizing that linear mapping across the network.

Consider the representation of a network as an edge incidence graph,say F, where to each edge corresponds a row and a column in F. A nonzero entry at the (i, j)th entry of F indicates that the ith edge is incident upon the jth one, i.e., that the endpoint of the ith edge is at a node that is the start of the jth edge. The code is then the values, over the field of choice, of the elements of F. If edge j carries the weighted sum of its incident edges i and i', then the process, call it Y_j, over that edge is $f_{i,j}Y_i + f_{i',j}Y_{i'}$. Let us now consider, in light of this F, how transmission over a network acts upon data. If the data remains at a node, it is equivalent to applying the identity matrix to $\underline{X}$. A single hop through the network has the effect of multiplying $\underline{X}$ by F. In general, any n-hop traverse through the network corresponds to multiplying $\underline{X}$ by F^n.

Note that the notion of route or path changes relatively to a traditional, routing based network, whence the use of traverse rather than route. The total effects of all traverses through the network, of n of fewer hops, is thus naturally $I+\Sigma_i^n F^i$. If we consider all traverses of the network, n can be as large as the depth of the network (if there are no cycles), hence the overall effect of the network is $I + \Sigma_i^\infty F^i$, that can be rewritten as $(I - F)^{-1}$. Note that cycles can be accommodated, but then there needs to be a notion of time, to avoid a network input's being dependent, via a cycle, anticausally on itself. The problem can be readily addressed if delay at nodes is taken into account. The addition of a technicality, where the algebraic representation is no longer scalar, but in terms of polynomials in D, where the power of D represents the accumulated delay, suffices. Note that then one no longer requires cycle-free traverses.

But what of routing, then? The connection between coding and routing in a network, by the mid 2000s, was confused. On the one hand, the algebraic network coding paper [20] provided a clean mathematical generalization of routing. Consider the action of a switch at a node. At any one time, a traditional, point-to-point switch connects an input port to one output port. Routing is thus, at any one time, a degenerate form of coding, where an edge carries only the data from one outgoing edge. In particular, that means that each row or edge in F can have only one non-zero entry, and, if such an entry exists, it is 1. Switching may be merely seen, at any time, as a code using a transfer matrix composed of permutation matrices. Given that any constraint on a problem might not be detrimental, but cannot be advantageous, it seems almost tautological that one cannot do worse with coding than with routing. Of course, the extra flexibility of coding over routing affords also provides fresh opportunities for poor engineering decisions, and for many years, and even until recently, there were a good number of papers announcing results purporting to show that coding performed worse than routing. With the great power of coding comes the great responsibility of good engineering (see Myth #5 in [27]).

3 A MESSY METAMORPHOSIS

The issue of the difficulties in coordinating coding with routing could not be dismissed, however, entirely as poor engineering decisions caused by limited understanding of mathematical underpinnings. On the theoretical side, multiple challenges remained, and still do. The taxonomy may be summarized as follows. For the multicast case (which includes point-to-point as a degenerate case), necessary and sufficient conditions can be readily stated for a set of connections to be feasible and a simple, distributed, practical algorithm exists for code construction. The problem of finding a minimum cost subgraph satisfying the min-cut max-flow conditions from the source(s) to the destinations in the multicast case requires only linear constraints. As befits the fact that coding is a relaxation of the constraints of routing, the problem of finding a minimum cost subgraph for multicasting is a relaxation of the integrality constraints that render the directed convex combination of Steiner trees NP-complete [21, 22]. For more general cases, where multiple flows with different sources and destinations share a network, the general problem remains open. For the general connection case, conditions are generally only sufficient, useful ones can be found only in special cases, and construction is generally difficult.

For a point-to-point single flow coding is not required. The celebrated Ford-Fulkerson condition states that, in a network represented by a graph, the minimum cut of a network between a source and destination governs the maximum flow between those nodes. For a single flow meant for multiple receivers, i.e., the multicast case, Ahlswede et al [4] showed that a necessary and sufficient condition was that the min cut from the sender (or senders) to reach individually each receiver exceed the flow. This result in effect generalized the single flow point-to-point Ford-Fulkerson necessary and sufficient condition. The proof was one of equivalence, without an algorithm for construction of codes.

The algebraic network coding papers of 2001-2002-2003 started by showing that one could express the point-to-point single flow Ford-Fulkerson theorem in an algebraic way. It may seem that considering from an algebraic standpoint a problem that requires no coding is superfluous, yet it is essential to establish the equivalence between flow, a physical condition, and an algebraic one. Such correspondence is the crucial articulation between network coding and traditional, routing-based networking, whose roots are to be found in the queueing and optimization literature borne out of operations research, often originally motivated by transportation and manufacturing systems. Recall our $(1-F)^{-1}$ matrix. For the receiver to be able to recover the input $\underline{X}$ from the output $\underline{X}(1 - F)^{-1}$, it must be that $(1 - F)^{-1}$ is invertible. That invertibility is thus equivalent to the min-cut max-flow condition of Ford-Fulkerson. Such invertibility is equivalent to requiring that the determinant of the matrix, which is a multinomial in the $f_{i,j}$s, be non-zero. This is the algebraic consequence of the fact that switching suffices, coding is not necessary, in the point-to-point case. It is enough to set the $f_{i,j}$s to be zero or non-zero and, when they are non-zero, selecting to be 1 is as effective as any other non-zero choice. Note that, since no coding is required, the point-to-point case imputes no constraint on field $\mathbb{F}_q$. For a few years after its appearance, this seemingly simple connection between point-to-point connections and invertibility was sufficiently mystifying for both coding theorists and networking researchers to lead to results examining the requirements of minimum field sizes in point-to-point connections, even though no such minima make sense.

The connection between network transfer matrix invertibility and Ford-Fulkerson for point-to-point, while in itself not useful from an engineering perspective, does provide a ready proof of the Ahlswede et al [3] multicast result. From the point of view of each receiver in the network, the network's effect can be encompassed locally as a submatrix representing the overall effect of the network's linear operations upon the inputs. For that receiver, the goal of recovering the $\underline{X}$ is reduced to inverting the sub-matrix corresponding to the algebraic impulse response of the network seen between the sender(s) from which $\underline{X}$ emanates and that receiver. For the multicast case, the matrix $(I - F)^{-1}$ services multiple receivers. While for any single receiver the binary choice of zero or non-zero for each $f_{i,j}$ suffices, the choice made for one receiver may impact another receiver. For the point-to-point case, the choice of coefficients in linear network coding was not important. Algebraically, the interaction among receivers prompted by multicast connections fortunately lends itself to a ready model.

It is in the multicast setting no longer a matter of ensuring non-nullity of a single determinant, as in the point-to-point case, but, rather, each receiver must have a non-zero determinant. Thus, it is the product of the determinants for the submatrices corresponding to the different receivers that must be nonzero. Such a product of determinants is itself a multinomial in the $f_{i,j}$s, but that multinomial is of higher degree than the determinants of the individual submatrices. Clearly, each submatrix's determinant must be a non-zero multinomial, otherwise the Ford-Fulkerson condition between source(s) and the receiver, or destination, in question would fail and the connection would not be viable by itself, even in the absence of other receivers. The fact that each submatrix determinant is nonzero means that there exist choices of the $f_{i,j}$s that make the determinant be non-nil. For the product of submatrix determinants to be nonzero, it must be that the choices of $f_{i,j}$s are compatible. Here is the rub of coding. Take the simple monomial x. It is a trivially a non-zero, quite degenerate multinomial. Over $\mathbb{F}_2$, We need only choose x to be 1. Consider now the euqally simple $x + 1$. The binary choice to have a non-nil realization is 0. Now what if we wish for both x and $x + 1$ to be non-zero, or, equivalently, $x + X^2$ to be non-zero? We need to operate over a field $\mathbb{F}_q$ for $q \neq 2$. Coding for multicast, hence for satisfying simultaneously that several determinants must be nonzero, leads to the possible need to code, and to code over a non-binary field. The field size grows modestly with the number of receivers, linearly at most.

The above discussion also illustrates how random linear network coding (RLNC) works [14]. If we operate over a $\mathbb{F}_q$ with a large enough q, then the probability that a multinomial evaluates to 0 if we select uniformly and independently the $f_{i,j}$s over $\mathbb{F}_q$ decays as $O\left(\frac{1}{q}\right)$. In the previous example, we need only avoid 0 and 1. If the number of choices for values of x, i.e., the value of q, is large enough, then with high probability the outcome of selecting the values uniformly at random will lead to a non-zero product of submatrix determinants, hence, to each submatrix being invertible, and, finally, to each receiver being able to receive all of the sources' data. Each node in the network can thus select randomly and independently coefficients for combining data, without the need to refer to the topology of the network, or the action of the other nodes. Recall that it is only the total effect of the network, embodied in $(I - F)^{-1}$, and not the individual coefficients, that need be known at the receiver.

4 A FIRST EMERGENCE

Let us return, then to XORs in the air [16], [17]. On the one hand, multicast connections can be readily managed. On the other hand, traditional networks are using routing, with a multiplicity of co-existing routes. In wireless networks, a natural linkage between multicast and traditional routing emerges through the fact that the wireless medium is inherently a broadcast one. When a node transmits in the wireless domain, all nodes within reception range will receive that transmission, whether that transmission was intended for them or not. The main inspiration is then the following. If node 1 has received a packet, say a, and if node 2 has received another packet, say b, but the routing, which we assume has been already done independently of coding considerations, requires node 1 to receive b and node 2 to receive a, then we may create a local multicast connection. This multicast connection is of a and b, and has nodes 1 and 2 as receivers. Any transmission, by a node within range of both node 1 and node 2, of a coded version of a and b with non-zero coefficients, such as $a + b$, will suffice to allow both nodes 1 and 2 to recover the packets that the routing mechanism had intended for them to receive. Thus, a node that was tasked to route b to node 1 and a to node 2 can readily send $a + b$, thus saving a transmission and naturally relieving congestion.

As with the butterfly example, the above example's simplicity and ease of explanation is sufficiently attractive to be misleading in multiple ways which we discuss below. First, most of the gains do not emerge from coding two packets together over a subnetwork consisting of nodes 1 and 2, discussed above, and a third node, say node 3, in between the two, to transmit $a + b$. Rather, as made clear in the original paper, the bulk of the gains of XORs in the air arise from coding three of more packets together. The simplicity of the two-packet example led several follow on works to consider coding together only two packets. The gains are far more considerable than if pairs of packets of packets are coded together. The gains of XORs in the air require the consideration of a five-node cross network, with a single node in the middle, to account for the bulk of the effects that are seen empirically in a mesh network [6].

Second, the multicast aspect in XORs in the air arises from wireless medium access control (MAC) and the network coding is managed below the routing layer, which operates as before. Some work tried applying XORs in the air to wireline systems, by introducing an artificial broadcast of packets to the neighbors of a node. This approach creates more traffic, rather than reducing it, and is a poor use coding. Note that the explanation of the results seen in XORs in the air was missing from our original work. It was not until years later that the general shape of the curves could be recovered theoretically via a tractable but sufficiently accurate model of the Wifi MAC [42]. That MAC model was empirically validated with simple canonical topologies [43].

Third, the independence of routing and network coding in XORs in the air was done to be backward compatible with existing routing. Recall, from our general taxonomy of network coding difficulty, that the general problem of coding multiple connections over a network is ornery. Some works, even in prestigious venues, combined routing and XORs in the air by creating opportunities for coding. This is not as evidently inappropriate as applying XORs in the air to wireline networks but still misses the fact that XORs in the air seeks to use coding to relieve, in effectively an opportunistic fashion, congestion occurring at nodes as a result of routing decisions. A node that is congested is more likely to be at the intersection of multiple streams and thus have more opportunities to perform coding. Creating congestion in order to relieve it with coding is not, however, a wise decision.

A fourth misunderstanding of the results is that binary codes sufficed. Rather, the sufficiency of XORs is tied to the fact that the multicasting that takes place is over a degenerate topology, a subnetwork of small depth, with a single node per subnetwork performing the coding. Network coding sees its full potential when there is a network, and such full potential generally requires codes that are more sophisticated than XORs. Consider one of the simplest versions of a network, namely a daisy chain of n nodes. Assume that the loss probability of a packet on any of the $n-1$ links is p and assume for simplicity that the losses within and across links are all mutually independent. The minimum cut of that network is $1-p$, the throughput on any of the links in the chain. If coding occurs only in an end-to-end fashion, then the throughput is $(1-p)^{n-1}$, which decreases exponentially with n. To achieve the min-cut, we can use RLNC at each node independently, or we can perform selective repeat of lost packets at each link. Note therefore network coding is inherently different from traditional structured end-to-end codes such as Reed-Solomon or Fountain codes (this misunderstanding is discussed at Myth #6 in [27]). If one were to attempt to use a code with structure for recoding, the structure would rapidly be lost. It is possible, however, to mix RLNC with structured codes [13], [28], as any structured part of a code is simply a realization of a random code and thus compatible with the RLNC framework.

Let us now consider a somewhat richer topology. Let us say that the ith layer of the network consists not of a single node, with a single incoming link and a single outgoing link, but of two nodes receiving, over a wireless network, packets from a single node at layer $i-1$. In this case, selective ARQ will be problematic. Say the top node at layer i received a packet and the bottom node also receives it. Which one should send and what protocol should be used to avoid duplicate transmissions from layer i? Say the top node received the packet but the bottom node did not, or vice versa, how should this information be shared to make sure the information is forwarded from layer i to the next layer? What is the overhead, in terms of transmission of control packets and associated time spent without transmitting new information, of effecting coordination? If instead RLNC is used, then with high probability coded packets representing independent equations will be sent from each of the nodes at layer i, with no coordination beyond assigning a frequency of transmission for each of the nodes. In this manner, the min-cut can be reached. Note that the min-cut in the wireless domain where there is a broadcast link should be viewed, when considered in terms of a graph, as a hyperedge with a single head and multiple tail nodes [9]. The hyperedge counts for a single unit in the min cut and can be readily represented in the edge-incidence matrix F. Other approaches, which instead use edges and create elaborate accounting of repeat information across them, are needlessly complex and provide no benefit.

5 TAKING FLIGHT

The ability of RLNC to provide erasure correction is central to the discussion above. It can be merged with XORs in the air, either by replacing XORs with RLNC, since XORs are a degenerate form of RLNC, or by having RLNC at one layer correct for erasures, while XORs operates at another layer assuming that erasures have been effectively managed by RLNC [30]. In our simple example of three nodes, say, as was the case in our original paper, that node 3 assumed, though overhearing, that node 1 had received a, but that the reception did not occur. The erasure correction feature of RLNC can provide the ability to receive packet a.

The paper XORs in the air proved, probably in a way that was for the first time convincing, that using network coding was a practical option for networks. It showed that using a locally multicast approach in networks where the connections are not inherently multicast is an effective idea [21]. In effect, some connections are coded together and then artificially merged into a multicast, using a code, which should be RLNC in general, but sufficed to be XOR in the degenerate case of a single coding node per subgraph. The approach of completing to multicast has been considered in multiple networks [7, 8].

The credibility that XORs in the air provided to network coding in many ways eased its way into adoption. There were still some challenges. Some common misunderstandings regarding network coding, particularly in wireless networks, have already been discussed in this note. Some of the most common and egregious ones can be found in [27], such as the misconception that network coding can only increase for up to a factor of 2 capacity in wireless settings (Myth # 3) or that selective repeat obviates the need for coding (Myth # 8). The second, somewhat related one, was the challenge of implementation in a way that was fast, portable and easy to use, a network coding library. The latter is now available and implemented over a wide array of platforms [1]. While XORs in the air used no feedback, combining feedback and network coding [23] can provide effective delay guarantees, by trading off in-order delivery delay with throughput, with simple code constructions [10, 15, 24–26, 29, 35, 41]. The fact that RLNC, unlike traditional end-to-end codes such as block codes or Fountain codes, can be used in a sliding window [39, 40] fashion for transport protocols such as TCP [33, 34] means that they are being considered in IETF/IRTF [31], [11], [12]. Multicast applications in 5G have motivated the consideration of the use of RLNC in 3GPP, in particular as, even when used just as rateless codes rather than as full network codes, they can significantly outperform traditional rateless codes such as Raptor [37, 38]. They are also currently being tested for V2X applications [36] by the US Department of Transportation. RLNC is also being considered for multi-source downloading by multimedia companies, as demonstrations and measurements of its usefulness have evolved [5, 32] from the early theory [2].

REFERENCES

[1] http://www.steinwurf.com/products/kodo.html.
[2] S. Acedański, S. Deb, M. Médard, and R. Koetter. 2005. How good is random linear coding based distributed network storage?. In *Proc. 1st Workshop on Network Coding, Theory, and Applications (Netcod'05)*.
[3] R. Ahlswede, Ning Cai, S.-Y.R. Li, and R.W. Yeung. 2000. Network information flow. *Information Theory, IEEE Transactions on* 46, 4 (2000), 1204–1216. https://doi.org/10.1109/18.850663

[4] R. Ahlswede, Cai Ning, S-Y.R Li, and R.W. Yeung. 2000. Network information flow. 46, 4 (July 2000), 1204–1216.
[5] P. J. Braun, D. Malak, M. Medard, and P. Ekler. 2019. Multi-Source Coded Downloads. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*. 1–7. https://doi.org/10.1109/ICC.2019.8761983
[6] J. Cloud, L. M. Zeger, and M. Medard. 2012. MAC Centered Cooperation — Synergistic Design of Network Coding, Multi-Packet Reception, and Improved Fairness to Increase Network Throughput. *IEEE Journal on Selected Areas in Communications* 30, 2 (February 2012), 341–349.
[7] Y. Cui, M. Médard, E. Yeh, D. Leith, and K. R. Duffy. 2018. Optimization-Based Linear Network Coding for General Connections of Continuous Flows. *IEEE/ACM Transactions on Networking* 26, 5 (Oct 2018), 2033–2047. https://doi.org/10.1109/TNET.2018.2865534
[8] Y. Cui, M. Médard, E. Yeh, D. Leith, F. Lai, and K. R. Duffy. 2017. A Linear Network Code Construction for General Integer Connections Based on the Constraint Satisfaction Problem. *IEEE/ACM Transactions on Networking* 25, 6 (Dec 2017), 3441–3454. https://doi.org/10.1109/TNET.2017.2746755
[9] E. Erez, M. Kim, Y. Xu, E. M. Yeh, and M. Médard. 2014. Deterministic Network Model Revisited: An Algebraic Network Coding Approach. *IEEE Transactions on Information Theory* 60, 8 (Aug 2014), 4867–4879. https://doi.org/10.1109/TIT.2014.2329840
[10] F. Gabriel, S. Wunderlich, S. Pandi, F. H. P. Fitzek, and M. Reisslein. 2018. Caterpillar RLNC With Feedback (CRLNC-FB): Reducing Delay in Selective Repeat ARQ Through Coding. *IEEE Access* 6 (2018), 44787–44802. https://doi.org/10.1109/ACCESS.2018.2865137
[11] J. Heide, S. Shi, K. Fouli, M. Médard, and V. Chook. [n. d.]. Random Linear Network Coding (RLNC): Background and Practical Considerations, https://datatracker.ietf.org/doc/draft-heide-nwcrg-rlnc-background/.
[12] J. Heide, S. Shi, K. Fouli, M. Médard, and V. Chook. [n. d.]. Random Linear Network Coding (RLNC)-Based Symbol Representation, https://tools.ietf.org/html/draft-heide-nwcrg-rlnc-02.
[13] C. Hellge and M. Medard. 2016. Multi-code Distributed Storage. In *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. 839–842. https://doi.org/10.1109/CLOUD.2016.0119
[14] T. Ho, M. Médard, R. Koetter, D.R. Karger, M. Effros, Jun Shi, and B. Leong. 2006. A Random Linear Network Coding Approach to Multicast. *IEEE Transactions on Information Theory* 52, 10 (Oct. 2006), 4413 –4430.
[15] M. Karzand, D. J. Leith, J. Cloud, and M. Médard. 2017. Design of FEC for Low Delay in 5G. *IEEE Journal on Selected Areas in Communications* 35, 8 (Aug 2017), 1783–1793. https://doi.org/10.1109/JSAC.2017.2710958
[16] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, and Jon Crowcroft. 2006. XORs in the Air: Practical Wireless Network Coding. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '06)*. ACM, New York, NY, USA, 243–254. https://doi.org/10.1145/1159913.1159942
[17] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. 2008. XORs in the Air: Practical Wireless Network Coding. *IEEE/ACM Transactions on Networking* 16, 3 (June 2008), 497–510. https://doi.org/10.1109/TNET.2008.923722
[18] R. Koetter and M. Medard. 2001. An algebraic approach to network coding. In *Proceedings. 2001 IEEE International Symposium on Information Theory (IEEE Cat. No.01CH37252)*. 104–. https://doi.org/10.1109/ISIT.2001.935967
[19] R. Koetter and M. Medard. 2002. Beyond routing: an algebraic approach to network coding. In *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 1. 122–130 vol.1. https://doi.org/10.1109/INFCOM.2002.1019253
[20] R. Koetter and M. Médard. 2003. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking* 11, 5 (Oct. 2003), 782 – 795.
[21] Desmond Lun, Niranjan Ratnakar, Muriel Médard, Ralf Koetter, David R. Karger, Tracey H, Ebad Amhed, and Fang Zhao. 2006. Minimum-cost multicast over coded packet networks. 52, 6 (June 2006), 2608–2623.
[22] D. S. Lun, M. Médard, T. Ho, and R. Koetter. 2004. Network coding with a cost criterion. In *International Symposium on Information Theory and its Applications (ISITA)*.
[23] D. S Lun, P. Pakzad, C. Fragouli, M. Médard, and R. Koetter. 2006. An Analysis of Finite-memory Random Linear Coding on Packet Streams. In *2006 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*. IEEE, 1– 6. https://doi.org/10.1109/WIOPT.2006.1666502
[24] D. Malak, M. Médard, and E. M. Yeh. 2018. ARQ with Cumulative Feedback to Compensate for Burst Errors. In *2018 IEEE Global Communications Conference (GLOBECOM)*. 1–6. https://doi.org/10.1109/GLOCOM.2018.8648058
[25] D. Malak, M. Médard, and E. M. Yeh. 2019. Tiny Codes for Guaranteeable Delay. *IEEE Journal on Selected Areas in Communications* 37, 4 (April 2019), 809–825. https://doi.org/10.1109/JSAC.2019.2898747
[26] D. Malak, A. Schneuwly, M. Médard, and E. Yeh. 2019. Delay-Aware Coding in Multi-Hop Line Networks. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*. 650–655. https://doi.org/10.1109/WF-IoT.2019.8767270
[27] M. Medard, F. H. P. Fitzek, M. Montpetit, and C. Rosenberg. 2014. Network coding mythbusting: why it is not about butterflies anymore. *IEEE Communications Magazine* 52, 7 (July 2014), 177–183. https://doi.org/10.1109/MCOM.2014.6852100
[28] J. Neu and M. Medard. 2019. Babel Storage: Uncoordinated Content Delivery from Multiple Coded Storage Systems. *IEEE Globecom* (2019).
[29] S. Pandi, S. Wunderlich, and F. H. P. Fitzek. 2018. Reliable low latency wireless mesh networks — From Myth to reality. In *2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC)*. 1–2. https://doi.org/10.1109/CCNC.2018.8319326
[30] Nikolaos Papanikos and Evangelos Papapetrou. 2014. Deterministic Broadcasting and Random Linear Network Coding in Mobile Ad Hoc Networks. *CoRR* abs/1411.4465 (2014). arXiv:1411.4465 http://arxiv.org/abs/1411.4465
[31] V. Roca. [n. d.]. The Sliding Window Random Linear Code (RLC) Forward Erasure Correction (FEC) Scheme for FECFRAME, https://tools.ietf.org/html/draft-roca-tsvwg-rlc-fec-scheme-01.
[32] C. W. Sørensen, D. E. Lucani, and M. Médard. 2017. On network coded filesystem shim: Over-the-top multipath multi-source made easy. In *2017 IEEE International Conference on Communications (ICC)*. 1–7. https://doi.org/10.1109/ICC.2017.7996675
[33] J.K. Sundararajan, D. Shah, M. Médard, S. Jakubczak, M. Mitzenmacher, and J. Barros. 2011. Network Coding Meets TCP: Theory and Implementation. *Proc. IEEE* 99, 3 (March 2011), 490–512. https://doi.org/10.1109/JPROC.2010.2093850
[34] J.K. Sundararajan, D. Shah, M. Médard, M. Mitzenmacher, and J. Barros. 2009. Network Coding Meets TCP. In *INFOCOM 2009, IEEE*. 280–288. https://doi.org/10.1109/INFCOM.2009.5061931
[35] J. K Sundararajan, P. Sadeghi, and M. Médard. 2009. A feedback-based adaptive broadcast coding scheme for reducing in-order delivery delay. In *Workshop on Network Coding, Theory, and Applications, 2009. NetCod '09*. IEEE, 1–6. https://doi.org/10.1109/NETCOD.2009.5437470
[36] M. Talas, R. Rausch, and N. Barhoum. 2018. New York City Pilot Acquisition and Installation Experiences. (2018).
[37] F. Tesema and V. Pauli. 2018. Layer 2 FEC in 5G Broadcast / Multicast Networks, http://nomor.de/wp-content/uploads/2018/09/Nomor_WhitePaper_2018_FT_VP_RAN_2nd_layer_FEC_v3.pdf. (2018).
[38] F. Tesema and V. Pauli. 2019. Coordinated Link Adaptation and Higher Layer Error Correction for 5G Broadcast / Multicast, http://nomor.de/wp-content/uploads/2019/07/white_paper_coord_la_and_ec.pdf. (2019).
[39] S. Wunderlich, F. Gabriel, S. Pandi, and F. H. P. Fitzek. 2017. We don't need no generation - a practical approach to sliding window RLNC. In *2017 Wireless Days*. 218–223. https://doi.org/10.1109/WD.2017.7918148
[40] S. Wunderlich, F. Gabriel, S. Pandi, F. H. P. Fitzek, and M. Reisslein. 2017. Caterpillar RLNC (CRLNC): A Practical Finite Sliding Window RLNC Approach. *IEEE Access* 5 (2017), 20183–20197. https://doi.org/10.1109/ACCESS.2017.2757241
[41] Weifei Zeng, C.T.K. Ng, and M. Médard. 2012. Joint coding and scheduling optimization in wireless systems with varying delay sensitivities. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*. 416–424. https://doi.org/10.1109/SECON.2012.6275806
[42] F. Zhao and M. Médard. 2010. On analyzing and improving COPE performance. In *2010 Information Theory and Applications Workshop (ITA)*. 1–6. https://doi.org/10.1109/ITA.2010.5454107
[43] F. Zhao, M. Médard, M. Hundebøll, J. Ledet-Pedersen, S. A. Rein, and F. H. P. Fitzek. 2012. Comparison of analytical and measured performance results on network coding in IEEE 802.11 ad-hoc networks. In *2012 International Symposium on Network Coding (NetCod)*. 43–48. https://doi.org/10.1109/NETCOD.2012.6261882