

# Wireless Network Coding: Opportunities & Challenges

Christina Fragouli  
EPFL

Dina Katabi  
MIT

Athina Markopoulou  
UC Irvine

Muriel Médard  
MIT

Hariharan Rahul  
MIT

**Abstract**—Wireless networks suffer from a variety of unique problems such as low throughput, dead spots, and inadequate support for mobility. However, their characteristics such as the broadcast nature of the medium, spatial diversity, and significant data redundancy, provide opportunities for new design principles to address these problems. There has been recent interest in employing network coding in wireless networks. This paper explores the case for network coding as a unifying design paradigm for wireless networks, by describing how it addresses issues of throughput, reliability, mobility, and management. We also discuss the practical challenges facing the integration of such a design into the network stack.

## I. INTRODUCTION

Wireless networks have been designed using the wired network as the blueprint. The design abstracts the wireless channel as a point-to-point link, and grafts wired network protocols onto the wireless environments. For example, routing uses shortest path protocols, routers forward packets but do not modify the data, and reliability relies on retransmissions. The design has worked well for wired networks, but less so for the unreliable and unpredictable wireless medium.

The wireless medium is fundamentally different. While wired networks have reliable and predictable links, wireless links have high bit error rate, and their characteristics could vary over short time-scales. Further, wired links are unicast links, but the majority of wireless links (with omni-directional antennas) are broadcast links. Transmissions in a wired network do not interfere with each other, whereas interference is a common case for the wireless medium. Wired nodes are usually static, while wireless was built to support mobility and portability. The wired network design conflicts with the characteristics of the wireless medium. As a result, current wireless networks suffer low throughput, dead spots, and inadequate mobility support.

The characteristics of wireless networks might all seem disadvantageous at first sight, but a newer perspective reveals that some of them can be used to our advantage, albeit with a fresh design. The broadcast nature of wireless provides an opportunity to deal with unreliability; when a node broadcasts a packet, it is likely that at least one nearby node receives it, which can then function as the next-hop and forward the packet. This is in stark contrast to the present wireless design, where there is a single designated next-hop, and when it does not receive the packet, the previous hop has to retransmit it. The property is called spatial diversity and has been explored in the literature [2, 30, 9].

Interestingly, wireless networks exhibit significant data redundancy, i.e., there is a large overlap in the information available

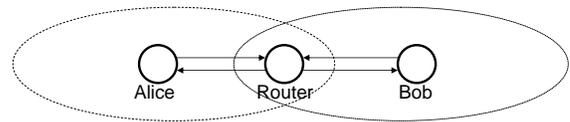


Fig. 1—A simple example of how network coding can improve throughput. It allows Alice and Bob to exchange a pair of packets through  $R$  using 3 transmissions instead of 4.

to the nodes. First, as a packet travels multiple hops, its content becomes known to many nodes. Further, wireless broadcast amplifies this redundancy because at each hop it delivers the same packet to multiple nodes within the transmitter’s radio range.

Can an alternative design of wireless networks exploit their intrinsic characteristics, such as spatial diversity and data redundancy, rather than foisting an artificial wireline abstraction? Network coding may be the answer.

For example, redundancy can be exploited to compress data, increasing the information flow per transmission, and thus improving the overall network throughput. Consider the example in Fig. 1, where Alice and Bob want to exchange a couple of packets. The radio range does not permit them to communicate directly and thus they need to go through a router. In the current design, Alice sends her packet to the router, which forwards it to Bob, and Bob sends his packet to the router, which forwards it to Alice. Yet, as noted in [22, 23], Alice and Bob could transmit their respective packets to the router, which XORs the two and broadcasts the resulting coded packet. Alice recovers Bob’s packet by XOR-ing again with her own, and Bob recovers Alice’s packet in the same way. The process exploits the existing redundancy in the network to compress the information, delivering two packets in a single transmission, and improving the throughput.

This paper explores the case for network coding as a design paradigm for wireless networks. We discuss how network coding can improve the throughput, reliability, fairness, and management of wireless networks. We build our argument using simple examples, some of which are taken from the literature, while others are developed herein. We also discuss the practical limitations of network coding and the challenges facing the integration of such design within a general network stack.

## II. BACKGROUND

More generally, in network coding, routers mix the content of different packets and broadcast the resulting coded packets on the wireless medium. Prior work shows that network coding achieves

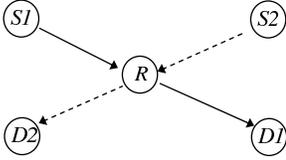


Fig. 2—“X” Topology: Two flows intersecting at R.

the multicast capacity of the network [1]. Two results have practical implications. First, the combination of [25, 24, 19] shows that, for multicast traffic, linear codes (i.e., having the routers create linear combinations of the packets they receive,  $\sum c_i p_i$ , where  $p_i$  is a packet and  $c_i$  is some coefficient) achieve the maximum capacity bounds, and coding and decoding can be done in polynomial time. Second, Ho et al. show that the above is true even when the routers pick random coefficients [16]. This enables distributed network coding, where routers do not need to consult each other on the choice of codes. Researchers have applied network coding to many areas including wireless networks [26, 22, 23], energy [33, 12], secrecy [3, 18], content distribution [5], and distributed storage [20]. Most work on network coding, however, is theoretical, assumes multicast traffic, and ignores traffic burstiness and application requirements [1, 25, 19, 24, 16, 17, 28, 7]. Recently, a few papers have employed network coding in wireless protocols to improve their throughput or reliability [23, 4, 21, 32]. These papers are more focused on implementable protocols and practical issues than the early theoretical foundation.

### III. OPPORTUNITIES

The following simple examples revolve around the same main theme, potential benefits of building future wireless networks around network coding.

#### A. Throughput

The throughput of today’s wireless networks leaves a lot to be desired; we outline here how a wireless architecture designed around network coding can help improve throughput. The intuition is that network coding increases wireless throughput because coding allows the routers to compress the transmitted information given what is known at various nodes (as in the Alice-Bob scenario in Fig. 1). By matching what each neighbor has with what another neighbor wants, a router can deliver multiple packets to different neighbors in a single transmission. This style of coding is called *inter-flow network coding* because the coding is done over packets that differ in their next hop, and thus from different flows.

1) *COPE*: Consider again the Alice-Bob scenario in Fig. 1. A recent protocol, COPE [23], extends the above idea beyond two nodes communicating on reverse paths. To do so, COPE exploits the shared nature of the wireless medium which, for free, broadcasts each packet in a small neighborhood around its path. This creates an environment conducive for coding because nodes in each area have a large and partially overlapping reservoir of packets they can use to decode. For example, Fig. 2 shows how two traffic flows traversing different paths can be encoded together.

Assume sources S1 and S2 are sending a stream of packets to destinations D1 and D2 respectively. Since wireless is a broadcast medium, by snooping, nodes D1 hears S2’s packets, and D2 hears S1’s packets traversing their neighborhood. Router R can exploit these overheard packets to XOR S1’s and S2’s packets and broadcast an XOR-ed version, which is useful to both destinations. The “X” topology is an extension of the Alice-Bob topology above. It is an important extension because in a real wireless network, there might be only a small number of flows traversing the reverse path of each other *à la* Alice-Bob, but one would expect many flows to intersect at a relay, and thus can be coded together.

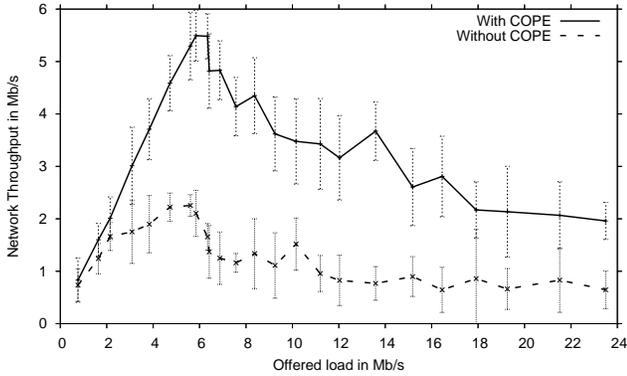
For more general topologies, COPE leads to larger bandwidth savings than are apparent from the above example. It can XOR more than a pair of packets and produce a multifold increase in the throughput. To summarize, COPE is a MAC extension that has two components:

- *Opportunistic Listening*: COPE exploits the shared nature of the wireless medium which, for free, broadcasts each packet in a small neighborhood around its path. Each node stores the packets it overhears for a limited period. It also tells its neighbors which packets it has heard by annotating the packets it sends. This creates an environment conducive to coding because nodes in each area have a large and partially overlapping reservoir of packets they can use for decoding.
- *Opportunistic Coding*: when a node transmits a packet, it uses its knowledge of what its neighbors have heard to deliver multiple packets in a single transmission. The node XORs multiple packets when each intended next hop has enough information to decode the encoded packet. More precisely, every node uses the following coding rule:

To transmit  $n$  packets,  $p_1, \dots, p_n$ , to  $n$  next hops,  $r_1, \dots, r_n$ , a node can XOR the  $n$  packets together only if each next hop  $r_i$  has all  $n - 1$  packets except the packet it wants.

Theoretically, COPE reduces the number of transmissions by a factor of two [23], and thus should double the throughput. Katti et al. [23] have shown, however, that in practice the throughput gain is much larger. Fig. 3 plots the aggregate throughput as a function of the traffic demands, for a 20-node mesh testbed. The figure shows that COPE brings a 3-4x increase in the throughput, when compared to current 802.11. Experimental throughput gain exceeds the theoretical gain because COPE alleviates hot-spots in the network. Specifically, in an ad hoc network, most paths cross at the center. As a result, nodes in the center of the network experience congestion, build queues, and drop packets. These dropped packets have consumed bandwidth to reach the center of the network. Dropping them in the center wastes network resources and significantly reduces the overall throughput. In contrast, with coding, congested nodes in the center of the network have the opportunity to send multiple packets in a single transmission, allowing them to drain their queues faster and avoid dropping packets.

2) *Content Distribution*: Many people today work in coffee shops, while listening to music. Imagine a scenario in which a music store like iTunes partners with a popular coffee store like Starbucks to provide music in hot-spots. A customer can listen



**Fig. 3—COPE, an inter-flow network coding protocol, provides a several-fold increase in the throughput of wireless Ad hoc networks. Results are for UDP flows with randomly picked source-destination pairs, Poisson arrivals, and heavy-tail size distribution.**

to any iTunes song over WiFi. iTunes and Starbucks may charge customers per song or may provide the ability to listen to personalized music as a value-added service. Efficient bandwidth usage will be essential to scale this service for many customers.

Consider the scenario where Alice and Bob are customers in a Starbucks. Each of them has some songs on her/his laptop, but is currently interested in listening to a song that she/he does not have. They contact the access point in Starbucks, which has access to iTunes music inventory. Each customer requests the access point for the song he/she wants to download, and also tells it the list of songs he/she already has. Say that Alice wants song  $s_A$  and Bob wants song  $s_B$ . Alice however happens to have  $s_B$  on her laptop, and Bob happens to have  $s_A$  on his laptop. Instead of sending separate streams to Alice and Bob, the access point can use network coding to compress the information and improve bandwidth utilization. Specifically, to deliver packet  $p_A$  to Alice and  $p_B$  to Bob, the access point XORs the two packets and broadcasts the XOR-ed packet. Since Alice has the song Bob is downloading, she can use the meta-data in the XOR-ed packet's header to locate the packet Bob is playing, XOR it with the received packet, and obtain the packet she wants. Bob can do the same to remove Alice's packet from the XOR-ed packet and obtain the packet he wants. In this example, network coding doubles the throughput.

Though the coding technique used in this section is similar to COPE (in the above section), it extends the work to a new application. Specifically, while COPE requires a multi-hop network, this application runs on a WLAN. Further, COPE exploits data redundancy at the network layer, i.e., it exploits the fact that the same packet is known to multiple nodes since it traveled along multiple hops. In contrast, here we exploit data redundancy at the application-level. The objective is to show that network coding is a versatile technique that can be customized to different applications.

## B. Reliability

The primary means of ensuring reliability in the present architecture is retransmission of lost packets. This works well in wired

networks where the bit error rate is very low, but is inefficient over error prone wireless channels.

This section presents a few canonical examples showing how network coding provides a more efficient approach to reliability. In contrast to §III-A, the examples in this section employ *intra-flow network coding*, i.e., routers mix packets heading to the same destination. As a result of this mixing, each received packet contains some information about all packets in the original file, and thus, no coded packet is special. Said differently, without coding, a transmitter needs to know which exact packets the destination (or the next hop) misses so that it can retransmit them. When the network is unreliable, communicating this feedback reliably consumes significant bandwidth. In the presence of coding, no specific packet is indispensable and as a result a transmitter does not need to learn which particular packet the destination misses, it only needs to get feedback from the destination once it has received enough packets to decode the whole file. The reader may have noticed that the above applies to erasure-correcting coding applied at the source too. Indeed, source coding is just a special case of intra-flow network coding, where the source is the only node allowed to mix the packets in the flow. There are however benefits to allowing other nodes to perform such mixing as seen from the examples below.

Finally, note that a natural synergy exists between reliability and throughput. Saying that network coding provides efficient reliability means that for the same level of reliability, it allows the network to achieve higher throughput.

1) *Single Path*: Consider the simple example in Fig. 4-(a), where the labels on the links refer to their delivery rate. The source has a perfect link to the router  $R$ . In contrast,  $R$ 's link to the destination has a delivery probability of 50% in both directions. Say we want to transfer a  $n$ -packet file. The transmission from the source to node  $R$  is perfect, so let us focus on the cost of delivering these  $n$  packets from node  $R$  to the destination.

802.11 unicast in today's wireless networks will require each packet to be retransmitted until the ack is successfully received at the sender, for an expected number of 4 transmissions per packet, and  $4n$  packets for the whole file, which is far greater than the lower bound of  $2n + 2$  transmissions (for the  $n$  packets and 1 acknowledgment) for the transfer. Ideally, one would want a scheme that achieves this performance with low complexity. A possible non-coding scheme could proceed in rounds, where the sender first transmits all  $n$  packets, waits for a batched acknowledgment from the receiver detailing which packets it has received, and then proceeds similarly with the packets that were not successfully received. This scheme requires  $2n + 2 \log_2 n$ , which, while far from ideal, is a significant improvement over the current scenario. Unfortunately, this improvement is achieved at the cost of increased protocol complexity, as well as larger sized acks from the receiver.

Network coding, however, provides the ideal solution in this case, with very low complexity. Consider an approach where  $R$  transmits random linear combination of the packets. In particular,  $R$  transmits to the destination packets of the form  $p'_i = c_{i1}p_1 + \dots + c_{in}p_n$ , where  $p_j$  is the  $j^{\text{th}}$  packet in the file, and  $c_{ij}$ s are random coefficients that  $R$  picks. The destination just waits to receive  $n$

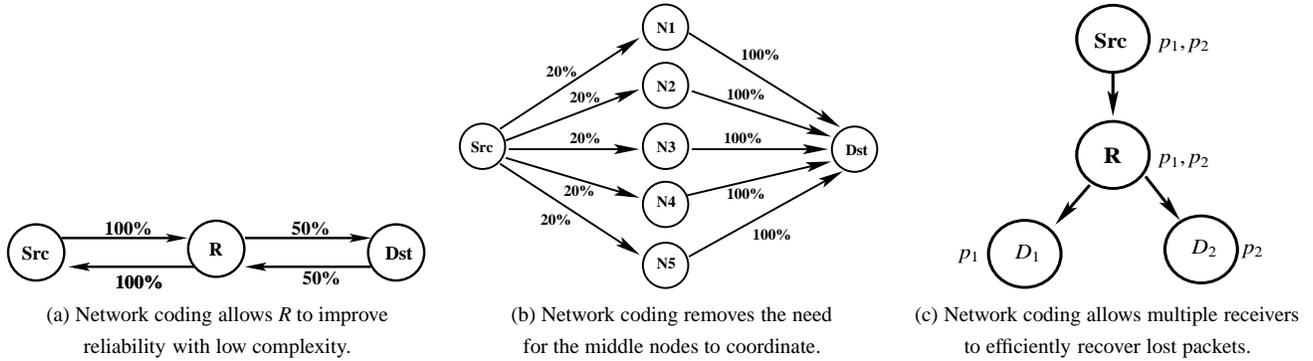


Fig. 4—Example scenarios where network coding provides reliability benefits

such coded packets and recovers the original packets in the file, as follows:<sup>1</sup>

$$\begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} = \begin{pmatrix} c_{11} & \dots & c_{1n} \\ \vdots & \ddots & \vdots \\ c_{n1} & \dots & c_{nn} \end{pmatrix}^{-1} \begin{pmatrix} p'_1 \\ \vdots \\ p'_n \end{pmatrix} \quad (1)$$

Once the destination has decoded the packets, it immediately sends  $R$  one acknowledgement for the whole file. Since the ack on average needs two trials, and assuming  $R$  continues transmitting during that period,  $R$  will deliver the file to the destination in  $2n + 2$  transmissions, on average.

2) *Dead Spots*: Network coding also helps in dealing with dead spots. Consider the scenario in Fig. 4-(b), where Alice would like to transfer a file to Bob. Bob is not within Alice's radio range, and thus Alice needs the help of an intermediate node. There are 5 nearby wireless nodes which could relay Alice's packets to Bob. Unfortunately, Alice is in a dead spot with 80% loss rate to every nearby wireless node. In today's 802.11 networks, Alice will pick the best path to Bob for her transfer. But since all paths are lossy, each packet will have to be transmitted 6 times (5 times from Alice to the relay and once from the relay to Bob.)

A better approach would make use of spatial diversity to improve Alice's throughput. Alice broadcasts her packets, and any relay that hears a packet can forward it to Bob. In this case, the probability of delivering a packet to the relay increases from 20% to  $(1 - 0.8^5) \times 100 \approx 67.2\%$ . Thus, on average a packet is transmitted 2.5 times (1.5 times from Alice to the relays and once from the receiving relay to Bob.) This increases Alice's throughput by 2.4x. But using diversity without coding can create a different problem. Multiple relay nodes can hear the same packet and thus attempt to transmit it to Bob. This creates spurious transmissions and wastes the wireless bandwidth.

The combination of spatial diversity and network coding solves the problem. Alice broadcasts her packets. Any relay can participate in forwarding Alice's packets to Bob. To do so, the relay creates random linear combinations of the packets it has received from Alice so far. Specifically, a relay transmits to Bob packets

<sup>1</sup>The use of random coefficients ensures that the generated packets  $p'_i$  are independent with exponentially high probability [16].

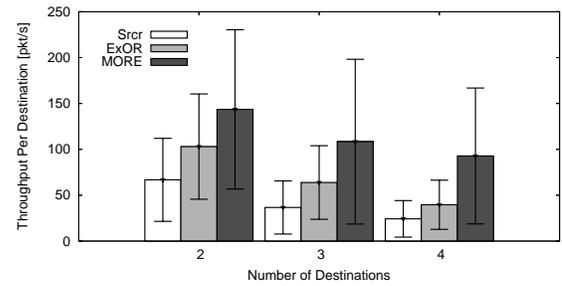


Fig. 5—MORE, an intra-flow network coding protocol, improves multicast throughput. The figure shows the per-destination multicast throughput of MORE, ExOR, and Srcr. The thick bars show the average per-destination throughput taken over 40 runs with different nodes. The thin lines show the standard deviation.

of the form  $p'_i = \sum c_{ij}p_j$ , where  $p_j$  is one of Alice's packets, and  $c_{ij}$ 's are random coefficients. If the file contains  $n$  packets, then Bob can decode the whole file after receiving any  $n$  coded packets, using the matrix inversion in 1. Once Bob decodes the file, he immediately broadcasts an acknowledgment for the whole transfer, causing the relays to stop their transmissions.

3) *Multicast*: Consider the example in Fig. 4-(c), where the source wants to multicast a video stream to nodes  $D_1$  and  $D_2$ . Say that the source transmits packets  $p_1$  and  $p_2$  to the router  $R$ , which broadcasts the two packets to the destinations. Since wireless receptions at different nodes are highly independent [31, 30], it is possible that when  $R$  broadcasts the two packets, node  $D_1$  receives only  $p_1$ , while  $D_2$  receives only  $p_2$ . In this case, node  $R$  has to retransmit both packets to allow both destinations to recover their corresponding losses. But, if node  $R$  is allowed to code, then it can XOR the two packets (i.e.,  $p_1 \oplus p_2$ ), and broadcast the XOR-ed version on the wireless medium. This single transmission allows both destinations to recover their corresponding losses, providing efficient reliability.

Ghaderi et al. [14] have studied this problem theoretically and shown that network coding improves the expected number of transmissions by a factor  $\Theta(\frac{\log K}{\log \log K})$ , where  $K$  is the fan-out of the multicast tree.

We have developed MORE, an opportunistic routing protocol that exploits intra-flow network coding [4]. We experimentally compare MORE with two multicast protocols. The first protocol,

Round Robin Routing						
time	1	2	3	4	5	6
D tx	$x_A^1$	$x_B^1$	$x_C^1$	$x_A^1$	$x_B^2$	$x_C^2$
A rx	-	-	$x_C^1$	-	$x_B^2$	$x_C^2$
B rx	$x_A^1$	-	$x_C^1$	-	$x_B^2$	$x_C^2$
C rx	-	$x_B^1$	$x_C^1$	$x_A^1$	-	$x_C^2$
Network Coding Solution						
time	1	2	3	4	5	6
D tx	$x_A^1$	$x_B^1$	$x_C^1$	$x_A^1 + x_B^1 + x_C^1$	$x_A^1 + 2x_B^1 + 3x_C^1$	$x_A^1 + 4x_B^1 + 5x_C^1$
A rx	-	-	$x_C^1$	-	$x_A^1 + 2x_B^1 + 3x_C^1$	$x_A^1 + 4x_B^1 + 5x_C^1$
B rx	$x_A^1$	-	$x_C^1$	-	$x_A^1 + 2x_B^1 + 3x_C^1$	$x_A^1 + 4x_B^1 + 5x_C^1$
C rx	-	$x_B^1$	$x_C^1$	$x_A^1 + x_B^1 + x_C^1$	-	$x_A^1 + 4x_B^1 + 5x_C^1$

TABLE I—Basestation  $D$  broadcasts information destined to nodes  $A$ ,  $B$  and  $C$  iid erasure channels with probability of erasure 0.5. The columns show the packets transmitted by  $D$ , and received by  $A$ ,  $B$ , and  $C$ . The network coding solution leads to the same aggregate throughput, but more evenly divided among the receivers.

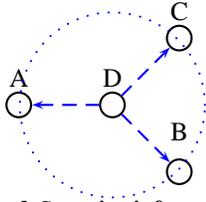


Fig. 6—Nodes  $A$ ,  $B$  and  $C$  receive information from node  $D$ .

referred to as Srcr, distributes the traffic to the multicast destinations along the edges of a multicast tree rooted at the source. The second protocol, called ExOR, exploits the broadcast nature of the medium to deliver a packet to multiple nodes simultaneously. Fig. 5 plots the average per-destination throughput in a 20-node wireless testbed. In this scenario, the source multicasts a file to a varying number of destinations. As expected, the per-destination average throughput decreases with increased number of destinations. Interestingly, however, the figure shows that MORE's throughput gain increases with increased number of destinations. Intra-flow network coding, and specifically MORE, has 35-200% throughput gain over ExOR and 100-300% gain over Srcr.

### C. Fairness

An inherent characteristic of wireless networks is the time variability of the received signal quality due to interference and attenuation effects (fading). The random fluctuations at the physical layer are perceived as packet erasures at higher layers, and may result in variability of the reception rates over short time periods. Real-time applications, on the other hand, tend not to tolerate variable reception rates very well, and there is a significant research body in dealing with such effects. Moreover, such variability may also require an involved scheduling scheme to ensure fairness over short time periods. Finally, reducing the variability of packet delivery may also serve to alleviate the problem of window closing, for instance in TCP.

We argue here that, combining network coding and broadcast over such an environment allows to naturally “smooth over” the rates that the receivers experience over short time periods and alleviate these problems. We illustrate this point through the following simple example.

Basestation  $D$  in Fig. 6 has three independent streams of packets  $x_A = \{x_A^i\}$ ,  $x_B = \{x_B^i\}$ , and  $x_C = \{x_C^i\}$  to transmit to nodes  $A$ ,  $B$  and  $C$  respectively, that are within its broadcast radius. In today's wireless systems independent information streams are transmitted in orthogonal dimensions, using a time, frequency or code division multiple access scheme.

For example, assume that the base-station uses timesharing to sequentially broadcast information to each of the receivers. A broadcast transmission successfully reaches each receiver according to independent Bernoulli distributions with probability  $p$ . The basestation has no a priori channel information, but the receivers can acknowledge successful reception.

Table I shows a possible scenario. Even with a seemingly fair transmission strategy such as round-robin, node  $A$  has received no packet from the stream it is interested in after six time slots, node  $B$  receives one packet and node  $C$  receives two packets. This is because of the interaction of the transmission schedule with random channel fluctuations. For example, note that during the first timeslot, the basestation broadcasts a packet destined to node  $A$ : although node  $A$  does not receive it, node  $B$  does receive it but has no use for it.

The network coding solution<sup>2</sup>, also included in the same table, capitalizes on such situations: the base station transmits linear combinations such that each transmission offers some benefit to all nodes that receive it.

Note that network coding in this case does not change the throughput of the system, as we are effectively requiring that all receivers receive all information (so, thrice the rate), and tripling channel usage. The benefit we get is that individual users experience a rate that converges towards the average value *i.e.* one packet every six time-slots, and that this rate is achieved without a complicated scheduling strategy involving knowledge of the channel.

More formally, let  $T$  be a random variable that denotes the total number of transmissions the base-station will need to convey the  $L$  packets to node say  $A$ , where packets get dropped independently with probability  $p$ . From basic probability theory, the base-station will need  $E(T) = L/(1 - p)$  transmission attempts on average before success. As  $L$  grows, the observed empirical distribution will converge to the actual distribution. That is, if  $L_0$  denotes the number of erasures after  $L$  transmissions,

$$Pr\left\{\left|\frac{L_0}{L} - p\right| > \epsilon\right\} \rightarrow 0$$

exponentially fast. If  $k$  receivers share the transmission medium, each one of them will observe erasures over  $kL$  instead of  $L$  timeslots, and hence the distribution converges exponentially (in  $k$ ) faster to the average than with the non-coding solution.

### D. Mobility

In mobile environments, where the network configuration changes quickly, routing updates are costly. Network coding can

<sup>2</sup>The network coding solution can also be thought of as a form of superposition coding for a broadcast channel.

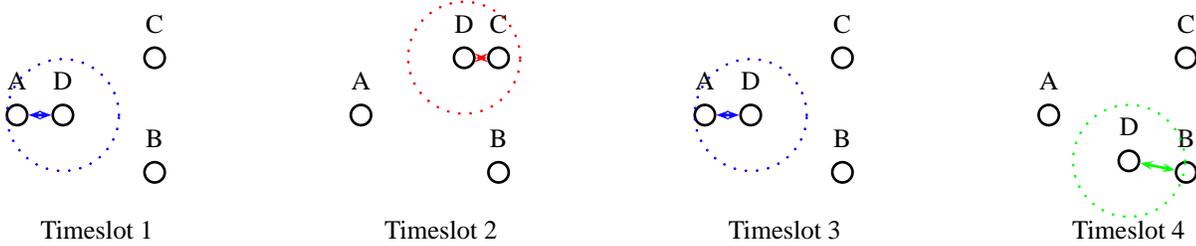


Fig. 7—Node  $D$  broadcasts information without knowing which nodes are within its range.

address the uncertainty in the topology and alleviate the need for exchanging routing updates.

Continuing the example in Fig. 6, assume now that all nodes are interested in receiving the information transmitted by node  $D$ . Node  $D$  however is mobile and does not know the identity of nodes within its transmission range. A possible random mobility pattern is depicted in Fig. 7. Without coding, when  $D$  comes in range with a node  $A$ , it needs to detect the existence of this new neighbor through some packet exchange. It also need to learn which packets  $A$  has already received during its prior interactions with  $D$ . Without this information  $D$  cannot ensure the transmission of new information to  $A$ . With network coding however,  $D$  can keep transmitting new random linear combinations, say  $\sum c_i p_i$ , where  $c_i$  is a randomly chosen coefficient. As long as any of the mixed packets, i.e., the  $p_i$ 's, has information unknown to  $A$ , the coded packet will be innovative and useful. Furthermore, the coded packet can bring new information to many of  $D$ 's neighbors at once because any neighbor who has missed any of the  $p_i$ 's in the linear sum will find the packet innovative. Thus in a network coding approach,  $D$  need not know who is in its radio range or which packets they have heard, removing the need for tracking the quickly changing topology.

The previous simple example can be generalized as follows. Consider a wireless ad-hoc network with  $n$  nodes, where each node is a source that wants to transmit information to all other nodes. Assume a uniform random mobility pattern, and that each node can successfully broadcast information within a radius of  $O(1/\sqrt{n})$ .

Consider two scenarios:

*Perfect-knowledge:* Each node, before transmitting, knows who its neighbors are, and what information they have already successfully received.

*No-knowledge:* Nodes have no information of their neighbors identity or past history. Then, as discussed in [13],

- Network coding, under the “no-knowledge” assumption achieves the same performance as routing under the “perfect-knowledge” assumption. Thus routing information updates are not needed.
- Under the “no-knowledge” assumption for both network coding and routing, network coding reduces by a factor of  $\log n$  the number of required transmissions. In this case we get benefits in terms of number of transmissions that is proportional to energy efficiency. The proof of this claim is by reducing the problem to the coupon collector’s problem over a complete graph, for

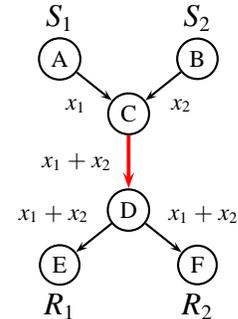


Fig. 8—Example of monitoring a network using probes at the edge and network-coding in the middle. Nodes  $A$  and  $B$  act as sources;  $E$  and  $F$  act as receivers;  $C$  XORs the incoming packets;  $D$  broadcasts the incoming packet to both  $E$ ,  $F$  downstream.

which Deb and Medard showed that network coding offers a factor of  $\log n$  [8].

Simulation results in [13] support the analysis and show the applicability of the results to more realistic mobility patterns.

### E. Monitoring

The highly stochastic nature of wireless environments makes it desirable to monitor the state of wireless links. In this section, we show how network coding can be exploited to better monitor the link loss rates in a multi-hop wireless network. As a concrete example, let us consider the simple sensor network shown in Fig. 8 and try to estimate the loss rate on all links using only end-to-end measurements. This approach is often used in wireless sensor networks [29] to exploit the data aggregation paradigm: data collecting nodes can infer loss rates on internal links by exploiting whether packets from various sensors are received, without the need to actively inject probes.

In the example of Fig. 8, let nodes  $A$ ,  $B$ ,  $C$ , and  $D$  be sensors, and nodes  $E$  and  $F$  be sinks that can easily communicate with each other (e.g., connected through a high-bandwidth link). Let nodes  $A$  and  $B$  send probes  $x_1$  and  $x_2$  respectively in one time slot; let  $C$  XOR the incoming packets and forward  $x_1 + x_2$  downstream to  $D$ ; let  $D$  broadcast  $x_1 + x_2$  to both  $E$  and  $F$ . Each of the nodes  $E$  and  $F$  may receive  $x_1$ ,  $x_2$ ,  $x_1 + x_2$ , or nothing, depending on the success/failure of the transmission on each link. This is one round of probes and can be repeated several times. By sending several rounds of probes from  $A$  and  $B$  and by observing the pairs received by  $E$  and  $F$  at each round, we can infer the link loss rates on all five links. This approach is essentially link loss tomography enhanced with network coding operations [10, 11]. Alternatively, one could

use either (i) traditional tomography with only multicast or unicast probes or (ii) per-link monitoring.

It turns out that the network coding approach has several advantages. Compared to unicast and multicast tomographic techniques [15, 6], it is able to identify a larger number of links. E.g., in Fig. 8, we are able to identify the loss rate on all five links; in contrast, a multicast probe from  $A$  to  $E$  and  $F$  would not be able to distinguish links  $A - C$  and  $C - D$  that are connected in tandem; similarly, a unicast probe from  $A$  to  $F$  would not be able to distinguish links  $A - C$ ,  $C - D$ ,  $D - F$  on its path. Another advantage is that we can improve the trade-off between estimation accuracy and number of probes, for two reasons. First, with network coding, each round of probes “observes” more paths; e.g., in Fig. 8, we observe all four paths ( $A - E$ ,  $A - F$ ,  $B - E$ ,  $B - F$ ) instead of just two (with multicast probes) or one (with unicast probes). Second, network coding combines probes flowing on the same link into one, thus resulting in exactly one probe per link. In contrast, to cover all five links with multicast probes, we would have to use two multicast trees rooted at  $A$  and  $B$ , which would unnecessarily overlap on the links downstream from  $C$ , namely  $C - D$ ,  $D - E$ ,  $D - F$ . Finally, compared to per-link monitoring, tomographic techniques allow to keep internal nodes simple and delegate complexity to special nodes at the edge. In addition, they “scan” the entire network in one pass, while per-link monitoring has to send updates to a processing node.

The reduction in the number of probes, achieved by the network coding approach, translates into bandwidth and energy savings, which are scarce resources in wireless networks [34]. These savings become even more important in dynamic scenarios (e.g., due to mobility or due to temporal changes of the link characteristics) and when up-to-date information about the state of the network is needed.

#### IV. NEW CHALLENGES

A new network architecture that employs network coding and exploits the broadcast nature of the wireless medium would necessarily require the research community to rethink the network stack. Our medium access control, routing, and transport protocols are all imported from the wired domain, with minor tweaks. They are optimized to work over point-to-point links, assume a single predetermined path and a layered architecture. The cost of redesigning our network stack is non-negligible. But the wireless throughput is intrinsically limited, which warrants efforts to investigate the potential of new high-throughput architectures. Furthermore, the wireless environment is much more amenable to new deployments than the wired environment; usually such deployments can rely solely on software updates.

##### A. Challenges of a Broadcast Network

The benefits of network coding utilize the broadcast nature of the wireless medium to simultaneously transmit a single packet to multiple receivers. Most of the new challenges are not caused by network coding, but are rather a side product of relying on the broadcast channel, which has implications on MAC, routing, and transport protocols.

1) *MAC*: The standard access mode of 802.11 and similar MACs is a Distributed Coordination Function (DCF) combining carrier sense multiple-access (CSMA) with collision avoidance (CA). A node that wishes to transmit senses the channel and begins sending only if it detects that the channel is clear for a pre-defined interval. It then waits for an acknowledgment from the receiver. If the sender does not receive an acknowledgement within a specific period, it assumes that there was a collision and selects a random backoff timer uniformly distributed within a contention window. The contention window doubles for every failed transmission in order to reduce the probability of collisions. In contrast, there is no acknowledgment for broadcast packets. Further, it is unclear how to add this functionality without creating significant complexity and the potential for ack implosion.

The lack of acknowledgments for broadcast packets implies that the MAC does not have any congestion avoidance function in such a mode. Hence, the use of an unmodified 802.11 style MAC leads to an increased number of collisions, and a drop in the throughput of the system. This is exacerbated by the fact that many 802.11 cards seem to have a flawed implementation of carrier sense (CSMA).<sup>3</sup> Additionally, the transmitter is not able to determine successful reception in the absence of broadcast. Hence, the MAC does not retransmit broadcast packets, and does not provide any reliability benefits. The lack of MAC reliability is not an issue for intra-flow network coding. But inter-flow network coding needs to compensate for it using other mechanisms [23].

2) *Routing*: Traditional routing protocols impose a point-to-point abstraction on wireless networks, and reduce routing in a wireless network to a shortest path computation on these directed links, *a la* wired networks. However, with broadcast, multiple nodes could simultaneously receive a packet and one or more of them might choose to transmit as a result. This changes the notion of routing from a single shortest path to a multipath problem, where decisions are made after packet reception, not at the time of transmission. Multipath routing can be formulated as a linear program (LP). The difficulty, however, arises from the broadcast property of the wireless links, which gives the LP formulation an exponential number of constraints [27]. The MORE protocol presents a practical low-complexity heuristic that addresses this issue [4]. We believe MORE presents a significant first step towards a general broadcast-based routing protocol. Further advances would integrate COPE [?] with MORE, enhance MORE’s round-robin service of multi-flows, and rethink auto-rate.

3) *Transport*: As described in §IV-A.1, the 802.11 MAC used in broadcast mode does not perform the usual link layer functionality of contention resolution and retransmission of lost packets. For inter-flow coding, the resulting high loss rate needs to be addressed; otherwise it would be mistaken as a sign of congestion by transport protocols like TCP, causing them to reduce their sending rate unnecessarily. For intra-flow coding, reliability comes for free, but a different problem surfaces. Coding and decoding involve linear operations over batches of packets. Batching does

<sup>3</sup>Our experience with 802.11 Atheros cards shows that a sender does not detect an ongoing transmission that started before its own ended.

not work well with window-based transport protocols like TCP. Hence, transport protocols would need to be redesigned to be effective in the presence of these changed forwarding semantics.

### B. Coding Challenges

The coding challenges arise from the desire to combine several attractive properties, such as low complexity, delay and memory requirements, high achievable rates, and adaptability to unknown channel conditions. In general, there is a trade-off between these properties.

1) *Fast Coding*: Network coding requires intermediate nodes in the network to perform operations over finite fields in real-time. While the cost of inter-flow coding is usually low, the general linear codes used in intra-flow coding could be computationally expensive. In this case, the encoding algorithms require polynomial time complexity that is bounded as  $O(n^2)$ , where  $n$  is the number of the linearly combined packets. Nodes may need to examine whether received packets are innovative (bring new information), in which case an additional complexity of  $O(n^2)$  operations is required. Decoding amounts to solving a system of linear equations, that typically requires  $O(n^3)$  operations [5]. There is an increased effort to design lower complexity encoding and decoding algorithms inspired by low density codes, but this effort is still at its first steps.

Coding however is not a deployment hurdle in current wireless networks. Chachulski et al. [4] report that MORE, a network coding protocol, can sustain a throughput of 44 Mb/s on low-end machines with Celeron 800MHz CPU and 128KB of cache.

2) *Forced Reliability*: Intra-flow network coding does not naturally lend itself to graceful performance degradation: receiving only  $n - 1$  linear combinations of  $n$  linearly combined packets is practically useless. This imposes stringent requirements for reliable delivery, and is in stark contrast to uncoded transmission.

3) *Realtime Traffic*: Intra-flow coding across  $n$  packets implies that a receiver needs in general to collect all  $n$  of them before extracting the data. In real-time applications, such as audio and video, the associated delay might be prohibitive for large values of  $n$ , which seems to indicate that small values of  $n$  would be desirable. Using small values of  $n$  on the other hand may not allow us to optimally mix the information and realize the theoretically promised benefits. Thus, there is a tension in balancing these two opposing requirements.

## V. CONCLUSION

Network coding enables more efficient, scalable, and reliable wireless network. These opportunities come with a need for rethinking our MAC, routing, and transport protocols. We believe, however, that future research will overcome these challenges and integrate network coding into the wireless network design.

## REFERENCES

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung. Network Information Flow. In *IEEE Transactions on Information Theory*, July 2000.

[2] S. Biswas and R. Morris. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM*, 2005.

[3] N. Cai and R. W. Yeung. Secure Network Coding. In *ISIT*, 2002.

[4] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. In *ACM SIGCOMM*, 2007.

[5] Christos Gkantsidis and Pablo Rodriguez. Network Coding for Large Scale Content Distribution. In *IEEE INFOCOM*, 2005.

[6] M. Coates, A. H. III, R. Nowak, and B. Yu. Internet tomography, 2002.

[7] S. Deb, M. Effros, T. Ho, D. R. Karger, R. Koetter, D. S. Lun, M. Médard, and N. Ratnakar. Network coding for wireless applications: A brief tutorial. In *IWWAN*, 2005.

[8] S. Deb, M. Médard, and C. Choute. Algebraic gossip: A network coding approach to optimal multiple rumor mongering. In *IEEE Transactions on Information Theory*, 2006.

[9] S. N. Diggavi, N. Al-Dahir, A. Stamoulis, and A. R. Calderbank. Great expectations: The value of spatial diversity to wireless networks, 2004.

[10] C. Fragouli and A. Markopoulou. A network coding approach to network monitoring. In *Allerton*, 2005.

[11] C. Fragouli, A. Markopoulou, and S. Diggavi. Topology inference using network coding. In *Allerton*, 2006.

[12] C. Fragouli, J. Widmer, and J. LeBoudec. Efficient broadcasting using network coding. *IEEE/ACM Transactions on Networking*, 2007.

[13] C. Fragouli, J. Widmer, and J. LeBoudec. On the benefits of network coding for wireless applications. In *NetCod 2006*.

[14] M. Ghaderi, D. Towsley, and J. Kurose. Reliability benefit of network coding. In *University of Massachusetts Amherst Technical Report 07-08*, 2007.

[15] G. Hartl and B. Li. Loss inference in wireless sensor networks based on data aggregation, 2004.

[16] T. Ho, R. Koetter, and M. Médard.

[17] T. Ho, B. Leong, M. Médard, R. Koetter, Y. Chang, and M. Effros. The Utility of Network Coding in Dynamic Environments. In *IWWAN*, 2004.

[18] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros. Resilient network coding in the presence of byzantine adversaries. *IEEE Transactions on Information Theory*, 2006.

[19] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 2003.

[20] A. Jiang. Network Coding for Joing Storage and Transmission with Minimum Cost. In *ISIT*, 2006.

[21] S. Katti, S. Gollakota, and D. Katabi. Embracing wireless interference: Analog network coding. In *ACM SIGCOMM*, 2007.

[22] S. Katti, D. Katabi, W. Hu, H. S. Rahul, and M. Médard. The importance of being opportunistic: Practical network coding for wireless environments, 2005.

[23] S. Katti, H. Rahul, D. Katabi, W. Hu, M. Médard, and J. Crowcroft. XORs in the Air: Practical Wireless Network Coding. In *ACM SIGCOMM*, 2006.

[24] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 2003.

[25] S. R. Li, R. W. Yeung, and N. Cai. Linear network coding. In *IEEE Transactions on Information Theory*, 2003.

[26] D. S. Lun, M. Médard, and R. Koetter. Efficient operation of wireless packet networks using network coding. In *International Workshop on Convergent Technologies (IWCT)*, 2005.

[27] D. S. Lun, M. Médard, and R. Koetter. Network coding for efficient efficient wireless unicast. In *International Zurich Seminar on Communications (IZS 2006)*, 2006.

[28] D. S. Lun, M. Médard, R. Koetter, and M. Effros. Further results on coding for reliable communication over packet networks. In *IEEE International Symposium on Information Theory (ISIT 05)*, 2005.

[29] Y. Mao, F. R. Kschischang, B. Li, and S. Pasupathy. A factor graph approach to link loss monitoring in wireless sensor networks. *IEEE SAC*, 23.

[30] A. K. Miu, H. Balakrishnan, and C. E. Koksal. Improving Loss Resilience with Multi-Radio Diversity in Wireless Networks. In *11th ACM MOBICOM Conference*, Cologne, Germany, September 2005.

[31] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Measurement-based models of delivery and interference. In *SIGCOMM*, 2006.

[32] J. Widmer and J.-Y. L. Boudec. Network coding for efficient communication in extreme networks. In *ACM SIGCOMM Workshop on DTNs*, 2005.

[33] Y. Wu, P. A. Chou, and S.-Y. Kung. Minimum-energy multicast in mobile ad hoc networks using network coding. *IEEE Transactions on Communications*, 2007.

[34] J. Zhao, R. Govindan, and D. Estrin. Computing aggregates for monitoring wireless sensor networks, 2003.