CATWOMAN: Implementation and Performance Evaluation of IEEE 802.11 based Multi-Hop Networks using Network Coding*

Martin Hundebøll, Jeppe Ledet-Pedersen, Janus Heide, Morten V. Pedersen, Stephan A. Rein, Frank H.P. Fitzek
Aalborg University, Department of Electronic Systems, Mobile Device Group
Email: jeppe@ledets.dk, {mhu, jah, mvp, sr, ff}@es.aau.dk

***Abstract*—This paper investigates the performance of network coding for an IEEE802.11 enabled meshed network. By means of basic setups the impact of the medium access control in combination with network coding is investigated. In contrast to prior work the network coding approach is tailored to commercial WiFi hardware without any special tweaks. The implementation of network coding is done on top of an existing routing scheme known as B.A.T.M.A.N. which has some inherent advantages to support network coding. We present schemes to utilize the B.A.T.M.A.N. routing to detect coding opportunities. One finding is that the performance gain for the well known Alice and Bob scenario using network coding is 60% compared to a pure relaying scheme. The software used in the presented measurement campaign is made publicly available.**

I. Introduction

To improve the performance in a wireless meshed communication system, network coding has been identified as one potential performance booster. The basic principle of network coding is to process and combine network packets instead of simply forwarding them. In order to make network coding readily applicable to mesh networks, we started a new project called CATWOMAN (Coding Applied To Wireless On Mobile Ad-hoc Networks) at Aalborg University. Based on an existing routing scheme called B.A.T.M.A.N. (Better Approach To Mobile Ad-hoc Networking), the goal was to implement network coding on standard WiFi access points.

Network coding has been introduced by Ahlswede *et al.* in 2000 [1]. Since then many works have investigated and constructed new network codes, see for instance [2], [3], [4] for linear and random linear network coding (RLNC). There also exist works that give a theoretical analysis or build simulative setups to estimate the possible gains due to network coding, e.g. for p2p networks as in [5] and for ad-hoc networks in [6], [7]. Recently, there have been provided implementations for coding and decoding algorithms [8], [9], [10] and practical setups for media distribution on current hardware [11], [12]. Little work has yet been devoted on the integration of network coding in practical ad-hoc networks. The most relevant work in this field is given in [13] introducing the COPE (Coding Opportunistically) architecture for a network coding enabled wireless mesh. While the aforementioned approach uses standard Linux PCs for each testbed node and is not optimized for energy consumption, we present a lightweight solution that runs on standard WiFi hardware. Our solution is very effective in that it utilizes the routing information and messages of the B.A.T.M.A.N. protocol to detect coding opportunities, thus making network coding readily applicable on top of an already in practice established ad-hoc protocol. Similar than in COPE we implement XOR coding. We however go for basic network topologies to get a better understanding where the particular performance gains result from. We selected the three topologies as described in the following.

*This work has been financed by the Green Mobile Cloud project granted by the Danish Council for Independent Research (Grant No. 10-081621).

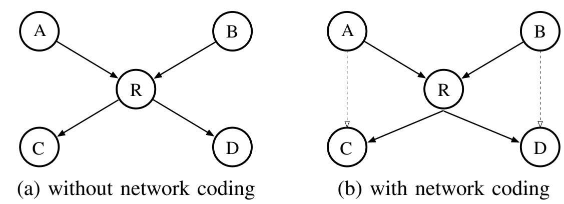
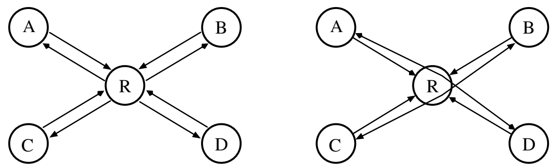


Fig. 1. X-Topology with two more nodes, Charlie and Dave, which in (b) *overhear* packets from A and B.

The first topology is called *Alice and Bob* and comprises three nodes, Alice, Bob, and a relay in the middle to allow for data exchange. In a pure relaying scenario, the relay would need to forward two packets to both, Alice and Bob. In total the pure relaying would result in four packets that need to be exchanged. Using network coding the relay combines the two incoming packets and broadcasts a combination of both in one packet. Each node can decode then a suitable packet from the received and the previously sent one. For this example a potential performance gain due to network coding of 25% is generally identified in the literature, as one packet out of four can be saved. As we will demonstrate in this paper the performance gain can even be larger.

The second setup named *X-Topology* consists of a relay, two sender and two receiver nodes, see Figure 1. Here Alice (A) transmits packets to Dave (D), and Bob (B) transmits packets to Charlie (C). Similar than in the Alice and Bob topology, the relay (R) combines the incoming packets from A and B and broadcasts the coded result to C and D. As C and D can *overhear* packets, they are able to decode the packets from R. With network coding the relay sends half the number of packets, while C and D listen twice as much. In terms of energy, network coding is not necessarily favorable [14].



(a) without network coding (b) with network coding

Fig. 2. Crossed Alice and Bob with two bidirectional flows. With network coding the relay combines the packets for each sender pair.

The third example is denoted as *Crossed Alice and Bob*. It considers two bidirectional flows, one between Alice and Dave and the other between Bob and Charlie, see Figure 2. All flows again go through the relay node R, and thus the total throughput is bound by the commonly used MAC capacity.

II. B.A.T.M.A.N. Protocol

In order to build a mesh network that utilizes network coding, we select the B.A.T.M.A.N. protocol as a starting point. In the following we describe the protocol features based on the *Originator Messages* (OGM) that are utilized for our coding opportunity discovery, see [15] for more details.

A. Node Discovery

Upon receipt of an OGM packet, the receiver node generally updates its routing table and then rebroadcasts the packet. Thereby every node in the mesh network receives the OGM message at least once if the packet does not get lost before. The packet contains the address of the OGM originator node, thus the receiver node can learn that the originator node is reachable and the next hop towards him. It also contains the address of the previous sender. In case the receiving node is also the previous sender (*echo* message), the packet is not sent again. It is also not resent if the receiving node is the originator. When rebroadcasting OGM packets, the TTL field in the packet is updated, thus preventing endless loops of OGMs.

B. Best Route Estimation

The B.A.T.M.A.N. protocol routes packets by letting each node on the path select the best next hop towards the destination. The routing table contains a quality metric for each of the possible next hops, which is estimated through the OGM messages. Each OGM packet contains a Transmit Quality (TQ) field to signal the route quality towards the OGM originator from the OGM sender node. When the OGM is received by a node, this node calculates the route quality towards the OGM originator by multiplying the TQ value from the received OGM with its locally available quality towards the OGM sender, as illustrated in Figure 3. The TQ field is updated in the OGM message with this combined quality on each node while it traverses the network. The locally available TQ quality to a neighbor node is estimated as follows. Every node maintains a receive quality RC and an echo quality EC. The receive quality is estimated from the last N (= 128 by default) OGM messages in that the percentage of messages received by the particular neighbor node is calculated, see Figure 4 (a). As given in Figure 4 (b), the echo quality is similarly calculated as the percentage of locally generated OGMs received (as an echo) from a neighbor. As the echo quality gives the probability that a packet can be sent back and forth, it is given as the product of RQ and TQ ($EQ = RQ \cdot TQ$), from which the TQ estimate can be derived.

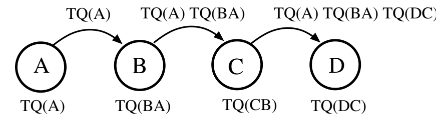
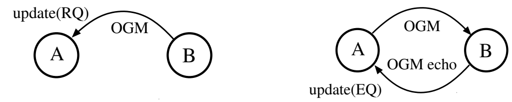


Fig. 3. B.A.T.M.A.N. transmit quality estimation to the OGM message originator.



(a) Receive quality (b) Echo quality

Fig. 4. B.A.T.M.A.N. local transmit quality estimation from B–A utilizing *receive* and *echo* quality estimated both from the last N received OGM messages.

III. CATWOMAN

In this Section we explain our contributions to the B.A.T.M.A.N. protocol to build the new network coding enabled CATWOMAN ad-hoc network.

A. Coding Opportunity Discovery

We present a scheme that utilizes the B.A.T.M.A.N. OGM messages to let the relay node discover possible coding opportunities. The scheme is two-fold, a) the relay node learns if there exists overhearing between his neighbor nodes, and b) it checks upon arrival of every packet if there is a coding opportunity using the information gained from a). If that is the case, it sends out a combined packet. To allow for this, each packet that arrives at the relay is buffered for a predefined time. Similarly, the overhearing nodes need to buffer each packet. We explain a) and b) by the X-topology example illustrated in Figure 1. The relay node learns that C can overhear packets from A in that it first stores the TTL number of an OGM message sent by A.

The OGM message is also received and resent by C. The relay node receives this OGM message again and compares its TTL with that of the previously received one. As the difference between the TTLs is only one, R now knows that C can overhear packets from A. Similarly R learns that node D can overhear packets from B. The relay node keeps a list of the overhearing opportunities, which is used in the example as follows. Node A sends a packet *a* towards node D. As no packet is in the relay's buffer, *a* is just hold in the buffer. Node B then sends a packet *b* towards C. The relay node checks its list for nodes that overhear packets from B and finds D. It then checks if C can overhear packets from other nodes and finds A. As packet *a* is still in the buffer and has A as source and D as destination, R can send out a combination of *a* and *b*.

Packet Type (1 byte)	Version (1 byte)	Destination (6 bytes)	TTL (1 byte)	Packet ID (2 bytes)

Fig. 5. Introduced packet header format for decoding and restoring packets.

B. Receiver Selection

As a coded packet needs to be received by multiple receivers, the sender has to employ an appropriate addressing for the packet. With the 802.11 MAC, there are two options, either using a broadcast packet and give the destination addresses in a packet field, or using a unicast packet and set the receiving nodes to the so-called *promiscuous mode* such that they receive all packets. The first option does not allow for the RTS/CTS sequence, thus hidden terminals are not detected. Also MAC retransmissions are not possible, and finally broadcast may achieve a lower throughput as there is no dynamic rate adaptation accounting for the channel conditions. We therefore select the second option – that is, sending the packet to a selected single destination, and accordingly signaling the remaining destinations over a packet field. One of the two destination nodes is selected randomly for the MAC802.11 destination field. The random selection is weighted with the quality estimator (TQ) to prefer the weaker link. Thus reliability is favored over throughput, and in case of almost similar channel conditions one node will not be preferred for the short-term.

C. Packet Format

In order to support network coding on top of the B.A.T.M.A.N. protocol we introduce the new coding header format given in Figure 5. It contains the B.A.T.M.A.N. type and version, the length of the coded data and two group fields, each of them describing one of the original packets. Each group description contains the original sources, the packet IDs, and the TTLs. The sources are together with the given destinations required to identify the decoding packets, since packet IDs are only unique for a specific source-destination pair. The description of the second included packet also contains the additional destination, which does not fit into the MAC header.

IV. The Network Coding Testbed

To evaluate the performance of the network coding protocol, tests are carried out in a real wireless network. The network coding testbed consists of five Open-Mesh OM1P routers (available at www.openmesh.com), and four laptops for traffic generation and throughput monitoring. Each laptop is connected to a central test-server for test coordination and data collection. The OM1P router is based on an Atheros AR2315 Wireless System-on-a-Chip, with a 200 MHz MIPS processor and 32 MB of RAM. All routers are configured with the open source firmware OpenWRT version 10.03.1-RC5 with the 2.6.30.10 Linux kernel. The selected B.A.T.M.A.N software is compatibility version 12 and is patched with the CATWOMAN implementation, which we make available at **https://github.com/jledet/batman-adv-nc/tree/nc**.

Since the test network contains a number of hidden node scenarios (Alice and Charlie should not overhear transmissions from Bob and Dave, and opposite), all nodes are configured to use the RTS/CTS mechanism. In order to avoid the tests being biased due to varying rates, the rate adaptation algorithm is disabled, and all nodes are set to a fixed speed of 11 Mbit/s. The maximum rate between two nodes has been measured to be 5.4 Mbit/s and is considered the channel maximum when using 802.11b devices. The nodes are all configured to transmit with 10 dBm and are placed in a university building with a large open space atrium. Every node is placed to have a direct line of sight to the relay node, as illustrated in Figure 6. The test execution is handled from a central server.

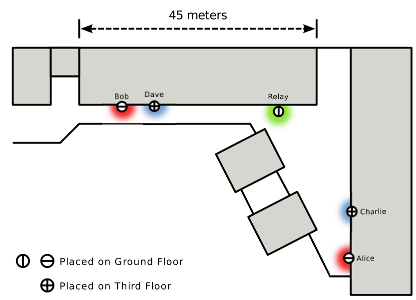


Fig. 6. Setup of the testbed. The nodes were placed in a large open space atrium to have line of sight to the relay node at different floors but far enough to not have direct routes between the sides.

V. Measurements and Results

The primary purpose of the test is a verification of the expected throughput increase when network coding is enabled for a large range of loads. To evaluate CATWOMAN in the described test network, we use the Iperf Linux tool to generate UDP traffic with increasing rates in each of the three topologies (using equally spaced 1498 byte packets). Iperf is also used to measure the throughput, which is calculated from the number of packets received and the transmission speed, e.g., if 50% of the sent packets is received, and the sending rate was 1 Mbit/s, the throughput is 500 kbit/s. Each measurement lasts 30 seconds and is repeated 10 times. Note that our testbed is running on a University campus and is surrounded by several WiFi access points. Due to beacons from 3rd party wireless access points the channel has a constant minimum load of approximately 200 kbit/s. This extra load should have a minimal influence when the network is congested. All tests are carried out at night to have a minimum of interference.

A. Alice and Bob

Figure 7 shows the aggregated throughput for pure relaying and network coding vs. totally offered load for the Alice and Bob topology. In addition the figure shows also the system gain achieved by network coding (indicated as coding gain) comparing both approaches in the bottom. In case the load from Alice and Bob is low (smaller than 2500 kbit/s) there is nearly no gain from network coding. In fact, the throughput with coding is slightly smaller than without coding due to the fact that the relay is holding incoming packets for up to 10 ms to wait for coding potential. At an induced load higher than

2600 kbit/s the performance without network coding degrades. The reason for this behavior is the MAC fairness that in case of network congestion shares the bandwidth between the three entities Alice, Bob, and the relay equally. The IEEE802.11 MAC is agnostic to the fact the relay is supporting Alice and Bob and thus shares the bandwidth in terms of bits equally. The point where the channel gets fully loaded is roughly 5.5 Mbit/s - 0.2 Mbit/s for the beacons as mentioned beforehand divided by two. This is the point where Alice as well as Bob each introduce 1.3 Mbit/s.

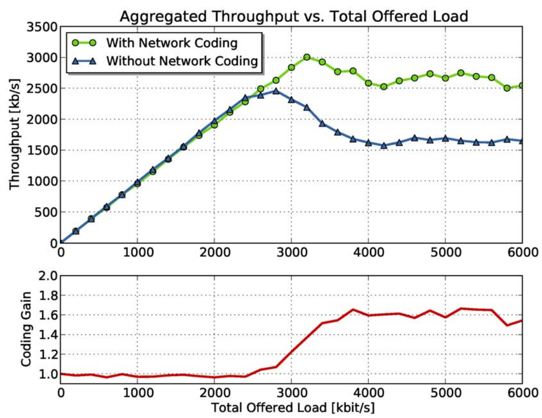


Fig. 7. Throughput vs. total load and coding gain for **Alice and Bob**.

With network coding enabled, the throughput peaks to 3 Mbit/s at a load of 3.2 Mbit/s, but then drops to a rate of approximately 2.7 Mbit/s. Consequently, the coding gain reaches 1.6. The reason why network coding achieves better throughput than the pure relaying approach lies in the reduction of required bandwidth by the relay. As a fair share is distributed among the sending entities, only network coding allows the relay to send as many packets as Alice and Bob. Without network coding the relay may not be able to handle the load from A and B if it becomes larger. The reason why the throughput does not stay stable at 3 Mbit/s is the asymmetric link behavior. Even though the testbed is configured to send packets with the same rate from Alice and Bob to the relay, there are situations where one link is better than the other due to the well known capture effect. Because of space limitation we are not adding a plot for individual throughput values, but underline that Alice has larger throughput values than Bob in highly congested network situations. The unbalanced link behavior is reflected by Figure 8. It shows the packet ratio sent by the relay with network coding compared to pure relaying, giving detailed information about the ratio of total, coded, and forwarded (without coding) packets. For the lowest offered load the total ratio is 0.66 which means the relay sends a mean number of 1.3 packets for packets coming from Alice and Bob. Note that the pure relaying would always use two packets in this case. The reason why the relay in the case of network coding is using more than one packet lies in the holding time of 10 ms. With an increased load the total ratio degrades and the portion of coded packets is increasing until no packet is forwarded any more. With a load of 800 kbit/s nearly all packets are coded. If the load exceeds roughly 3.0 Mbit/s we see that the relay forwards packets again due to asymmetrical incoming traffic from Alice and Bob.

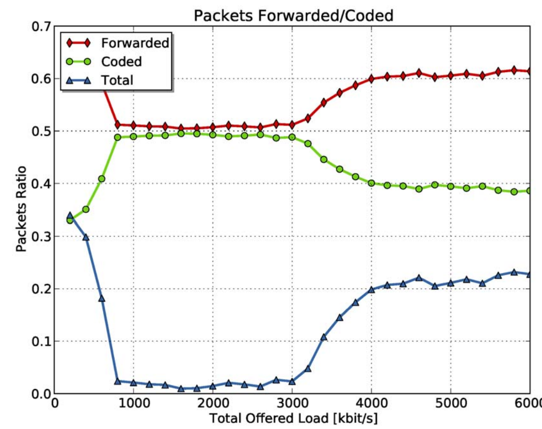


Fig. 8. Coded and forwarded packets vs. induced load for Alice and Bob. The numbers have been normalized such that 1.0 reflects the number of transmitted packets without network coding.

B. X-Topology

Figure 9 shows the aggregated throughput for pure relaying and network coding for Alice and Bob to Dave and Charlie for the X-topology. Compared to the Alice and Bob scenario the coding gain is decreasing from 1.6 to 1.4. Normally we would expect the same results as in the prior scenario, but due to failures in the overhearing as well as on the secondary link to Charlie and Dave (which is even more likely due to our setup given in Figure 6) the performance degrades for both approaches by 200 kbit/s. The achieved throughput for the individual nodes (plots are not given here due to space constraints) shows a clear advantage for Bob over Alice when the network becomes congested. This shows the impact of a real channel behavior on the coding performance. Nevertheless there is still a coding gain which should lead to the usage of network coding.

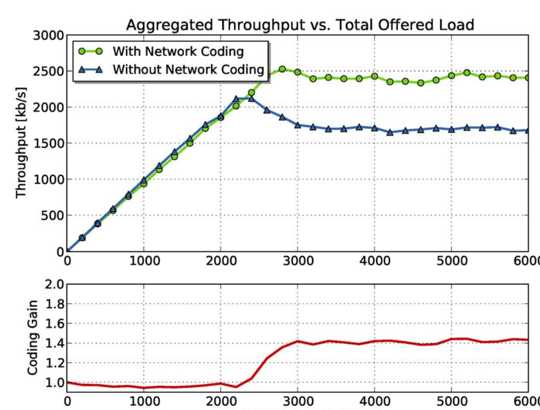


Fig. 9. Throughput vs. total load and coding gain for the **X-topology**.

C. Crossed Alice and Bob

Figure 10 gives the throughput and coding gain results for the crossed Alice and Bob scenario. The last scenario is just a duplication of the first scenario. We note that for network coding only intra-flows are coded resulting in two coded packets for the relay. The maximum coding gain of 1.6 is achieved shortly after the point of congestion when using network coding. The gain then slightly drops to 1.4 due to the equal capacity distribution, where the relay only gets $1/5$ of the capacity (due to IEEE802.11 MAC) while it would require $1/3$. The measured maximum throughput without network coding is 2780 kbit/s, which is roughly half of the measured possible maximum throughput. This is the case as the relay node needs to transmit the same amount of traffic as the sending nodes, so the senders and the relay each allocate half of the capacity.

With network coding enabled a maximum throughput of 3500 kbit/s is achieved, which is roughly $2/3$ of the measured possible maximum throughput. When the network is fully congested, the expected throughput would be $5400 \cdot 1/5$ for the case without network coding, as the MAC assigns each of the five nodes the same capacity, and the total throughput cannot be higher than the relay throughput. Using network coding the relay only needs to send half of the packets, and the expectation for the total throughput in the case of congestion would be $5400 \cdot 2/5 = 2160$ kbit/s. The measurement gives a slightly lower throughput of 1985 kbit/s, which is again due to the unequal MAC capacity distribution at the nodes. If we would rely on overhearing of all neighboring nodes (e.g. Alice can overhear Charlie and Bob) then the relay might just send one coded packet (all packets from the outer nodes into one) instead of two. This scenario remains for future work.

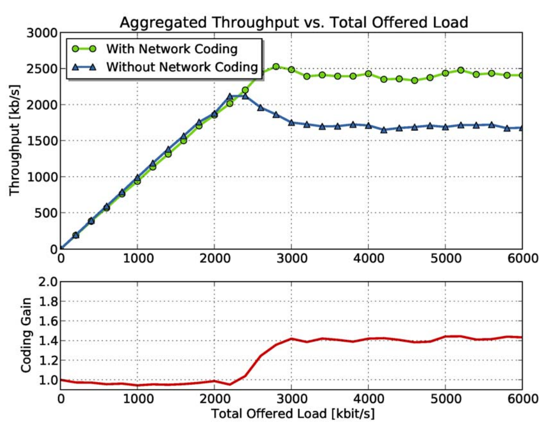


Fig. 10. Throughput vs. total load and network coding gain for the **Crossed Alice and Bob**.

VI. Conclusion

In this paper we have introduced our network coding testbed based on commercial WiFi access points. The software used for pure relaying as well as network coding, referred to as CATWOMAN framework, has been made available for the research community to be able to repeat and extend our measurements. We have integrated network coding into a given routing protocol via a new scheme that utilizes the existing routing information for detection of coding opportunities. Three basic topologies have been investigated, namely Alice and Bob, X-topology, and crossed Alice and Bob. The basic setups allow us to show several impact factors such as the medium access, the topology, the load, the asymmetry of the links, and the capture effect. Our performance evaluation revealed that for all three scenarios the use of network coding was advantageous with coding gains between 40% and 60% for the high load situations. One finding is that network coding is not associated with a fixed and pre-known performance gain, but is dependent on the aforementioned parameters. The results show a clear benefit of network coding over the pure relaying strategies and will hold true for larger networks as well, because these build on the elementary topologies. In the future we will investigate the energy due to sending, receiving, and computation, especially for lower load situations where no capacity improvement takes place.

References

[1] R. Ahlswede *et al.*, "Network information flow," in *IEEE Trans. on Inform. Theory*, vol. 46, no. 4, July 2000, pp. 1204–1216.

[2] R. Koetter and M. Medard, "An algebraic approach to network coding," in *IEEE Trans. on Networking*, vol. 11, no. 5, Oct. 2003, pp. 782 – 795.

[3] S.-Y. Li, Q. Sun, and Z. Shao, "Linear network coding: Theory and algorithms," in *Proc. of the IEEE*, vol. 99, no. 3, 2011, pp. 372 –387.

[4] T. Ho *et al.*, "The benefits of coding over routing in a randomized setting," in *Proc. of the IEEE Int. Symp. on Inform. Theory*, June 2004.

[5] Z. Zhang *et al.*, "Network coding based live peer-to-peer streaming towards minimizing buffering delay," in *Proc. of the Int. Conf. on Computer App. and Syst. Modeling*, vol. 4, Oct. 2010, pp. 136–140.

[6] S. Tarapiah, C. Casetti, and C.-F. Chiasserini, "Network coding in ad hoc networks: A realistic simulation study," in *Proc. of IEEE INFOCOM Workshops*, April 2009, pp. 1–2.

[7] J. Meng and X. Pan, "The optimization study of network coding for coding-aware wireless mesh networks," in *Proc. of the Int. Conf. on Future Computer and Communication*, vol. 2, May 2010, pp. 537–540.

[8] J. Heide, M. Pedersen, F. Fitzek, and M. Medard, "On code parameters and coding vector representation for practical RLNC," in *IEEE Int. Conf. on Communications*, June 2011, pp. 1 –5.

[9] J. Heide, M. Pedersen, and F. Fitzek, "Decoding algorithms for random linear network codes," in *Proc. of the Int. Conf. on Networking*, 2011.

[10] M. Pedersen, J. Heide, and F. Fitzek, "Kodo:an open and research oriented network coding library," in *Proc.of the Int. Conf.on Netw.*, 2011.

[11] P. Vingelmann, F. Fitzek, M. Pedersen, J. Heide, and H. Charaf, "Synchronized multimedia streaming on the iphone platform with network coding," in *IEEE Comm. Magazine*, vol. 49, no. 6, May 2011.

[12] M. V. Pedersen, J. Heide, F. H. P. Fitzek, and T. Larsen, "A mobile application prototype using network coding," in *European Trans. on Telecomm.*, vol. 21, no. 8, Dec. 2010, pp. 738–749.

[13] S. Katti *et al.*, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 4, Oct. 2006.

[14] J. Heide *et al.*, "Cautious view on network coding - from theory to practice," in *Comm. and Networks*, vol. 10, no. 4, 2004, pp. 403–411.

[15] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better approach to mobile ad-hoc networking," in *IETF Internet Draft*.