

Impact of Network Coding on Delay and Throughput in Practical Wireless Chain Topologies

Hundebøll, Martin; Rein, Stephan Alexander; Fitzek, Frank

Published in:
2013 IEEE Consumer Communications & Networking Conference Proceedings

DOI (link to publication from Publisher):
[10.1109/CCNC.2013.6488474](https://doi.org/10.1109/CCNC.2013.6488474)

Publication date:
2013

Document Version
Early version, also known as pre-print

[Link to publication from Aalborg University](#)

Citation for published version (APA):
Hundebøll, M., Rein, S. A., & Fitzek, F. (2013). Impact of Network Coding on Delay and Throughput in Practical Wireless Chain Topologies. In *2013 IEEE Consumer Communications & Networking Conference Proceedings* (pp. 381-386). Article 6488474 IEEE Press. <https://doi.org/10.1109/CCNC.2013.6488474>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Impact of Network Coding on Delay and Throughput in Practical Wireless Chain Topologies*

Martin Hundebøll

Department of Electronic Systems
Aalborg University, Denmark
Email: mhu@es.aau.dk

Stephan A. Rein

Department of Electronic Systems
Aalborg University, Denmark
Email: sr@es.aau.dk

Frank H.P. Fitzek

Department of Electronic Systems
Aalborg University, Denmark
Email: ff@es.aau.dk

Abstract—In this paper, we present results from a practical evaluation of network coding in a setup consisting of eight nodes deployed in a chain topology. With the tradition pure relaying, delay increases dramatically as the network gets congested, and here network coding helps to moderate this increase in delay, as well improving throughput. The practical evaluation shows that network coding provides up to a five-fold decrease in delay, while retaining the expected gain in throughput. To address an unnecessary delay when using network coding in low-load scenarios, we propose and evaluate a scheme for adaptive buffering. With this, we show that the benefits from pure relaying can be combined with the improved performance from network coding. The software used to apply network coding and evaluate this in a practical network is made publicly available for further research and tests.

I. INTRODUCTION

Network coding has, since its introduction in [1], been subject to intensive research, and now finds its applications in areas as wireless networks, distributed storage, security, content delivery, sensor networks, etc. In ad-hoc wireless mesh networks, network coding can exploit the shared medium to broadcast information to multiple nodes in a single transmission, either by using Random Linear Network Coding or simple XOR'ing. The latter is investigated thoroughly in [2], where the COPE scheme implements network coding to combine multiple packets in one transmission. A similar approach is implemented by CATWOMAN in [3], which shows results similar to COPE. Furthermore, CATWOMAN has been evaluated with respect to energy efficiency in [4] and [5], and also confirms analytical results in [6].

An obvious use case of wireless mesh networks is to establish internet connectivity in environments that lack the needed infrastructure. Examples of this range from single buildings without the needed cabling, to disaster areas where the existing cabling is damaged. Common for most of the scenarios is a single point with internet connection, e.g. in a hotel lobby or on a satellite phone, which users access through wireless networks. These scenarios form topologies where multiple relays carry packets between the user and



Fig. 1. General chain topology with N relays to forward packets between the two end-nodes, A and B .

the internet access point. In these chained topologies where multiple relays forward packets between two end-nodes, as illustrated in Figure 1, network coding can increase throughput as described in [2], and also influence the delay.

The shared medium in such chain topologies limits the performance supplied to the users, who expect minimal latency when browsing, messaging, calling, video conferencing, etc. While delays in wireless mesh networks has been researched thoroughly (e.g. [7], [8]), the performance of network coding with regard to delays has received little attention.

In [9], an analysis with network calculus of three simple topologies from [2], concludes that network coding can decrease the delay in wireless mesh networks. We take this conclusion a step further by analysing and investigating the delay in chain topologies with and without the CATWOMAN network coding scheme. Furthermore, we propose and evaluate an adaptive buffering scheme, that improves delay performance in low-load scenarios.

By deploying a real wireless mesh network and evaluate the performance with and without CATWOMAN, we make the following findings: 1) CATWOMAN can significantly reduce the packet delay and our measurements confirms this with a five-fold decrease; 2) The throughput gain reaches 200% in a seven-hop topology; and 3) Adaptive buffering eliminates the delay caused by CATWOMAN in low-load scenarios.

The contents of this paper are organized as follows. In Section II we give a brief introduction to the network coding concepts relevant for the investigation of throughput and delay in wireless multi-hop networks. In Section III we describe the practical setup used to evaluate the performance. We then present and describe the obtained results in Section IV, before we finally discuss the result and conclude in Section V.

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

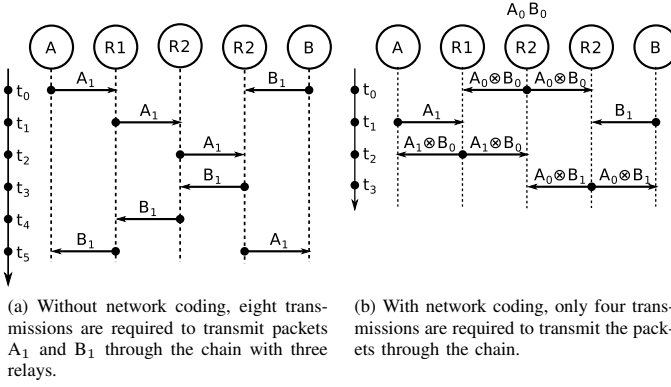


Fig. 2. Transmissions required to transmit one packet from Alice to Bob, and one packet from Bob to Alice, with and without network coding. The center relay initially holds one packet from Alice and one from Bob and ends up with new packets from Alice and Bob.

II. NETWORK CODING

By applying intersession network coding to wireless mesh networks, we let relays exploit the shared medium to transmit combinations of P packets to P destinations in a single transmission. To extract information from such a combined packet, the receiver reverses the combination by adding $P - 1$ packets already included in the combination. In the case of two packets, we express this as:

$$(p_1 \otimes p_2) \otimes p_1 = p_2$$

where \otimes is the XOR operation.

In the simple *Alice-and-Bob* topology with two end-nodes (Alice and Bob), and a relay, we save one of four transmissions with network coding, leading to a *coding gain* of 1.25. The coding gain is calculated as

$$1 + \frac{(N \cdot 2) - (N + 1)}{N \cdot 2}$$

where N is number of relays, $(N + 1)$ is the number of transmissions with network coding, and $(N \cdot 2)$ is the number of transmissions without network coding. [2]

Figure 2 illustrates transmissions in the Alice-and-Bob topology extended with two additional relays. Here, the relays receive coded packets from each other, decode them, and recode them with packets from Alice and Bob. Since we now have three relays, the coding gain increases to 1.33. When increasing the number of relays, the coding gain reaches 1.5 as illustrated in Figure 3.

In a real network, where the maximum number of packets per second is limited by the channel capacity, we can utilize the saved transmissions from network coding to increase the maximum throughput. This is illustrated in Figure 4, where the maximum number of transmissions per second in a chain with six relays is limited to 100. As the load from Alice and Bob increases, the number of total transmissions in the chain

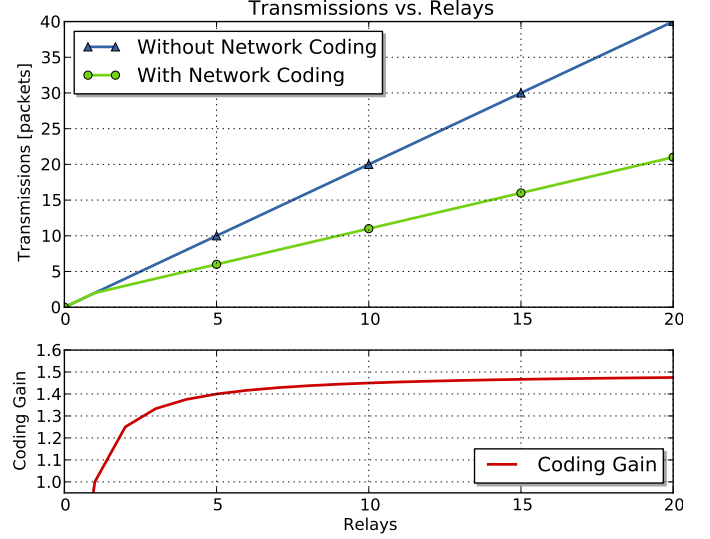


Fig. 3. Transmissions needed to send one packet from Alice to Bob, and one packet from Bob to Alice, as the number of relays increases (top), and the resulting coding gain (bottom).

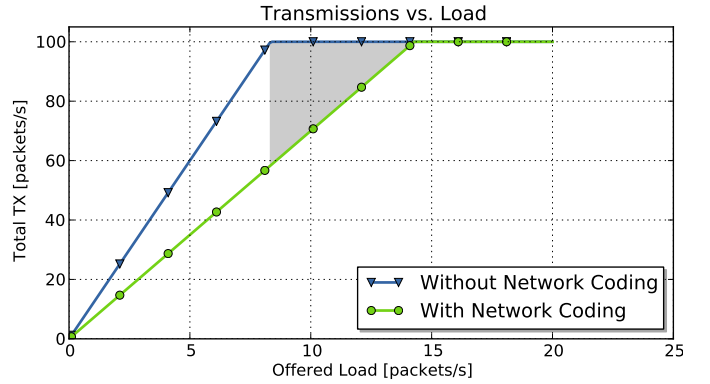


Fig. 4. When applying network coding to a wireless mesh network, the maximum number of transmitted packets is reached at a later point when compared to without network coding.

increases accordingly. In the greyed area in Figure 4, network coding allows Alice and Bob to transmit additional packets and thus increase the throughput.

A. Adaptive Buffering

The network coding is implemented as simple XOR'ing of two packets. The combined packet is transmitted as IEEE 802.11 unicast data, with an additional address in the payload used to identify the second receiver, which receives the packet in promiscuous mode. This approach enables the driver to do retransmissions to one of the two receivers and transmit at higher bit rates.

All nodes buffer non-combined packets to use to decode combined packets. Packets are buffered upon own transmissions, when overhearing transmissions between other nodes, and before combining two packets with network coding. The

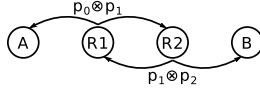


Fig. 5. When a $R2$ receives a combination of p_0 and p_1 , it combines the decoded p_1 with p_2 . Now $R1$ uses the buffered p_1 to decode the combination of p_1 and p_2 .

latter is necessary to be able to decode combinations of packets decoded from another combined packet as illustrated in Figure 5.

To achieve opportunities to combine packets, packets are buffered before being forwarded to the next hop. When a new packet arrives, the buffer is searched for a packet to combine. If one is found, the combined packet is transmitted. However, in low load scenarios, the next packet may arrive too late to be combined, and thus the first packet has been buffered unnecessarily, leading to increased delay at each hop. To address this issue, we propose an adaptive buffering scheme.

Adaptive buffering detects congestion by tracking the retry count from the wireless driver. Every second it calculates the number of retries since the last check, and if the count exceeds a user defined threshold, the wireless device is marked as congested and packets are buffered before being forwarded.

In high load scenarios, new packets arrive with short intervals and buffered packets are combined with arriving packets and are thus transmitted without too much additional delay.

III. EXPERIMENTAL SETUP

To evaluate the throughput and delay performance with network coding, we deploy eight nodes in the campus buildings. Each node is controlled and configured from a central server, which also executes tests and fetches results from the two end-nodes (Alice and Bob). We use the CATWOMAN (Coding Applied To Wireless Over Mobile Ad-hoc Networks) scheme from [3]. CATWOMAN is developed as an extension to the *batman-adv* kernel module, which is a state-of-the-art implementation of the B.A.T.M.A.N. mesh routing protocol [10]. The CATWOMAN scheme accesses neighbor information from the B.A.T.M.A.N. protocol, which it uses to identify coding opportunities, and thus requires no configuration. The source code of CATWOMAN and *batman-adv* is available at <http://kom.aau.dk/~mhu/img/catwoman.tar.gz>.

A. Configuration

The deployed nodes are based on two platforms: Six *OM1P* routers from <http://open-mesh.com> are used as relays, and two *Pandaboard ES* from <http://pandaboard.org> are used to generate traffic on the end-nodes. Each node is configured to use rate adaptation with *minstrel* on the following rates: 6, 9, 12, 18, and 24 Mb/s. (Initial tests with higher rates actually revealed worse performance both with and without network coding.) Minstrel is a scheme for rate adaptation, which periodically probes another rate than the one currently



Fig. 6. The testbed topology, where eight nodes are placed to form a chain.

selected, to measure the expected throughput of each rate and pick the best one. RTS/CTS is disabled and TX power is left untouched. When CATWOMAN is enabled, packets are buffered for 50 ms before being forwarded; transmitted and overheard packets, which might be needed to decode combined packets, are buffered for 1500 ms.

The OM1P runs on the Atheros AR2317 MIPS SoC with 180 MHz CPU, 32 MB RAM, and 8 MB flash. It is configured with an Atheros AR5312 A/B/G wireless device and runs a custom build of OpenWRT Attitude Adjustment (revision 32613), which can be downloaded at <http://kom.aau.dk/~mhu/img/openwrt-om1p.img.gz>.

The Pandaboard runs on the Texas Instruments OMAP4460 SoC with dual-core 1.2 GHz CPU, 1 GB RAM and a 4 GB SD-card. It is configured with a D-Link AirPlus G DWL-G122 USB wireless dongle, which is based on a Ralink RT2070 chip. The operating system is Arch Linux ARM and an image is available at <http://kom.aau.dk/~mhu/img/pandaboard.img.gz>.

Performance evaluations are coordinated by a controlling server, which connects to each node and configures it by sending a list of parameters. Each evaluation is conducted by using *iperf* (patched to measure end-to-end delay) to load the network with a range of rates and measure the actual throughput and delay. Each single rate is tested 10 times for 30 seconds with and without network coding. To minimize the influence from other surrounding wireless networks, tests are conducted during the night, and the test network is idle for 15 seconds between every test to let the network settle.

B. Node Deployment

The nodes are placed in the campus buildings to form the chain topology illustrated in Figure 6. The buildings create a rather harsh environment with heavy concrete and has several surfaces covered in metal. The nodes are placed, as illustrated in Figure 7, approximately 20 meters apart, except for the two end-nodes, which are placed next to a relay, due to the low range of the embedded antennas on the USB wireless dongle. Each relay is not entirely isolated to its next-hop neighbors, and to ensure the use of the entire path when testing, each node is configured to discard protocol packets from all nodes except its neighbor(s).

IV. EXPERIMENTAL RESULTS

The performance of CATWOMAN is evaluated both with and without the adaptive buffering scheme. First, we compare plain CATWOMAN to pure relaying and look at both throughput and delay, as well as how CATWOMAN interacts with MAC retries and transmissions. Secondly, we enable the adaptive buffering, and compare it to pure relaying.

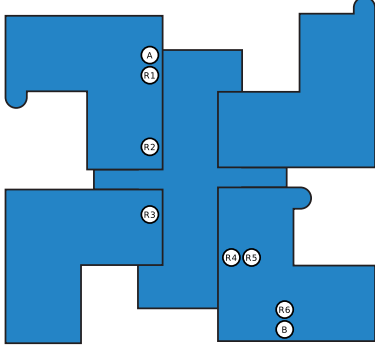


Fig. 7. The location of the eight nodes in the campus buildings. Note that $R3$ and $R4$ are placed on 2nd floor, while the rest is on 1st floor.

A. Plain CATWOMAN

The expected coding gain for a chain with 6 relays is 1.42, but since we have a topology with multiple hidden terminals, and our combined packets are not retransmitted to one of the two receivers, we expect higher packet losses. On the other hand, CATWOMAN benefits from the MAC gain described in [2] and [3], which makes up for the increased packet loss. We also expect CATWOMAN to show high delays when the network is not congested. At the point where the network gets congested (point of congestion), delays with network coding should stay low until combined packets also requires retransmissions.

Figure 8 shows the throughput of the system with and without network coding. Initially, the throughput with CATWOMAN is roughly 80% of pure relaying, which confirms the packet loss due to missing retransmissions. At 800 kbit/s, we see the peak with pure relaying, which then drops as the induced load is increased. CATWOMAN peaks with 1380 kbit/s and drops slightly as pure relaying also does. When comparing the two peaks, we get a throughput gain of 1.7 and when comparing the throughputs at an induced load of 1800 kbit/s, we see a throughput gain of 2.5.

The delay measurements in Figure 9 are in line with our expectations: The buffering of forwarded packets with CATWOMAN increases the base delay from 25 ms without CATWOMAN, to 275 ms in the worst case at 200 kbit/s. As the number of combined packets increases, the delay with CATWOMAN decreases to 162 ms at the point before the network is congested (at an induced load of 800 kbit/s). After congestion, delay with pure relaying increases rapidly, while delay with CATWOMAN increases more steadily. At an induced load of 1600 kbit/s, the delay with pure relaying is 5.4 times higher than with CATWOMAN.

In Figure 10 we see the reason for the better performance of CATWOMAN. When saving transmissions by combining packets, the outgoing queue size is reduced and thus fewer retries are required to transmit the double amount of data. The number of transmissions is plotted in Figure 11. When

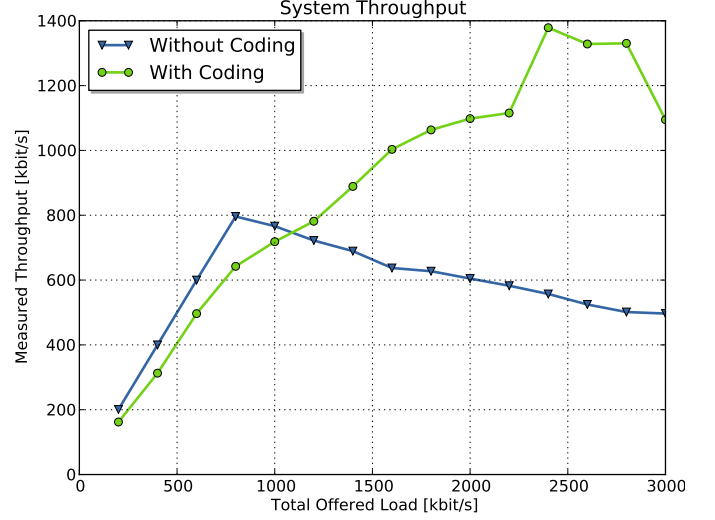


Fig. 8. Throughput with and without network coding. Initially, throughput with network coding suffers from packet loss due to collisions and lack of retransmissions towards the overhearing receiver. In high loads, the throughput drops due to increasing number of retransmission and thus larger intervals between packets.

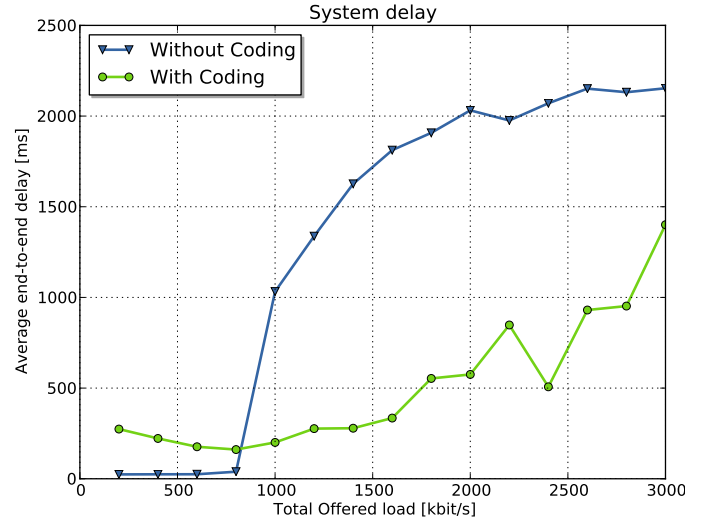


Fig. 9. Delay with and without network coding. Under low loads, the buffer time with network coding causes higher delays than with conventional store-and-forward routing. In high loads, network coding squeezes more packets through the network and thus decreases the delay.

comparing the two numbers at the point of congestion without CATWOMAN (800 kbit/s), we get a measured coding gain of 1.42, which is equal to the expected coding gain. After the point of congestion, the saved transmissions are used to transport additional packets, and the coding gain is thus not deducible.

B. Adaptive Buffering

To address the issue with CATWOMAN and high delay in low load scenarios, we propose and evaluate the scheme with

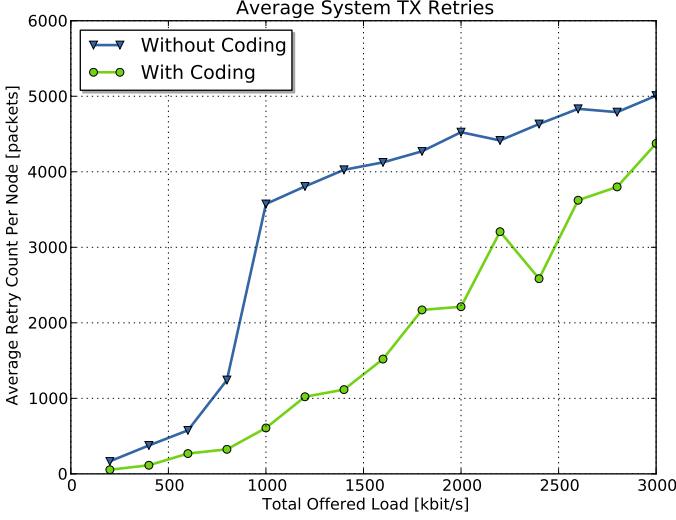


Fig. 10. TX retries with and without network coding when network coding is always enabled.

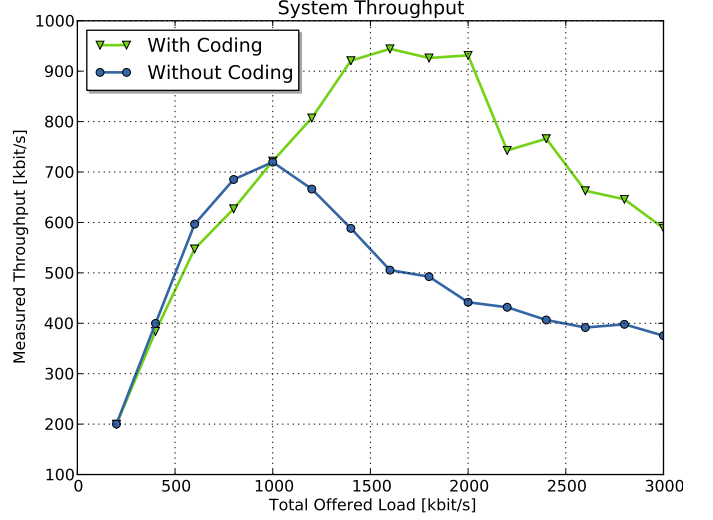


Fig. 12. Throughput with and without network coding when using adaptive buffering.

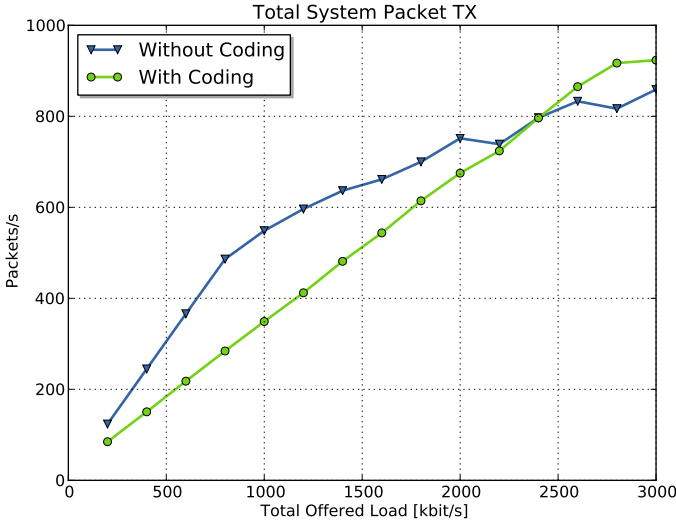


Fig. 11. Packet counts with and without network coding when network coding is always enabled.

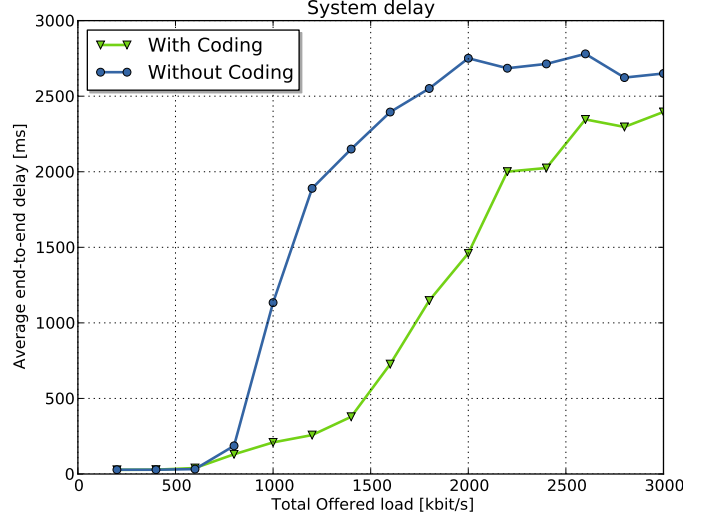


Fig. 13. Delay with and without network coding when using adaptive buffering.

adaptive buffering. The evaluation is performed in the same manner as with plain CATWOMAN, except that network coding is only enabled when the driver reports more than 50 retries per second. With adaptive buffering, we expect to see low delay with CATWOMAN until the point of congestion, and otherwise results similar to the tests with plain CATWOMAN.

In Figure 12, we see that adaptive buffering improves the packet loss with CATWOMAN, since fewer packets are combined and thus lost because of missing retransmission. As the induced load increases, the number of combined packets increases accordingly, resulting in a slight increase in packet loss. At the point of congestion, CATWOMAN performs similar as without adaptive buffering, because the retry counts

are constantly above the threshold.

The delay in Figure 13 reveals the desired result of adaptive buffering, where the delay during low load matches the delay with pure relaying. However, when the network gets congested, CATWOMAN still reduces the delay to a level that is five times lower than without CATWOMAN.

V. CONCLUSION

In ad-hoc wireless mesh networks, packets often travel through multiple relays, and each hop must share the medium with its neighbors. The shared medium can be utilized more efficiently with network coding, which enables the relays to transmit multiple packets in one transmission.

To investigate how network coding influences delay in multi-hop wireless networks, we use the CATWOMAN implementation on a setup with eight nodes deployed in a chain topology and run repeated tests with and without network coding. In our tests, network coding reduces the delay up to five times, while keeping the throughput gain from network coding.

Since CATWOMAN seeks coding opportunities by buffering packets before forwarding them, an increase in delay is seen when few packets travel the network. This leads to an unneeded increase in delay when network load is low. To address this issue, we propose and evaluate a scheme for adaptive buffering, which shows promising results. Adaptive buffering eliminates the increased delay with low loads, while retaining the reduced delay and increased throughput at high loads.

The transmission of combined packets relies on promiscuous mode to allow multiple receivers of one packet. With this approach, CATWOMAN experiences an increased packet loss, caused by missing retransmissions to one of the two receivers in case of packet loss, which is more likely to happen in a chain topology with multiple hidden nodes. One alternative is to implement acknowledgement packets in CATWOMAN, but this adds complexity to an otherwise simple protocol and should be added to the MAC protocol in the wireless driver.

REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network Information flow," in *IEEE Transactions on Information Theory*, vol. 46, Jul. 2000, pp. 1204–1216.
- [2] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," in *ACM SIGCOMM Comput. Commun. Rev.*, vol. 36, Aug. 2006.
- [3] M. Hundebøll, J. Ledet-Pedersen, J. Heide, M. Pedersen, S. Rein, and F. Fitzek, "CATWOMAN: Implementation and Performance Evaluation of IEEE 802.11 based Multi-Hop Networks using Network Coding," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*, accepted, Quebec, Canada, Sep. 3–6, 2012.
- [4] A. Paramanathan, U. W. Rasmussen, M. Hundebøll, S. A. Rein, F. H. Fitzek, and G. Ertli, "Energy Consumption Model and Measurement Results for Network Coding-enabled IEEE 802.11 Meshed Wireless Networks," in *IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD)*, accepted, Barcelona, Spain, Sep. 17–19, 2012.
- [5] A. Paramanathan, J. Heide, P. Pahlavani, M. Hundebøll, S. A. Rein, F. H. Fitzek, and G. Ertli, "Energy and Data Throughput for Asymmetric Inter-Session Network Coding," in *IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (CAMAD)*, accepted, Barcelona, Spain, Sep. 17–19, 2012.
- [6] F. Zhao, M. Médard, M. Hundebøll, J. Ledet-Pedersen, S. Rein, and F. Fitzek, "Comparison of Analytical and Measured Performance Results on Network Coding in IEEE 802.11 Ad-Hoc Networks," in *The 2012 International Symposium on Network Coding*, accepted, Boston, USA, Jun. 29–30, 2012.
- [7] N. Bisnik and A. Abouzeid, "Delay and Throughput in Random Access Wireless Mesh Networks," in *ICC '06. IEEE International Conference on Communications*, vol. 1, Jun. 2006, pp. 403–408. DOI: 10.1109/ICC.2006.254761.
- [8] J. Yackoski and C.-C. Shen, "Managing End-to-End Delay for VoIP Calls in Multi-Hop Wireless Mesh Networks," in *INFOCOM IEEE Conference on Computer Communications Workshops*, Mar. 2010, pp. 1–6. DOI: 10.1109/INFCOMW.2010.5466678.
- [9] H. Li, X. Liu, W. He, J. Li, and W. Dou, "End-to-End Delay Analysis in Wireless Network Coding: A Network Calculus-Based Approach," in *31st International Conference on Distributed Computing Systems (ICDCS)*, Jun. 2011, pp. 47–56. DOI: 10.1109/ICDCS.2011.31.
- [10] A. Neumann, C. Aichele, M. Lindner, and S. Wunderlich, "Better Approach To Mobile Ad-hoc Networking," in *IETF Internet Draft*, Apr. 2008.