

Network Coding for Flooding-Based Routing in Underwater Sensor Networks

Elvin Isufi
University of Perugia, Dept. Eng.,
06125 Perugia,
Italy
elvinisufi@yahoo.it

Geert Leus
Delft Univ. of Technology, Dept.
Electrical Eng., 2628 CD Delft,
The Netherlands
g.j.t.leus@tudelft.nl

Henry Dol
TNO, Acoustics & Sonar Dept.
2597 AK The Hague,
The Netherlands
henry.dol@tno.nl

ABSTRACT

In this work, we propose a flooding-based routing protocol using network coding for underwater communications. Due to the high amount of duplicates that flooding-based protocols flood into the network, the sharing of information between the duplicates can improve the packet delivery ratio (PDR). Our simulations show that network coding increases the PDR, but a price is paid in terms of end-to-end delay and number of forwarded duplicates, with respect to other flooding-based protocols. In order to reduce the number of duplicates, while keeping the PDR and the end-to-end delay unchanged, we propose to upgrade the protocol with specific geographical information of the nodes.

General Terms

Performance, Design, Reliability.

Keywords

Network coding, underwater communications, flooding-based routing, geographical routing.

1. INTRODUCTION

Today, underwater applications need an extension from an underwater acoustic link to an underwater sensor *network* [1]. Due to the peculiarities of the acoustic links [2] – [5], efficient routing schemes must be found in order to deliver the packets to the destination and increase the packet delivery ratio (PDR). Some of them are summarized in [2] and [6]. Network coding (NC) has been proposed recently as a novel approach to increase the PDR in a sensor network [7], where each node does not simply *store and forward* the packets, but instead uses a *store, encode and forward* approach. Due to its benefits, in the last decade, *linear* NC has been proposed for underwater networks [8] – [10], where the output packets are linear combinations of the packets presented in the node's buffer.

In this work, we propose a novel NC approach for underwater sensor networks. Our proposal fuses linear NC with the Dflood routing protocol [11]. These authors have proposed a duplicate reduction flooding-based routing protocol for underwater sensor

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

WUWNET '14, November 12 - 14 2014, Rome, Italy
Copyright 2014 ACM 978-1-4503-3277-4/14/11...\$15.00
<http://dx.doi.org/10.1145/2671490.2674570>

networks. Considering that the number of packets flooded in the network is still high, even when using the Dflood rules,¹ we think that sharing the information by NC will help the destination to receive enough independent packets (not only replicas) and retrieve the information. With respect to the other NC papers for underwater sensor networks, our proposal differs in some points. Instead of [8], our idea does not consist of overlapping NC over the considered protocol. Instead, we review the protocol's rules, in order to fully exploit the full NC potential. We further have developed our protocol for implementation in real scenarios, and not in a chain topology as in [9]. The difference from [11] is that our proposal does not assume that the packets to be encoded arrive at the same time at the next relays. Instead, they arrive with different and variable delay or some of the relays do not receive any packet due to the link losses.

In our proposal, we have considered the linear NC scheme proposed in [13], where the g information packets of a generation that a source node has are linearly encoded into $h \geq g$ output packets. The selection of the encoding coefficient is done following the rules of [14], where each node in the network selects the coefficients in a random manner, uniformly distributed in F_{2^s} . In this way, all the operations are done independently and in a completely decentralized manner. However, selecting randomly the coefficients may lead to linearly dependent combinations, which happens with a probability related to the size of F_{2^s} [14], but [15] has shown that, in practice, $s = 8$ is sufficient to have a full-rank decoding matrix with very high probability. Considering that the coefficients are chosen locally at each node, the encoding vectors must be included in the packet headers in order to decode them and to allow for recursive encoding.

The rest of this paper is organized as follows. In Section 2, we will present our proposal on using network coding in a flooding-based fashion. The simulation setup and results are shown in Section 3, and the work will be concluded in Section 4.

2. PROPOSED PROTOCOL

For our protocol (NC-Dflood), we have considered that each packet has a *hop count*, a *source* and *destination address*, and a *generation number*. The combination of the last three forms a unique identifier for the packets of the same generation. We will call innovative packets those that are not linear combinations of the packets present in the node's buffer holding the same

¹ To the rules presented in [11], there should be added one rule which was inadvertently left out: "Forwarding is delayed by a time, T_{Dup} , when a duplicate is received (with hop counter greater than that of the original reception).", [12].

identifier. On the other hand, the not-innovative packets are those that are linear combinations of the packets present in the node's buffer, as in [13]. When NC-Dflood is applied, each node in the network follows the following rules:

- When a *source* node has g packets to send, it encodes them into h output packets, and sends them down to the MAC layer with hop count equal to one.
- Each time a *relay node* receives a packet of a generation for the first time, i.e. an innovative packet, it will forward that packet after a random back-off time uniformly drawn from $[T_{\min}, T_{\max}]$.
 - If during this time another innovative packet arrives, the relay node will encode these two packets in two other packets. One of them will replace the packet ready to be transmitted, while the second will be forwarded after the relative back-off time, still uniformly drawn from $[T_{\min}, T_{\max}]$.
 - If the back-off time expires without receiving an innovative packet, the relay will forward the packet with the same encoding vector as received. In this way, the end-to-end delay is kept limited. However, the packet will be conserved in order to do the recursive encoding process for the other innovative packets of that generation that may arrive later.
- For each innovative packet received at a *relay node*, with hop count lower than or equal to H_{\max} , another encoded output packet will be created. This new packet will have a hop count increased by one with respect to the previous received packet. The other packets present in the buffer, not forwarded yet, will be updated considering also this packet in the encoding process. H_{\max} is the maximum number of hops that a packet can travel in the network.
- If a *relay node* receives an innovative packet with hop count higher than H_{\max} , it will be considered for updating the encoding packets present in the buffer, and a new packet will not be created.
- The *relay node* will treat as duplicates all the not-innovative packets with the same identifier, regardless of the value of the hop count they have.
- The *relay node*, after forwarding g packets of a generation, will not take part in the relay process for the packets with that unique identifier.
- All received duplicates at a *relay node*, before forwarding all the packets of that generation, will be counted. n_d is the number of duplicates received.
- For each duplicate received, the *relay node* will delay the forwarding by T_{dupl} . In the NC-Dflood case, we have different possibilities to delay the forwarding of the packets, where depending on the specific scenario, one solution can be preferred over the other. We propose two approaches:
 - Delay by T_{dupl} only the first packet to be sent, with the same identifier of the duplicate received.
 - Delay by T_{dupl} all the packets ready to be sent, with the same identifier of the duplicate received.
- Each time a duplicate is received, before forwarding g packets of that generation, the *relay node* will draw a random number $r \in (0, g]$. If $n_d > N_{\text{dupl}} - r$, then the forwarding of the packets

of that generation will be discarded. $N_{\text{dupl}} = c \times g$ is the maximum number of duplicates, and c is a constant value.

- When the *destination node* receives g innovative packets with the same identifier, it will immediately broadcast a "Receive Notification" (RN) message containing the unique identifier of the completed generation. This can be received by all its neighbors, but will never be forwarded.
- When a *relay node* receives an RN, it will discard any forwarding of the packets related to that generation.

Even though NC-Dflood aims to reduce the number of duplicates that are flooded in the network, i.e. the energy consumption, it may still be too high for some scenarios. In order to reduce the number of duplicates even more, we propose to extend the proposed protocol with the node's position information. Our proposal, named NC-Geographical Dflood (NC-GDflood), uses the position information of the node that is transmitting and the final destination position. Using this information during the relay, the relays that are farther from the destination can be avoided.

Each node in the network can guarantee a certain transmission performance, i.e. a fixed bit error rate, with a fixed transmission power to cover a distance d . Each time a node has a packet to send, it first calculates the distance D between itself and the sink. Then, it quantizes this distance in hop counts, $D_{\text{HC}} = \lceil D/d \rceil$, where $\lceil \cdot \rceil$ indicates the ceil operator. Each node puts the value ' $D_{\text{HC}} + f_n$ ' in the hop count field at the moment the packet is transmitted, where $f_n \leq 0$ is a factor that reduces the number of relays that can consider the packet. Only the source nodes consider a redundancy factor, $f_s \geq 0$, for their own packets, and the hop count value of these packets when they are sent for the first time by the source node is ' $D_{\text{HC}} + f_n + f_s$ '. This redundancy factor increases the robustness and connectivity in the first-hop transmission, allowing more nodes to participate. With the above considerations, each *relay node* in the network, for each packet received, applies the duplicate procedure if the packet is not innovative, and uses it for re-encoding the actual packets ready to be sent if the received packet is an innovative one. But the nodes will create another packet to be sent only if the local D_{HC} will be lower than or equal to the hop count value contained in the received innovative packet.

3. SIMULATION RESULTS

For our simulations, we have considered the network topology in Figure 1. The network is composed of 22 bottom nodes and 1 AUV. The intra-node distance d is 3 km. The red line shows the trajectory of the AUV, which makes a round trip starting from checkpoint A to B (CP.A and CP.B in the figure) with a speed of 4 knots. We have considered 3 source nodes: the bottom nodes 1 and 10, and the AUV, and the packets are intended for node 22, i.e. the access point (AP). We have compared our proposal with the Dflood protocol. In the latter case, the application layer of the source nodes generates the packets with a constant bit rate for all of them. The arrival times of the packets are generated from a Poisson process with the λ parameter defined as $\lambda = L_b / \text{bps}$, where L_b is assumed to be 160 bits and bps (bits per second) is a simulation parameter that influences the traffic introduced in the network by the application layer of the source nodes. When NC is applied, we use a Poisson process to model the arrival times of a stock of g packets, with $\lambda = (L_b \cdot g) / \text{bps}$. With respect to the Dflood traffic model, when NC is used each source node waits to have g packets before sending them down to the network layer.

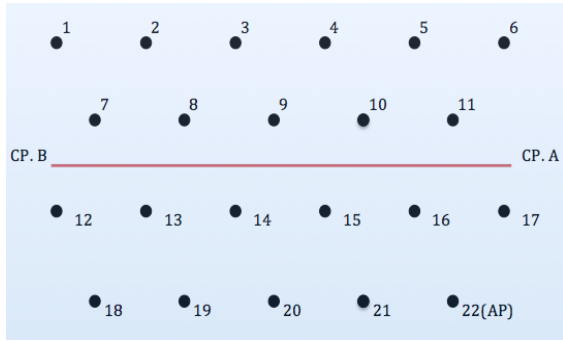


Figure 1. Network topology used in the simulations, composed of 22 fixed nodes and one AUV. For simplicity, we have considered a regular grid with intra-node distance $d = 3$ km.

In our simulations, we have assumed a PER, p , common to all the links present in the network. This parameter includes all the physical-layer errors due to noise, fading, Doppler effects and other link-loss phenomena. The transmission data rate in the physical layer is considered in such a way that the packet duration is one second. This assumption is considered for simplicity. Nevertheless, it is justified in underwater acoustic (UW-A) communications, since the packet lengths are very short and the bit rates are low, e.g. in [11] the authors consider packet lengths of 160 bits and a transmission rate of 200 bits/s for the Dflood simulations. All the nodes in the network are equipped with isotropic antennas with a transmission power such as to offer a PER = p for links up to 3 km. Farther nodes receive the packet erroneously with probability one. In the MAC layer, we have considered an unslotted ALOHA protocol, without collision avoidance, link or end-to-end acknowledgments. With the assumption that the adopted MAC protocol does not use any interference avoidance precautions, we have used a simple interference model for our scenario. If we consider a sound speed of 1500 m/s underwater and the intra-node distances present in the scenario, then the packet will take some seconds to be fully received by the destination side of the link. We have assumed a total destructive interference if two or more neighboring nodes are transmitting during this time. These nodes will not receive any packet from their neighbors, which were transmitting during this time interval. Furthermore, also the nodes that are neighbors with two or more transmitting nodes will not receive any of these packets. With the interference model implemented, in the NC case, we assume that the network layer sends the packets down to the MAC layer with a back-off time uniformly drawn from [1 s, 10 s]. This is done with the goal to avoid the interference that may affect all the packets of the same generation, if they are transmitted in a row.

In order to see the performance of the protocols, we will express our results in terms of PDR, end-to-end delay, and average number of packets forwarded by the network (Av. PKT), for each information packet produced by the source nodes. The Dflood parameters used are: $T_{\min} = 0$ s, $T_{\max} = 50$ s, $T_{\text{dupl}} = 35$ s and $N_{\text{dupl}} = 2$. For the NC-Dflood protocols, we have considered the following parameters, $T_{\min} = 0$ s, $T_{\max} = 70$ s, $T_{\text{dupl}} = 30$ s, added only to the first packet ready to be sent, $N_{\text{dupl}} = 2.5 \times g$, common to all the nodes in the network, $H_{\max} = 15$, a generation size $g = 2$, and $h = 3$. In NC-GDflood, an inaccuracy is considered when measuring the distance D . We have assumed $D = D^{\text{real}} + u$, where u is uniformly distributed as $u \sim U[-100 \text{ m}, 100 \text{ m}]$, and D^{real} is the real distance in meters between the node that is transmitting

and the AP. The parameters f_n and f_s are 0 and 1, respectively. For both protocols, the parameters are selected as a trade-off between the three evaluation criteria, paying more attention to the PDR. For NC-GDflood, we used the same parameters as for NC-Dflood, since we consider it as an upgrade of the latter. The selection of these parameters in both protocols is done by a heuristic approach. After many simulations, we may conclude that the boundaries of the interval $[T_{\min}, T_{\max}]$ influence the interferences in the network, as expected. Regarding their respective value, we may have more or less interference. Meanwhile, N_{dupl} , T_{dupl} , and H_{\max} influence more the number of packets forwarded in the network. A higher value of N_{dupl} brings us to a pure flooding mechanism, and the consideration of T_{dupl} tries to allow the nodes to collect more duplicates before forwarding the packet or reaching the N_{dupl} value. We have selected a generation size of $g = 2$ because of a few reasons. First and foremost, to keep the overhead in the packet header limited, since for each packet it increases by 8 bits. Second, we have seen that with an increasing generation size, the end-to-end delay and the Av. PKT increases as well.

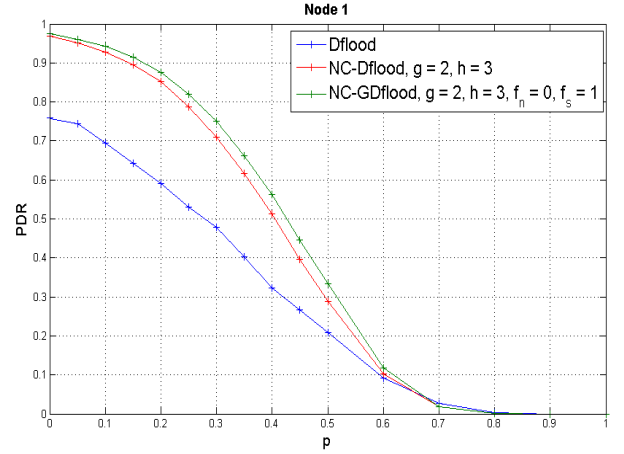


Figure 2. PDR vs. p for node 1's transmission.

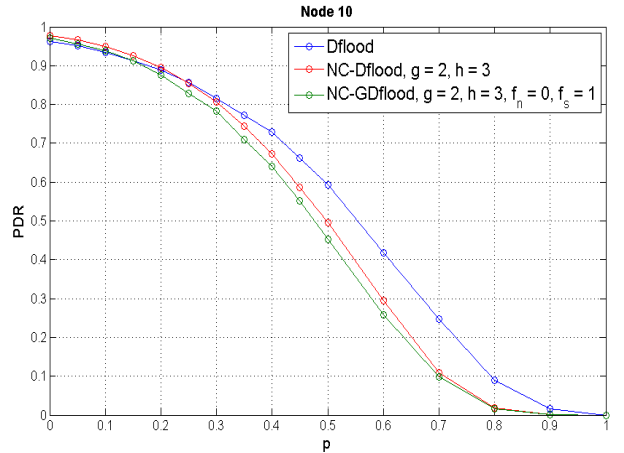


Figure 3. PDR vs. p for node 10's transmission.

We have simulated both protocols for different values of p and bps. The first simulations are done to evaluate the robustness of the protocols (Figures 2-6). In this case, we have assumed a bps = 1 bit per second for all the source nodes. Meanwhile, the simulations done for different values of bps are done with the goal to check how the performance changes when the network

becomes overloaded. In this case, p is selected 0.1 (Figures 7-11). The simulation results have shown that the NC approach achieves the goal to increase the transmission robustness, with respect to the PDR. The price to pay in this case is an increment of the end-

to-end delay and Av. PKT. The upgrade with position information reduces the number of packets forwarded, conserving the PDR and the end-to-end delay of the NC-Dflood.

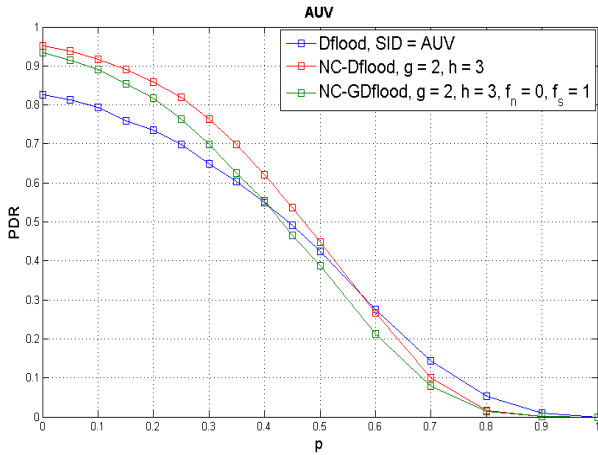


Figure 4. PDR vs. p for the AUV's transmission.

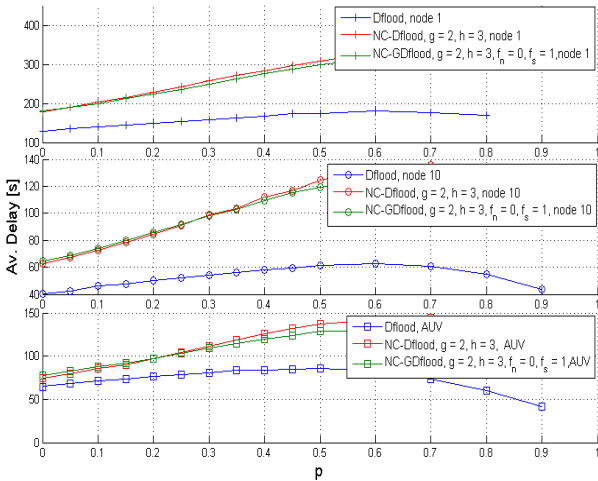


Figure 5. End-to-end delay vs. p of all the transmissions for all the protocols, and for different values of the link PERs.

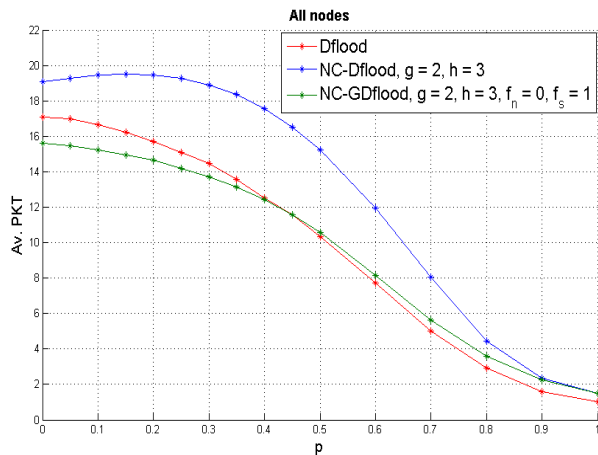


Figure 6. Average number of packets forwarded vs. p for each information packet produced by the source nodes.

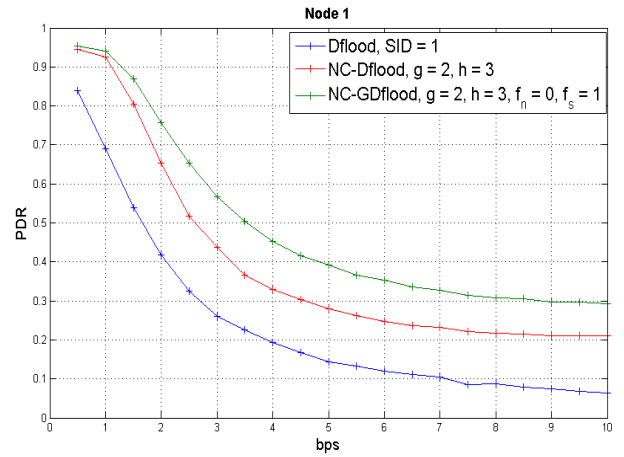


Figure 7. PDR vs. bps for node 1's transmission.

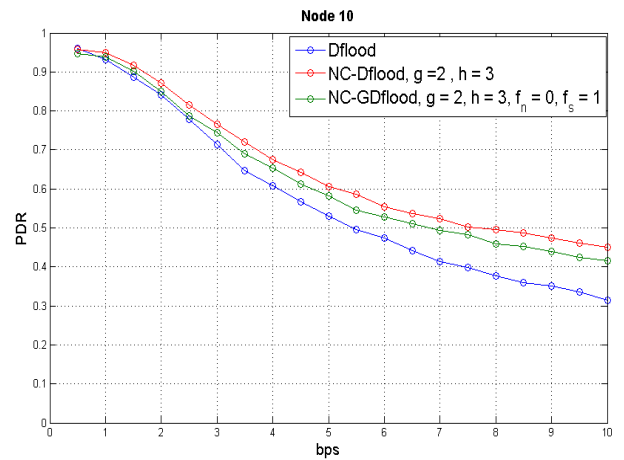


Figure 8. PDR vs. bps for node 10's transmission.

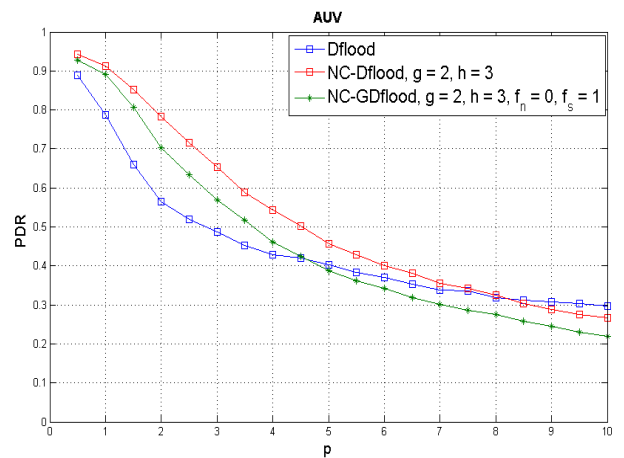


Figure 9. PDR vs. bps for the AUV's transmission.

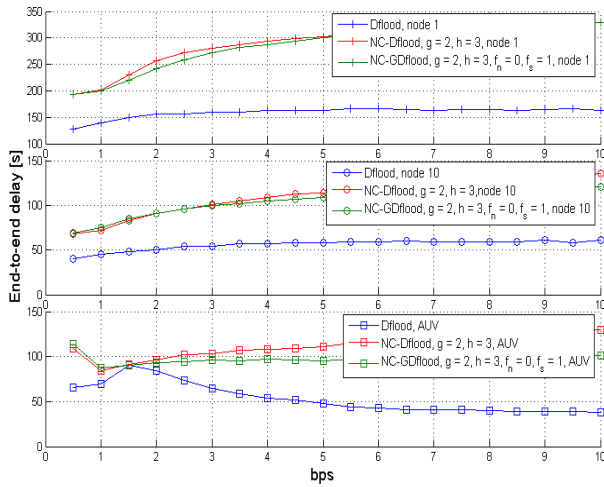


Figure 10. End-to-end delays of all the transmissions for all the protocols, for different values of the traffic rate.

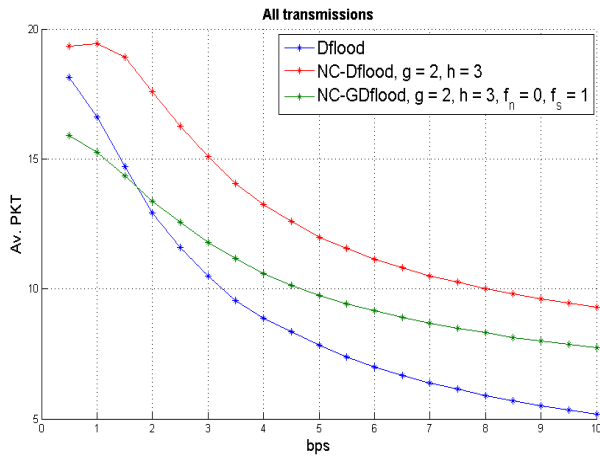


Figure 11. Average number of packets forwarded vs. bps for each information packet produced by the source nodes.

4. Conclusions

In this work, we have proposed a novel approach on how to use network coding in underwater communications. Our proposal consists of a different usage of the duplicates forwarded in the network by flooding-based protocols. Considering the high number of packets that are forwarded by the flooding-based protocols, if they share and carry information about the original packets produced by the source nodes, then the system performance increases. Our proposal, NC-Dflood, increases the PDR of the transmission, with an increment of the end-to-end delay and the number of duplicates in singular transmissions. Better results are obtained for nodes that are relatively far from the destination, for which the number of relays that can apply network coding is larger. In order to reduce the number of duplicates forwarded in the network, we propose to extend the NC-Dflood idea with position information of the node that is transmitting and the final destination position. In this way, mainly nodes in the direction of the destination will participate in the routing process, by encoding and forwarding the packets. We have shown that this upgrade, called NC-GDflood, maintains the same PDR and end-to-end delay of NC-Dflood, but it gains in terms of the number of duplicates. For low link PERs and low traffic, being characteristics of most underwater sensor networks,

it can reduce the number of packets even below those of the Dflood protocol.

5. Acknowledgements

The presented research has been performed in the framework of the European Defense Agency (EDA) project RACUN (Robust Acoustic Communications in Underwater Networks), sponsored by the ministries of defense of The Netherlands, Italy, Norway, Sweden and Germany.

We kindly acknowledge the support of Paolo Casari (University of Padova, Italy) and of Roald Otnes (FFI, Norway).

6. REFERENCES

- [1] Akyildiz, I., F., Pompili, D., and Melodia, T. 2005. "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks* (Elsevier), vol. 3, no. 3, May 2005.
- [2] Otnes, R. et al. 2012. Underwater Acoustic Networking Techniques. *Springer Briefs in Electrical and Computer Engineering*, DOI: 10.1007/978-3-642-25224-2_1.
- [3] Stojanovic, M., Preisig, J. C. 2009. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE Communication Magazine*, Jan. 2009.
- [4] Urlick, R. 1983. Principles of Underwater Sound. *McGraw-Hill*.
- [5] Sozer, E. M., Stojanovic, M., and Proakis, J. G. 2000. Underwater Acoustic Networks. *IEEE Journal of Oceanic Engineering*, Vol 25, No. 1. Jan. 2000.
- [6] Casari, P., and Zorzi, M. 2011. Protocol design issues in underwater acoustic networks. *Computer Communications*. June 2011.
- [7] Ahlswede, R., Cai, N., Li, S. R., and Yeng, R. W. 2000. Network information flow. *IEEE Transactions on Information Theory*. July 2000.
- [8] Guo, Z., Xie, P., Cui, J. H, Wang, B. 2006. On applying Network Coding to Underwater Sensor Networks. *WUWNET'06*. Sept. 2006. LA California, USA.
- [9] Lucani, D. E., Médard, M., and Stojanovic, M. 2007. Network Coding Schemes for Underwater Networks. *WUWNET'07*. Sept. 2007. Montreal, Quebec, Canada.
- [10] Chirdchoo, N., Chitre, M., and Soh, W. S. 2010. A Study on Network Coding in Underwater Networks. *IEEE*. 2010.
- [11] Otnes, R., and Haavik, S. 2013. Duplicate reduction with adaptive backoff for a flooding-based underwater network protocol. *In Proc. MTS/IEEE OCEANS*. Bergen, Norway. June 2013.
- [12] Tapparello, C. et al. 2013. Performance evaluation of forwarding protocols for the RACUN network. *WUWNET'13*. Nov. 2013. Kaohsiung, Taiwan
- [13] Fragouli, C., Le Boudec, J. Y., and Widmer, J. 2005. Network Coding: An Instant Primer. *LCA-Report-2005-010*. Available online from infoscience.epfl.ch.
- [14] Ho, T., Koetter, R., Médard, M., Krager, D. R., and Effros, M. 2003. The benefits of coding over routing in a randomized setting. *ISIT*. July 2003.
- [15] Chou, P. A., Wu, Y., and Jain, K. 2003. Practical Network Coding. *In Proc. Allerton*. October 2003.