

TenDoc: Network Coding-based Software for Wireless Ad hoc Networks

David Lim, Stéphane Rousseau, Farid Benbadis, Damien Lavaux
 Thales Group, Colombes, France
 david.lim@fr.thalesgroup.com
 stephane.rousseau@fr.thalesgroup.com
 farid.benbadis@fr.thalesgroup.com
 damien.lavaux@fr.thalesgroup.com

Abstract—This work builds upon previous studies that highlight the benefits of Network Coding for all to all traffic patterns. This is the case within the framework of an optimized link state routing protocol (OLSR) where topology control messages are flooded in the entire network in order to provide a precise knowledge of the topology to all single nodes in the network. The flooding efficiency is often measured either in terms of successful dissemination delay or the number of transmissions needed to achieve this goal. In this paper we present the new TenDoc software that has been first designed to support any applications and that has been adapted especially for this OLSR use case. After recalling our own previous simulation-based investigations, we detail the architecture of this TenDoc software and finally present results from our results from our experimental platform.

Keywords - network coding; wireless network; topology dissemination; multi-point relays.

I. INTRODUCTION

A Public Safety Network (PSN) is a multi-hop wireless Network specially used for emergency service organizations (e.g., police, fire services). The ease of deployment and self-management features make such networks very attractive and useful during emergency interventions in crisis areas where any other means of communications are often no longer available (e.g., earthquake, flood). Due to the multi-hop nature of Public Safety Networks, the design of routing protocols is central to optimizing network capacity and usage efficiency. In the literature, two kinds of routing protocols emerge: i) reactive protocols where routes are computed only when needed, and specially adapted for; ii) pro-active protocols where routing tables are maintained. The former are well suited for Mobile Ad-Hoc Networks while the later are tailored to infrastructure-like networks such as Mesh networks [1]. Public Safety Networks, where the time for communication establishment is critical, clearly belong to the second category.

One of the most used pro-active protocols is OLSR [6]. OLSR protocol consists of two distinct parts: i) local topology gathering and dissemination; ii) routing table updates based on the topology deduced from the gathered topology information. Within the framework of this study, we focus on the first part and examine how topology information is disseminated.

In OLSR, local topology information is collected by each node and aims at evaluating the link quality between each node and its neighbors. The protocol is based on bi-directional exchange of messages, called *Hello messages*. The link quality is estimated by taking into account both the successful transmission and reception of those messages. Once this local information is gathered, it must to be disseminated to all nodes in the network by *Topology Control messages (TC)*. A naïve approach consists in simply flooding these messages to the entire network. In [6] authors present a distributed Connected Dominating Set algorithm that reduces dramatically the amount of needed transmissions to achieve a successful dissemination. In contrary to flooding dissemination wherein all nodes forward all messages, the distributed Connected Dominating Set algorithm consists in selecting a subset of nodes in charge of forwarding. This algorithm significantly outperforms the flooding dissemination [5].

Optimizing the topology information dissemination based on TC forwarding within OLSR has been the subject of many studies [2], [3], [4]. The most recent ones aim at reducing the number of transmissions needed to achieve a successful dissemination (when all nodes have the knowledge of the entire topology) by introducing Network Coding techniques. The advantage of this method lies in the transmission. Indeed it does not just relay messages to other nodes but combines several messages together before transmitting them. In this way, we have more information in each message.

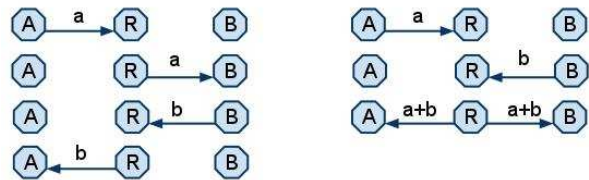


Figure 1. This figure depicts an application example of network coding technique onto three nodes, A, R, B. A resp. B has to send one packet towards B resp. A via R. On the left, without network coding; On the right with network coding. The benefits of using network coding in this case is 25%.

In figure 1, we consider 3 nodes A, B and R (the relay node). In the normal network context, only one packet may be transmitted at a time. However, with the network coding

technique, once A and B have transmitted, the relay node can broadcast the coded information to both nodes, thus transmitting two packets simultaneously.

In a preliminary study, several message dissemination protocols (MPR-based forwarding, pure flooding, dominant pruning and network coding) were for the first time compared. As a main result, network coding was shown to outperform other protocols in terms of number of transmissions [2].

In this paper, we extend our previous study by implementing the concept proposed in [2] and run an experiment within our 7 node lab test-bed described on Figure 3. This software implementation enables the use of network coding for topology information dissemination in OLSR. This software is fully technology independent and does not require any modification within OLSR. Furthermore, this software can be used for other applications such as sensing dissemination in Cognitive Radio Networks or Common Operational Picture applications in the context of PSNs. The main contributions are twofold:

- The software architecture for network coding for all to all traffic patterns.
- Software design for the special case of OLSR.

This paper is organized as follows: we first present the related work that motivates this software development, then we detail the architecture of this software, and finally, we present the first experimental results of this novel approach conducted in a 7 nodes wireless test-bed.

II. RELATED WORK

Recent studies have been done in the framework of TC message dissemination improvement by using Network Coding. In [4], Kadi and Al Agha evaluated the benefits to use network coding for TC messages dissemination in the context of OLSR [6]. They focus on the determinist network coding that consists in combining messages in order to maximize at each step the number of neighbor nodes that will be able to decode. The benefits of such methods are also demonstrated in CODEB in [8]. However, an additional information exchange protocol is required to inform neighbor of each node of the set of TC messages already collected. Using simulations, the authors show that network coding applied to MPR-based dissemination [7] significantly reduces the number of transmitted TC messages. In [3], Kadi and Al Agha proposed an additional study by using Random Network Coding which consists in combining messages randomly without any knowledge of the set of TC messages already collected by the neighborhood. Once again, results are dramatically better when dissemination combines MPR-tree and Random Network Coding. Relying on those first results, in [2], an overview of dissemination techniques is done either based on Connected Dominating Set algorithms or network coding (Determinist, Random). All relevant previous works and novel combinations are investigated and compared with each other. Finally, performance gains assessed by simulations show that network coding for TC message dissemination can improve the efficiency in terms of delay of information delivery and of the number of transmissions needed.

III. TENDOC ARCHITECTURE DESIGN

We first describe the TenDoc modules that compose its architecture and then modules interactions. They are illustrated by the different steps in the case of OLSR application. Although the TenDoc software is designed for the OLSR protocol, it can also be used for other applications.

A. Architecture description

In this section we present the modules that compose TenDoc software.

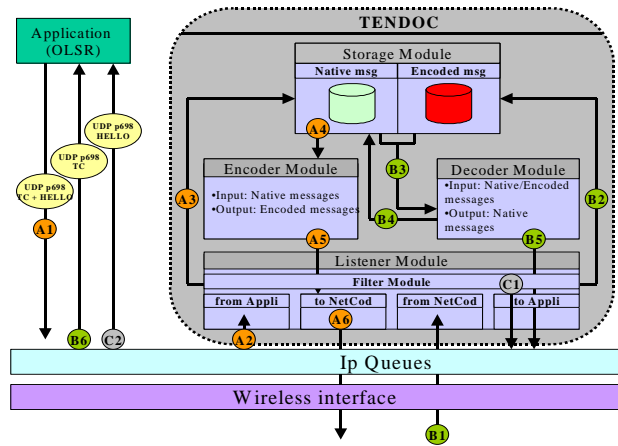


Figure 2. The software architecture.

1) Listener Module

This module aims at fetching TC messages from the IP queue, either from the local OLSR application or from the wireless interface. It is composed of sub-modules that interface with the Linux IP queues at a kernel/user space level and 1 sub-module in charge of dispatching fetched messages.

- **‘From appli’** listens to UDP traffic on port 698 that is generated locally. Both TC and Hello messages are sent on this port.
- **‘To appli’** sends native messages that are extracted by decoded module towards OLSR applications.
- **‘From TenDoc’** listens UDP traffic on an unassigned port (1024 for example), forwards it to be stored into encoded message buffer in the Storage Module.
- **‘To TenDoc’** sends encoded messages to neighbor nodes within the radio coverage area, by using a UDP broadcast socket opened on port 1024.
- **‘Filter’** identifies and dispatches messages to the proper sub-module. In the context of OLSR, TC and Hello messages are sent on the same port using the protocol. It is necessary to identify TC from Hello and redirect Hello messages towards their defined destinations.

2) Storage Module

This module is composed of two buffers. The first one aims at storing native TC messages that have been either created by the local OLSR daemon or received from neighbor nodes. The second buffer, Encoded message buffer, stores all messages that are not decoded at this time, either because needed native messages have not been received yet or the decoder module does not treat them at this step.

3) Encoder Module

The goal of this module is to encode two TC messages together from *native message buffer* by xoring two messages contained in the storage module into encoded messages to be sent. The number of encoded messages is a parameter which can be tuned for the experiment. Then, the created encoded messages are sent to the listener module.

4) Decoder Module

This module regularly checks if some encoded messages stored in the *encoded message buffer* (storage module) could be decoded by using native messages contained in the *native message buffer* (storage module). If this is possible, the decoded module removes the considered encoded message from the *encoded message buffer* and stores the native messages into the native message buffer. Moreover, those native messages are also sent towards the OLSR application via the listener module.

5) Module interactions

To emphasize module interactions, all single steps of OLSR TC message dissemination are enumerated, by starting with the first sending of one TC message from the local daemon of OLSR towards the IP queues. Neighbor node TenDoc software exchanges are presented and finally the forwarding of recovery TC messages from encoded messages towards OLSR applications. As illustrated in Figure 2, those steps are ordered into three groups: the A_x steps describe the treatment of TC messages from the local OLSR application, the B_x steps describe the exchanges between TenDoc software running onto neighbor nodes and finally, the C_x steps deal with unexpected fetched messages treatment.

Herein, we present those steps in details:

- **A1:** OLSR sends TC and Hello messages by using broadcast UDP traffic on port 698.
- **A2:** Before sending, the Listener Module fetches those messages.
- **A3:** TC messages are filtered, extracted and sent towards the Storage Module to be stored into *native message buffer*.
- **A4:** Encoder Module takes some native messages from the *native message buffer* and encodes them. An encoded message is created, with a header indicating the number and the sequence number of the TC messages encoded and within the payload the result of the encoding function.
- **A5:** The encoded message is sent towards the Listener Module.
- **A6:** The Encoded message is broadcast towards all neighbor nodes by using a UDP socket on port 1024.

This is the conclusion of the first step of the TenDoc software. The second step begins when an encoded message is received from a neighbor node.

- **B1:** The Listener Module gets all messages from port 1024 that have been sent as UDP traffic.
- **B2:** Encoded messages are identified and sent to the storage Module to be stored into the *encoded message buffer*.
- **B3:** The Decoder Module decodes encoded messages from *encoded message buffer* by using native messages from *native message buffer*.
- **B4:** Native messages decoded are stored into *native message buffer*.
- **B5:** A copy of the native messages decoded is sent towards Listener Module.
- **B6:** Local OLSR application receives TC messages as a UDP traffic on port 698 that makes this software fully seamless from the application point of view.

This ends the second steps of the TenDoc software process. The last one is mainly designed for an OLSR application that sends TC and Hello messages onto the same port number that makes a special filter necessary to distinguish TC from Hello messages before treatment. The following steps deal with the Hello messages once they have been fetched by the 'from appli' sub-module within the Listener Module.

- **C1:** The Filter sub module extracts Hello messages from fetched messages and sends them towards neighbor nodes
- **C2:** The local OLSR application receives Hello messages from neighbor nodes as UDP traffic on port 698.

As mentioned before, OLSR is only one of many possible applications of the TenDoc software. The next section deals with the experiment that will be conducted to prove the software concept and assess the performance gains.

IV. EXPERIMENTAL ENVIRONMENT

A. Test-bed description

Seven nodes equipped with wireless interfaces (Atheros wireless Card) are deployed to form the topology depicted in Figure 3. A version of Ubuntu is running on each node and the interface is configured in Ad Hoc mode on channel 2 (channel that is used by no other nodes in the radio coverage range). On each node, the OLSR daemon is running. We use OLSR Version 0.5.6.

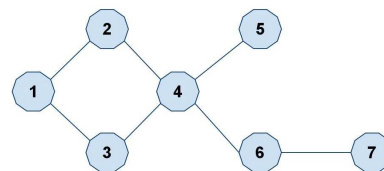


Figure 3. Our 7 node indoor test-bed topology.

B. Scenario description

To evaluate the performance gains of using TenDoc instead of OLSR, we focus on two scenarios: (i) standard OLSR running alone, and (ii) TenDoc software running on all nodes of the platform. In both cases we measure the number of transmissions during a 1 day period.

In order to measure this information, we use “iptables” Linux command that can keep track of how many messages have been sent on an interface. We complete this information by using the wireshark software to scan traffic generated within an area (i.e., the platform area).

V. RESULT ANALYSIS

First, we present last year’s simulation results. Then we introduce the new data from our experiment. For our test, we wanted to consider the number of messages and the delay of a successful dissemination. The two following figures provide us evidence about the advantages of network coding.

A. Preliminary Results

Figure 4 shows the comparison of the message dissemination in the network between MPR-based and network coding based method.

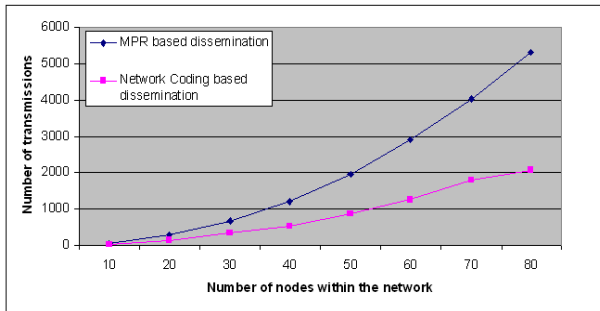


Figure 4. Number of transmissions of MPR based compared to the network coding dissemination

The first thing that we can notice is the number of transmissions is significantly lesser than the MPR based with almost a gain of fifty percent. The result is expected because the network coding process combines packets together and broadcast them. Therefore more information is transmitted per broadcast. Furthermore, the gain depends on the number of the nodes.

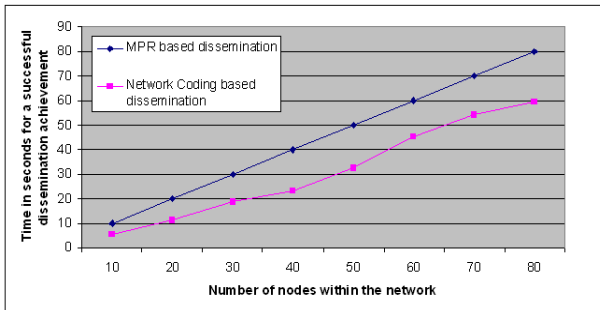


Figure 5. Total time achievement of dissemination of MPR based compared to the network coding.

In figure 4, the noticeable gap starts at 20 nodes. This result indicates that the density of the mesh network should be considered. In figure 5, we consider the time of dissemination achievement. We can see the network coding has an advantage; the completion time is lesser than the MPR-based network. In term of delay, the result is almost the same, but the number of packets which pass through the network is lesser to have a total dissemination.

B. Experiment results

In this section, we compare MPR based and network coding based dissemination of TC messages. This experiment is conducted on our 7 indoor test-bed nodes –see Figure 3. The MPR based dissemination requires 23 transmissions for a successful dissemination.

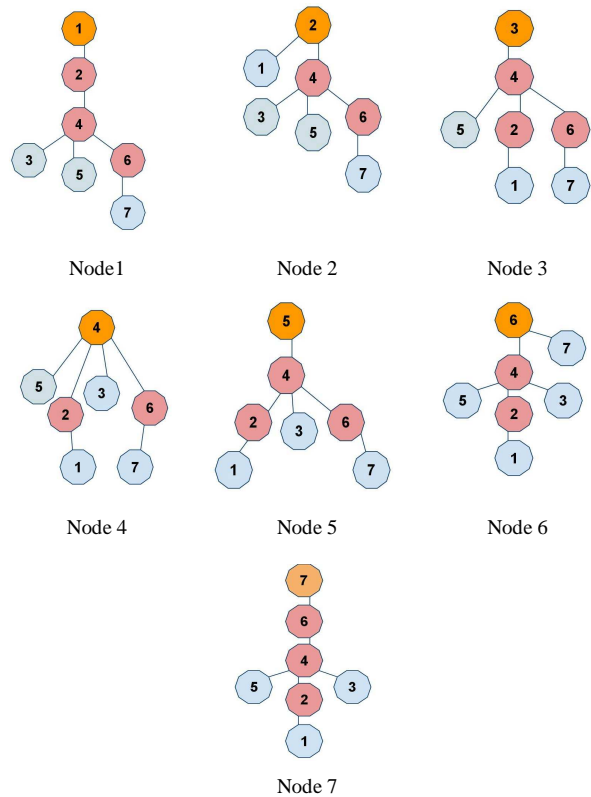


Figure 6. MPR diffusion tree.

We conduct this experiment by calculating the number of transmissions from the diffusion tree. We consider that each node has to diffuse its own information. Therefore, we select each node one by one to calculate the number of packets which circulate in the network. In figure 6, the root node is the orange node, the red ones are the MPR nodes and the blue one is the terminal node. We calculate the number of the messages that each node disseminates in the network by counting the number of hops from each root node to the terminal nodes.

Table I shows the comparison of all methods that we have used. As we can see the pure flooding method is high. With MPR-based method there were 23 messages which

were disseminated in the network. When we use TenDoc software, it combines the pure flooding and the network coding method. We did not expect these results in the practical way. The number of the transmitted messages is higher than for pure flooding.

TABLE I. CLASSIFICATION OF DIFFUSION METHODS

Methods	Number of transmissions
<i>PF</i>	49
<i>MPR based</i>	23
<i>PF+NC</i>	53

As we observed before, in the simulation result the number of messages that each node diffuses depends also on the mesh network density. The experiment that we conducted with seven nodes was not very significant. We would make some other experiments with more than fifty nodes to observe the same performance gains as those obtained in our previous work [2].

VI. SUMMARY

In this paper, we introduced the implementation of network coding in the OLSR protocol for minimizing the number of required transmissions. This was a natural combination. Indeed OLSR is the standard for ad-hoc routing in mesh networks, while the network coding concept is a really efficient way to optimize the number of transmissions and the use of the rare wireless spectrum.

We also investigate on the TC message dissemination by using the Network coding. Moreover we wanted to be application independent (without any modifications onto OLSR). Finally, this method could optimize the radio resource usage.

In this paper, first we introduce the network coding and the OLSR protocol in the ad-hoc mesh network which is the way to optimizing the message dissemination in the mesh. The network coding is added to the routing protocol to minimize the number of required messages. We go beyond the state of art of the network coding to propose a practical solution to implement the application. In the theoretical approach [2], when we associated OLSR and network coding, the simulation results showed that network coding could improve the performance of the message dissemination by fifty percent, thereby avoiding avoiding the waste of radio resources.

Furthermore, to implement our solution, we strive to be seamless from the application point of view and develop our solution as a module that we can plug or unplug whenever we want. It connects to the OLSR framework. In this solution we have two levels: the connection between the OLSR and the inter-module connection. The first one we use the OLSR port connection the 698 to get the packet of TC and hello message by listening the port. We have created a listener and the sender for communicate with the OLSR the packet is transferred to the second level to send in the

network through another port connection. Basically we do not use the OLSR port connection to communicate with another nodes but TenDoc's port. The second level is in the part where we have the network coding treatment for encoding and decoding the message that we want to send. Basically, only random network coding is available but we plan to enrich this software by integrating deterministic one.

As we can see, the results are not very significant with a study of seven nodes dissemination. And we think about further work to increase the number of nodes to have a high density of a mesh network. Because previously in [2] in the theoretical way, we saw the density was very important and it can transfer more messages at each time. And the implementation in a practical solution could improve the radio communication by introducing this concept of message dissemination.

Finally, this software will be combined with an efficient control plane that will enable to activate or not some nodes to be network coding active in order to optimize network capacity usage.

ACKNOWLEDGEMENT

This work was supported in part by the European Commission through ICT project CONECT (Cooperative Networking for High Capacity Transport Architectures), FP7 ICT-2009.1.6, and by CELTIC Office through project HNPS (Heterogeneous Network for European Public Safety), CP5-010.

REFERENCES

- [1] S. Jan, I.A. Shah, H.S. Al-Raweshidy, "Performance Analysis of Proactive and Reactive Routing Protocols for Mobile Ad-hoc Grid in e-health Applications," Communication Software and Networks, 2009. ICCSN '09. International Conference on, pp.484-488, 27-28, Feb. 2009.
- [2] S. Rousseau, F. Benbadis, D. Lavaux, and L. San, Thales Communication France "Overview and optimization of flooding techniques in OLSR", HotMESH 2011, third IEEE International Workshop on Hot Topics in Mesh Networking, June 2011.
- [3] N. Kadi, K. Al agha, "MPR-based flooding with distributed fountain network coding," Ad Hoc Networking Workshop (Med-Hoc-Net), 2010 The 9th IFIP Annual Mediterranean, pp.1-5, 23-25, June 2010.
- [4] N. Kadi, K. Al agha, "Optimized MPR-based flooding in wireless ad hoc network using network coding," Wireless Days, 2008. WD '08. 1st IFIP, pp.1-5, 24-27, Nov. 2008.
- [5] A. Qayyum, L. Viennot, A. Laouiti, "Multipoint relaying for flooding broadcast messages in mobile wireless networks," System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on, pp. 3866-3875, 7-10 Jan. 2002.
- [6] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, L. Viennot, "Optimized link state routing protocol for ad hoc networks," Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International , pp. 62- 68, 2001.
- [7] P. Jacquet, Ed. T. Clausen, Ed. The Internet Engineering Task Force (IETF). "Optimized Link State Routing Protocol (OLSR)" Project Hipercom, in INRIA, October 2003.
- [8] L. Li, R. Ramjee, M. Buddhikot, S. Miller, "Network Coding-Based Broadcast in Mobile Ad-hoc Networks," INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE , pp.1739-1747, 6-12 May 2007.