Backpressure Approach for Bypassing Jamming Attacks in Wireless Sensor Networks

Amit Dvir
Laboratory of Cryptography and System Security (CrySyS)
Budapest University of Technology and Economics
Hungary
Email: azdvir@gmail.com

Levente Buttyan
Laboratory of Cryptography and System Security (CrySyS)
Budapest University of Technology and Economics
Hungary
Email: buttyan@crysys.hu

***Abstract*—The wireless medium used by sensor networks makes it easy for adversaries to launch jamming attacks that can block communication. In order to bypass the jamming area, tree-based routing protocols need to reconstruct the tree, a path or choosing new parent which is time consuming. In addition, bypassing congests the nodes at the border of the jamming area. In this paper, we present and implement a recovery algorithm based on a weighted backpressure function that bypasses the jamming area by spreading the congestion over a large subset of the sensor nodes, while no tree reconstruction and mapping of the jamming area are needed. As future work, we will implement and simulate our recovery algorithm using the IPv6 Routing Protocol for Low-power and Lossy Networks (RPL).**

I. Introduction

A wireless sensor network (WSN) consists of a large number of distributed sensor nodes, which are small, low cost, and low-power devices with limited capabilities of sensing, computing, mobility, and communicating. Tree-based routing protocols, which maintain routing information by means of tree structure, have been proposed for WSNs [1]. However, the existing protocols have two main shortcomings: long paths and the need to be reconstructed in case of a single link failure.

An adversary, broadcasting in the same frequency band as the network for long periods of time causes a jamming attack. Therefore, the other nodes experience low throughput by not being able to access the channel [2]. In case of tree-based routing protocols, a jamming attack could trigger the reconstruction of the tree (global repair), which increases message and energy overhead. Moreover, if the sensor nodes try to bypass the jamming area locally (local repair), congestion of the nodes at the border of the jamming area increases (see Fig. 1 A).

Dynamic Backpressure routing [3]–[5] does not perform any explicit path computation from source to destination. Instead, the routing and forwarding decision is made independently for each packet by computing for each outgoing link a backpressure function based on the localized queue and link state information.

In this paper, we present a new jamming recovery algorithm for WSNs based on a weighted backpressure function. The recovery algorithm will be part of the routing protocol and react when a node tries to bypass the jamming area. Our recovery algorithm allows for bypassing the jamming area without increasing the load (number of messages a node forwards) of the nodes at the border of the jamming area. This can be done by spreading the congestion over the sensor nodes (see Fig. 1 A).

II. Related Work & Our Contribution

The basic backpressure refers to techniques grounded in stochastic network optimization [3], [4], referred to as Utility Optimal Lyapunov Networking algorithms in a recent work by Neely [4]. Moeller et al. [3] presented the Backpressure Collection Protocol (BCP) for sensor networks. Dvir and Vasilakos [5] presented backpressure routing protocols for DTN.

Li et al. [2] presented network defense policies in WSN against jamming attack. They propose to [2] map the jamming area by transferring the attack notification message out of the jammed area.

Our recovery algorithm is based on weighted backpressure function with the following parameters: backlog queue (number of messages in the buffer); level; and neighbors' routing table. Moreover, our algorithm improves the WSN efficiency in terms of load (number of messages a node forwards), balance (number of nodes participating in the bypass procedure), recovery time (the time needed for a node, without a route to the destination, to find a new route), and energy. Essentially, the main advantages of the recovery algorithm are the following: it does not reconstruct a new tree such as other tree-based routing protocols, it does not map the jamming area and minimizes the recovery time to zero, while in the tree-based protocols even the recovery time of a node using local repair is not always zero.

III. Backpressure as Recovery Algorithm for WSN Routing Protocol

A node without a route will trigger the recovery algorithm. The recovery algorithm (Eq. 1) of node i is based on calculating for each neighbor j the weighted backpressure function (Eq. 2) and choosing the neighbor with the highest value. The backpressure function is based on: the difference L of the levels between the node and its candidate neighbor; the difference B between the node and its candidate neighbor's queues (backpressure); and the routing status R, which is

based on the routing table of the candidate neighbor. The motivation to use R is based on the assumption that a node will prefer a neighbor that have a new route (not creating a loop) to the destination or, at least a neighbor that did not see the message. Based on some experiments we set the values of R as following: for each neighbor j, if the neighbor j has a route to the destination, check if it is not creating a loop (next hop is i). If there is a loop, $R = 1$, if not, check if node j received this message in the last 3 hops. If received $R = 3$, if not $R = 5$. If node j does not have a route, and it is an old message $R = -2$, otherwise $R = -1$. One of our future goals is to develop a wise systematic approach for tuning parameter R. Note, in order to be able to calculate the above values, node i has to be able to exchange some information with node j. Therefore, we assume that neighbor nodes can exchange some information, periodically or per packet, before sending data packets. Moreover, the understanding of which message exchange procedure is more efficient will be part of our future work.

$$E_i = \max_j(E_{i,j}) \tag{1}$$

$$E_{i,j} = L_{i,j} + B_{i,j} + R_{i,j} \tag{2}$$

IV. Simulation

Our main future work is to test and simulate our recovery algorithm over RPL in Low power and Lossy networks (LLNs) [6]. Moreover, the ROLL working group [6] has identified that multipoint-to-point (MP2P) traffic, from sensors to sink (gateway node), is dominant among the several types of traffic encountered in LLNs [6]; therefore, we assume that the network traffic is MP2P.

We first implemented a comprehensive simulation environment based on the ONE simulator [7], (in our simulations, the network were always connected). In each scenario we simulate two protocols, the shortest path tree protocol as routing protocol with finding a new shortest path as a recovery algorithm (SPT) in case of jamming and the shortest path tree protocol as routing protocol with our new approach as a recovery algorithm (BP_Back) in case of jamming. In order to create effective jamming attack, we first simulate the network with the shortest path protocol without jamming (SPT_Per) and identify two nodes; the hub node and the degree node, where the hub node is the node that forwards the greatest number of messages and the degree node is the node with the largest number of neighbors. In each jamming scenario, we delete the hub/degree node. The protocols' performances are measured using two major evaluation methods: Load - number of messages a node forwarded in the simulation; Participating nodes - number of nodes that forwarded more/fewer messages compared to the case without jamming. In Figure 1 B, we see the nodes load while using the SPT_Per, SPT, BP_Back in case of typical run of the simulation. From the results, we can conclude that more nodes in the BP_Back (triangle) increase their load (participating in the bypass procedures) while this occurs in only a few cases of SPT (rectangle). The more nodes participating, the more the network become load

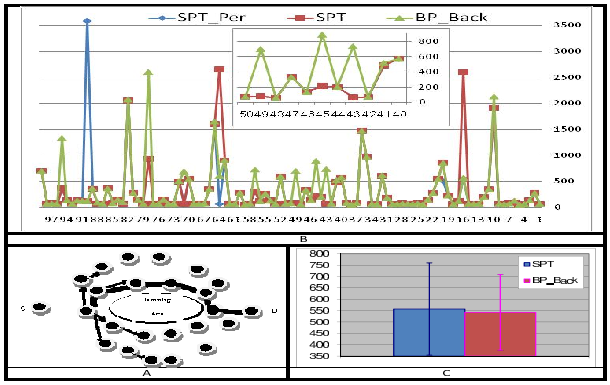

Fig. 1. (A) Backpressure Motivation, the thick line shows local bypassing using a tree-based routing protocol while the thin lines showx a recovery algorithm spreading the congestion over the sensor nodes. (B) An example of nodes load of typical run of the simulation, jamming attack on node 90, in the SPT only 6 nodes participating while in the BP_Back 25 nodes participating (small figure, only nodes $40 - 50$) in order to bypass the jamming area. (C) Average and Standard Deviation of standard deviation loads computed over 20 simulations, $Ave(SD_1, ..., SD_{20})$, $SD(SD_1, ..., SD_{20})$.

balanced. Figure 1 C show the average and standard deviation (SD) of the standard deviation loads computed over twenty simulations. From the results, one can see that the SD is much bigger while trying to bypass the jamming area with only the border's nodes (SPT), compared to the case of spreading the load over the network (BP_Back).

V. Conclusion

The contributions of this paper include adoption of a recovery algorithm based on a weighted backpressure function into tree-based routing protocols, in order to bypass jamming attacks in WSN. Moreover, our simulation results showed that combining our recovery algorithm with shortest path protocol improves the efficiency of WSN in terms of load, energy, and number of nodes participating. For future work, we will implement and simulate our recovery algorithm into the RPL proposal; investigate the advantage and disadvantages of our recovery algorithm compared to RPL local and global repairs [6]; and test the implantation on real sensor nodes, in order to modify the R parameters and neighbor messages exchange procedure. This research has received partial funding from the WSAN4CIP project (FP7/225186) and HUMAN-MB08-2.

References

[1] Y. Park and E.-S. Jung, "Plus-Tree: A Routing Protocol for Wireless Sensor Networks," in *ICHIT*, Jeju, Korea, Nov. 2006.
[2] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal Jamming Attacks and Network Defense Policies in Wireless Sensor Networks," in *INFOCOM*, May 2007.
[3] S. Moeller, A. Sridharan, B. Ksrishnamachari, and O. Gnawali, "Routing without routes: The backpressure collection protocol," in *IPSN*, 2010.
[4] M. J. Neely and R. Urgaonkar, "Opportunism, backpressure, and stochastic optimization with the wireless broadcast advantage," in *ACSSC*, Oct. 2008, pp. 30–41.
[5] A. Dvir and A. V. Vasilakos, "Backpressure-based routing protocol for DTNs," in *SIGCOMM*, Sep. 2010.
[6] IETF, "Routing Over Low power and Lossy networks (ROLL) - Working Group," http://datatracker.ietf.org/wg/roll/.
[7] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE Simulator for DTN Protocol Evaluation," in *STT*, March 2009.