Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 205 (2022) 127–136

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Military Communications and Information Systems (ICMCIS 2022)

Investigation of a GNN approach to mitigate congestion in a realistic MANET scenario

Yann Maret[a,b,1], Mohsin Raza[a], Franck Legendre[c], Junyuan Wang[d], Nik Bessis[a], Jean-Frederic Wagen[b]

[a] Edge Hill University, St Helens Rd, Ormskirk L39 4QP, UK
[b] University of Applied Sciences HES-SO, Bd de Pérolles 80, Fribourg 1700, Switzerland
[c] Armasuisse, Feuerwerkerstrasse 39 , Thun 3602, Switzerland
[d] Tongji University, Shanghai, China

Abstract

Mobile Ad-hoc Networks (MANETs) can be modelled as time-varying graphs as their topology and traffic demands change. Optimizing routing in MANETs by proactively adapting the routes is a challenge, even with a fixed topology and user demand. Omniscient Dijkstra Routing (ODRb) is one of the best known approaches, which computes alternative paths but has limitations to mitigate congestions. In this paper, we investigate Graph Neural Networks (GNNs) for routing optimization in MANETs. Our contribution is inspired by the centralized GNN-based Data Driven Routing (GDDR) framework developed by Hope [1]. GDDR was developed for optical fibre networks to support time varying user demands. After failing to obtain good results using the GDDR approach on the tactical Anglova MANET scenario, we adapted GDDR to minimize the maximum number of traversals. Our GNN-t proposal is able to find alternative longer paths mitigating congestion on central nodes. Considering a challenging static topology, the first second of the 24-node Anglova scenario: GNN-t achieves a Completion Ratio of CR=99% for a traffic of acked-messages averaging 1msg/s/node and CR=77% when the traffic is doubled (2msg/s/node). For the challenging first 300s of Anglova CP1, similar performance is reported for ODRb and GNN-t: CR=81% without fading and CR=54% with fading.

Keywords: Routing Protocol, Reinforcement Learning, Graph Neural Networks, MANET, EMANE, Network Emulation

1. Introduction

Mobile Ad-hoc Networks (MANETs) are well known type of networks for tactical and emergency scenarios. A MANET consists of a self-reorganizing and an infrastructure-less network. Routing remains a challenge in MANETs. A Popular routing protocols is the Optimized Link State Routing (OLSRd2) version 2 (RFC 7181)

*Corresponding author
E-mail address: yann.maret@hefr.ch.

10.1016/j.procs.2022.09.014

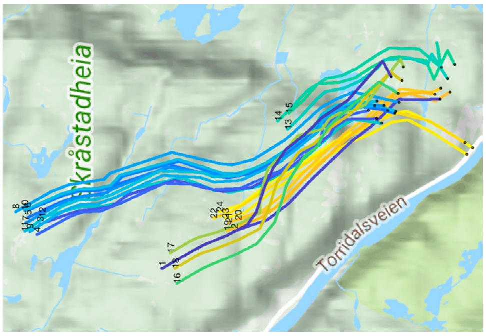

Fig. 1: The Anglova CP1 Scenario.

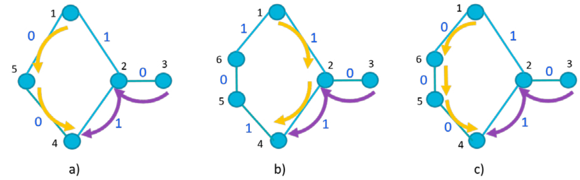

Fig. 2: Illustration of simple topologies with a) and c) showing the appropriate actions to be learned. The case b) shows inappropriate actions which should be penalized since node 2 will be congested.

protocol. One implementation: OLSRd2 [2, 3] is exploited in real scenarios [4] and is used here as a benchmark. Changes in the topology is challenging for routing algorithms. In tactical conditions, the link quality can drop abruptly leading to disastrous Quality of Experience (QoE). An efficient routing should trade-off between the link quality, the stability of the routes, and the congestion due to traffic. Poor performance in dynamic topology scenarios such as the realistic Anglova company 1 (CP1) scenario Figure 1 were reported in [5] using the well-known routing protocol OLSRd2 and compared to a so-called Omniscient Routing (ODRb) protocol [5]. Routes of OLSRd2 can be improved by reducing the Topology Control interval to avoid conserving routes with poor link quality [6]. Longer routes to avoid congestion can be implemented by changing the so-called LINKSPEED in OLSRd2. Changing the LINKSPEED or directly the routing tables to improve the performance remains a challenge.

Several Machine Learning (ML) solutions were proposed to predict the best routes and, possibly, reduce congestions, e.g., [7, 8, 9, 1]. In [7, 10, 6], three approaches using Q-learning were proposed to detect and try to avoid congested nodes. The underlying routing protocol, OLSR, was kept to benefit from the remarkable features including network discovery and dissemination of the information. Q-learning based solutions have known difficulties in learning a policy for high dimensional state and action spaces [11]. The Q-learning algorithm needs to visit many possible state-action in order to converge. Deep Neural Network (DNN) with approximation functions can mitigate such limitations [12].

New techniques have emerged in Reinforcement Learning (RL) to train a model from a graph: Graph Neural Network (GNN) algorithms are designed to solve problems formulated as a graph. In [13], the authors provided a new building block for Deep Learning (DL) and published a library called "graph nets" which is available on GitHub from the deepmind repository [14]. GNNs were explored for improving the routing: on Optical Transport Network (OTN) using Software Defined Networks (SDNs) in [8], for real-time optimization in SDNs in [9] and for autonomous vehicles using simulated robots in [15]. An end-to-end delay estimator has been proposed in [16] based on a GNN model to reduce the time required for a simulator to compute the delay metric. Recently, solutions using GNNs for 5G networks have been proposed, e.g., [17] and [18].

This contribution presents, GNN-t, a proposed solution for MANET derived from the GNN framework designed by Hope [1] for demand based routing in optical fiber-SDN. The objective of the GNN-t is to find an optimal routing policy reducing the potential for congestion by minimizing the number of traversals, oblivious to the traffic demand. GNN-t adapts the weight of the links according to the topology. Quantitative comparisons are analysed for 3 routing algorithms, GNN-t model, ODRb [5] and OLSRd2. The GNN-t model is trained off-line using a simulated environment to predict the edge weights leading to routes expected to minimize congestion. These routes are imposed on the MANET topology reproduced by using the open-source real time emulation platform EMANE. The realistic Anglova CP1 scenario, Figure 1, consists of 24 nodes in hilly and forested area moving from North to South in unison and is used to assess the performance of the proposed method. Performance are quantified using mainly the Completion Ratio (CR) for an acknowledged-message service, as in [5, 6].

The next section, Section 2, presents a brief discussion of publications on reinforcement learning for routing protocols. In Section 3, a model derived from [1], called GNN-t, is proposed as an attempt to compute optimal routing for MANETs. The experimentation and the performance of the GNN-t are described in Section 4. Finally,

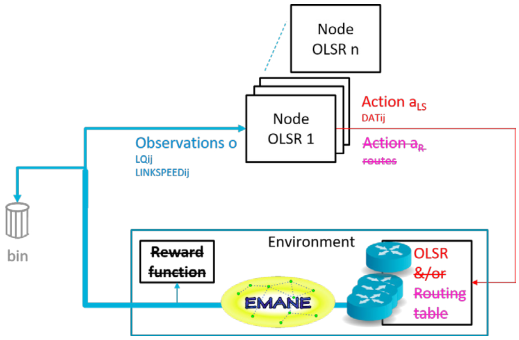

Fig. 3: Architecture of the OLSRd2 routing protocol to match the classical Reinforcement Learning architecture but without the unused reward.

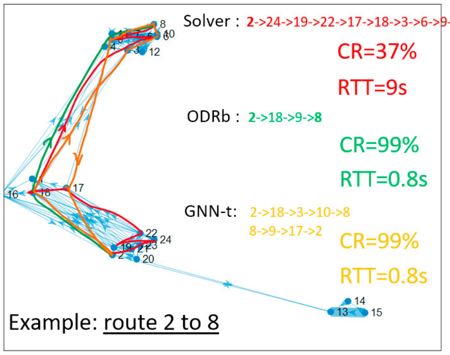

Fig. 4: Routes for the solver in [1], ODRb and GNN-t in AnglovaTs1.

Section 5 concludes that GNN-t is capable of minimizing and finding alternative paths for routing optimization problems but appears to be limited as long as traffic is not taken into account.

2. State of the Art

In MANET, routing decision should mitigate the effect of packet loss due to links with a poor quality or due to congestion. To simulate a realistic environment, radio links can be characterized by a Probability Of good Reception (POR) as in EMANE [19] or, equivalently since POR=1-PER, the more conventional Packet Error Rate (PER). The packet reception depends on the transceiver characteristics, including the packet size, and on the environment via the radio channel between the Tx and Rx, e.g., [20, 21].

Traffic congestion will lead to packet delay and, possibly, loss and occurs when a node cannot transmit all the packets queued on the node. The congestion problems and routing solutions can be visualized in Figure 2: Panel a) shows the expected optimal routes with possible weights in the simplest possible topology. In Figure 2 b), an additional node (node 6) is inserted on the path $1-5-(6)-4$. The traffic $1 \rightarrow 4$ (in yellow) congests the node 2 when wrong decisions about the weights are made. The congestion is due to the traffic $3 \rightarrow 4$ (in purple) going through the node 2. In Figure 2 c), a correct estimation of the weights of the route $1-6-5-4$, allows the traffic $1 \rightarrow 4$ to avoid node 2.

OLSR discovers the network using Hello and Topology Control messages and use Multi-Point Relays (MPRs) to exchange routing information. The Figure 3 shows an overview of the OLSRd2 architecture (RFC 7181). The OLSRd2 daemon [22] runs on each node to observe the link states, i.e., Link Quality on each link, and compute a Directional Air Time (DAT) metric per link. The DAT metrics are used to compute the routing table of the node. OLSR performs poorly in the realistic Anglova CP1 scenario [5]. The complexity of the Anglova scenario with a highly dynamic environment is challenging for the routing protocol OLSR.

Finding an optimal routing configuration that maximizes the link utilization in communicating networks can be solved using linear programming when fractional flows are allowed: the problem is called fractional multicommodity flow problem [23]. The integer multicommodity flow problem is NP-hard as presented in [24] and [25]. Heuristic and ML based solutions can be considered to find faster and close to optimal routing configuration for fractional or integer flows [26].

A Reinforcement Learning solution called OLSR-Q was proposed in [7] to keep the benefits of the routing protocol OLSR but modifies the link states using a Q-learning algorithm with the queues on each node as reward. In [7] first and later in [10, 6], Q-learning algorithm was considered to detect alternative paths and to reduce the congestion. The OLSR-Q architecture is similar to the OLSRd2 architecture shown in Figure 3. OLSRd2 does not benefit from the queues to detect congestion. Instead, The OLSR-Q uses the queues on each node to compute a local reward per node and set a Q-value for each link. The Q-value sets an artificial LINKSPEED which then depends on the local congestion. Instead, OLSR-Q agent runs on each node and updates the Directional Air Time DAT_{ij} (action) according to the computed LINKSPEED. The process is repeated at each time interval of

5 seconds. OLSR-Q variants presented in [10, 6] perform as expected, with oscillations, in a canonical 9-node 2-ring topology but did not preform well on the Anglova CP1 scenario (CR=83% < OLSRd2 = 88%). Future rewards, to be approximated by the OLSR-Q agent, proved to be more complex than expected.

Function approximators such as Neural Networks should be able to learn a good policy without visiting all possible state-action. Due to the natural graph representation of MANETs, Graph Neural Network (GNN) techniques have been investigated. Recently, Graph Neural Network (GNN) techniques have been applied to Software Defined Networks (SDNs) to improve the routing protocol. Here, three papers are discussed.

In [27], three different environments are considered to assess the performance of a generic graph convolutional based algorithm called DGN. One environment is a routing problem where the packets have to be routed in what the authors claimed to be a Software Defined Network (SDN). The network is generated randomly at the start and all nodes have exactly 3 connections to other nodes. The model is trained and exploited on the same topology. The links have limited capacity based on a maximal link bit rate. Once a packet leaves a link, the load of the link decreases by the size of the packet n_b. The packet size $n_b \in]0, 1]$ is randomly generated. The propagation time on the cable between nodes depends on the length in meters of the link. A longer link will delay the packet by a number of time steps t. Consequently, the environment proposed in [27] is not realistic and does not apply to wireless communication. Despite the author's Gitlab repository [28] which greatly helped to understand their work, we have not been able to use the generic DGN for MANET.

In [8], a Deep Reinforcement Learning (DRL)+GNN model is used to predict the best route to allocate the demand in an Optical Transport Network (OTN) environment. The environment generates a demand defined by a source node, a destination node and a bandwidth at any step. The topologies used to assess the performance are NSFNet (14 nodes) and GEANT2 (24 nodes). Note that the agent is trained on different networks with smaller number of nodes than the testing networks. The environment has the following constraints when an allocation is performed: when a route is allocated, the routes cannot be removed or replaced. The agent receives a positive reward each time an allocation is performed successfully. If the allocation of the resource cannot be accepted by the environment, the agent gets a negative reward and the episode ends. The implementation of the model in [8] is the following: the observations used by the model are the link available capacity in bps, the link betweenness and an action vector. The link betweenness in graph theory is the number of paths that pass through the node. The betweenness is a measure of the centrality of the node in the network. Note that we will use the term: traversal. The action vector is a one-hot representation of the demand. $k = 4$ actions are evaluated by the agent and it predicts a numerical value for the Q-Value $Q(s, a)$. Four different routes which have the same number of hops are evaluated by the agents. The objective of the agent is to predict among $k = 4$ paths the best route which leads to maximize the allocation of future demands. The results in [8] show that the model outperforms the state of the art DRL agent, the load balancing (LB) algorithm and a theoretical fluid model. The model in [8] has two limitations. First, no more than four routes can be evaluated at each step. In the Anglova scenario, the number of possible paths is often larger than four. Second, the model allocates the demand but does not free up the routes that are no longer used by the communicating pairs. The complexity to adapt the DRL+GNN agent to our MANET problem leads us to consider the recent work by Hope and Yoneki [1].

A solution called GNN-based Data Driven Routing (GDDR) for optimal routing in SDN was proposed in [1] and uses a RL scheme. The framework consists of a simulated environment with topology scenarios from the topology-zoo dataset [29]. The demands are provided by the totem dataset [30]. The RL algorithm, an actor-critic based on Proximal Policy Optimization (PPO) [31] and available from the stable-baselines library [32] is used in the framework. The proposed solution, a policy model based on GNN that is capable of minimizing the maximum link over-utilization of the network in multicommodity flow problem. Appropriate weights of links are predicted by the agent to route the demand. The agent is tested on graphs unseen during training and graphs with larger number of nodes with good performance. The GNN model is compared to a MLP implementation originally proposed in [33]. The GNN model outperforms the MLP model and has a better generalization capability. An interesting feature of the GDDR solution is the use of Linear Programming solver to compute the reward. The basic idea of the GDDR approach is to train off-line a GNN using CPU intensive optimal solutions and then exploit the GNN in real time environments. As explained in more details in the next section, the GDDR Linear Programming solver resulted in poor routing decision for MANETs. However, the GDDR model [1], even without the LP Solver, is of major interest as the GNN is trained to predicts, similarly to OLSR-Q, weights leading to optimal routing. The next section introduces in more details our proposed GNN-t based on the GDDR model [1].

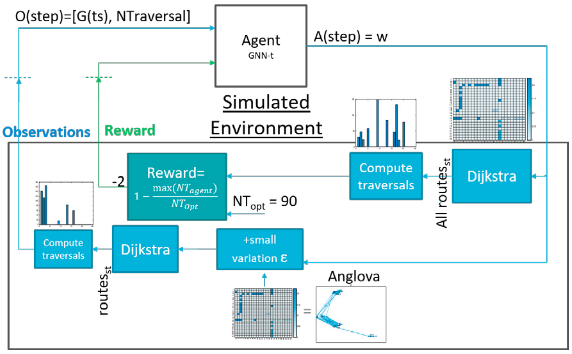

Fig. 5: The proposed GNN-t model based on [1].

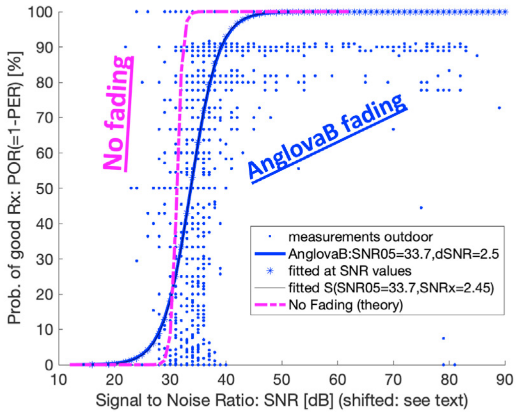

Fig. 6: Probability Of good Reception (POR) versus SNR.

3. GNN for MANETs

Papers presented in Section 2 consider SDNs to improve the performance of the routing protocol. MANETs and SDNs are relatively distinct types of networks and features three major differences. (1) in SDN, optical cables usually connect interfaces of a router with the rest of the network, in MANET radios are used, (2) in SDN, since a router connects another unique router with cables (two cables for bidirectional communications), the access of the medium is simple. In MANETs the communication relies on Medium Access Controller (MAC) typically Carrier Sense Multiple Access (CSMA), slot-based access such Time Division Multiple Access (TDMA) or OFDMA. TDMA is consider here to allocate and guarantee a time slot to access the medium for tactical applications. (3) Increasing the number of interfaces on a router will usually increase the throughput, i.e. multicommodity flows in [34, 1]. In MANET, radio communications broadcast the signal: all neighbours should listen and the neighbours of the intended receiver should not interfere. The MANET throughput is reduced by the nodes routing the packets. To our knowledge, no work has been published on GNN to improve the quality of Service (QoS) in realistic MANETs.

The GDDR frameworks [1] is not well suited for MANETs. For example, GDDR does not limit the number of hops to the destination. In the Anglova CP1 scenario, a route between node 2 to node 8 has 9 hops as shown in red in Figure 4 when using the GDDR-based solution [1]. Only 3 hops are needed when using the ODRb algorithm [5]. An experiments was conducted on MANET using the solver for SDN. We tried to used binary flows instead of fractional flow to impose a single route. A fraction of flow is associated on each edge, the path with the largest fraction of flow was considered in the initial tests on MANETs. The binary flow showed poor performance in MANET: a CR of 37% compare to 99% for ODRb. In similar conditions, CR is 88% for the routing algorithm OLSRd2. The flow should be forced on a single path using additional constraints.

The proposed model GNN-traversal (GNN-t) presented in Figure 5 was designed for MANETs. The RL algorithm used with GNN-t is PPO [35]. GNN-t implements the original Message Passing GNN (MPGNN) in [1] with small changes to the policy network. The observation of the agent is the graph and the number of traversals NT. NT were reported in [5] to be an important metric for routing protocols. The number of traversals $NT_i \in NT$ is the sum of the number of times node i routes (except for source node s and target node t) in all paths P. More formally, a path $p_{st} = \{v_{k(i)}, ..., v_{k(|p_{st}|)}\}$ from source s to target t has a sequence of routing nodes $v_i | i \in (1, ..., N)$. $k(i)$ is the index of i-th node on path p_{st}, as defined in [16]. The number of traversals on node i is computed as $NT_i = \sum_{p_{st}}^{P}(v_i)$.

The GNN-t agent predicts weights w_{ij} between $[0, 1]$ for the edges of the MANET graph. The Dijkstra algorithm is used to compute all shortest paths. Then, the number of traversals NT is derived for each node. The reward is computed using Equation 1 and forces the agent to minimize NT in order to obtain large rewards. The optimal NT_{opt} is estimated to be equal to $max(NT)$ computed by using the ODRb routing scheme. For example, ODRb has 90 maximum traversals on node 16 for the AnglovaTs1 scenario shown in Figure 7. Since ODRb might not provide the optimal routing, the reward is set to be negative for a number of traversals larger than NT_{opt} and positive when less traversals are achieved by GNN-t. To compute a set of unique route, a small variation $\epsilon = 1e-8$ is randomly applied to the weights.

$$r = 1 - \frac{NT_{agent}}{NT_{opt}} \tag{1}$$

The summary of the parameters of the GNN-t model are listed below:

- Observations: Binary connectivity graph (or adjacency matrix) and number of traversals NT_i per node
- Action: list of weight w_{ij} for the connected links
- Reward: the reward is presented in Equation 1
- Number of steps: 100000 steps which lasts 30min on a 2.5GHz machine
- GNN-type: Message Passing GNN (MPGNN) with 3 layers of 64 neurons, the tanh activation function and a shared value function are used
- Number of iteration of message passing: 2 iterations
- RL-algorithm: PPO [35]

4. Results

Results have been obtained by training the GNN-t agent using a simulation tool specially developed as described above.

The OLSRd2, ODRb and the trained GNN-t agents are used on the Anglova CP1 scenario [36]. Our so-called TAKE user messages generator has been used [37, 38, 6]. The TAKE generator is sending UDP short messages (<300B) and expect an acknowledgment within 20s. The user traffic is considered all to all with 1 message/s/node. The message completion ratio (CR) is chosen as a simplified QoE metric.

Results are obtained using the open-source EMANE framework. Details of our implementation of the EMANE platform are provided in the open repository [39] and published in [38, 6]. The real time emulation platform EMANE relies on packets based transmission. For each packet, a binary "drop - no drop" decision is made based on the packet size, the computed SINR and on a POR-SINR look-up table. The POR-SINR look-up table must be provided for a default packet size (0 to avoid the conventional packet size correction) and a given bit rate, typically to account for a particular modulation and coding scheme. It is well known [21, 40], that the POR-SINR (or SNR) curves depends also on the radio environment, i.e., on the propagation channel and speed of the vehicles. Numerical experiments using MATLAB, although not shown here, were confirmed with measurements using Silvius STREAMCASTER 4200 radios. Measurements using iperf3-UDP are used to approximate a realistic POR-SINR curve as shown in Figure 6. The real radios to be emulated are different from the Silvius radios and the POR-SINR curve were shifted to fit the no fading case. The non-fading case has been presented in [38]. The POR-SINR curve under fading is called here the AnglovaB curve due to the similarity of the POR-SINR curve with the IEEE802.11 TGn model B channel results. More details are claimed to be out of the scope of this contribution focusing on the GNN approach. Preliminary real time emulation results with and without fading lead to the somewhat expected degradations in lines with the simulations by Sterner and Uppman [21].

The GNN-t model was designed to minimize the number of traversals: Figure 7 shows the number of traversals per node. GNN-t minimizes the $max(NT)$ to 84 traversals compared to ODRb: 90 traversals and OLSRd2: 95 traversals. The node 13 has similar number of traversals: 84 for the two omniscient approaches: ODRb and GNN-t and 85 for OLSRd2. The node 13 connects 2 nodes: 14 and 15 to all other nodes of the network. As node 13 must route all traffic to and from nodes 14 and 15 from and to all other 21 nodes: 2 directions x 2 nodes x 21 pairs = 84 traversals are to be expected and cannot be decreased. The total number of traversals for GNN-t: 599 is larger than ODRb: 582 because GNN-t seeks to reduce the $max(NT)$ by searching for longer routes with less traversed nodes. OLSRd2 has an even smaller total number of traversals: 435 because routes with less hops but less reliable links: with POR <= 100%, might be chosen. The less-hop and less-reliable routes of OLSRd2 explains well the lower completion ratio CR=88%; lower than the CR=99% using ODRb or GNN-t. ODRb has 90 traversals because only links with POR = 100% are considered. Consequently, the GNN-t approach is claimed to be a promising scheme

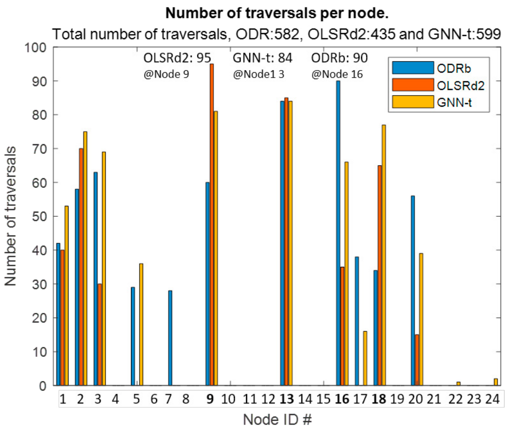

Fig. 7: The number of traversals per nodes resulting from ODRb (blue), OSLRd2 (red), and the proposed GNN-t (orange).

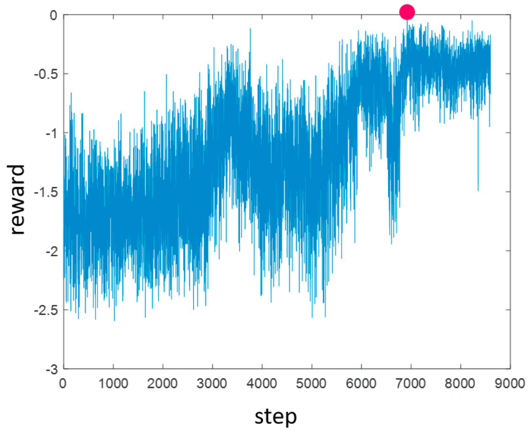

Fig. 8: Reward of the GNN-t agent for the first 9000 steps (3min CPU). The red dot at step 7000 shows the best model leading to a maxNT=84 (Reward = 1-84/90 = 0.07).

Table 1: User traffic [bps/node]/ Round Tip Time (RTT) [s] / Completion Ratio (CR) [%] are presented for OLSR, ODRb and the GNN model on AnglovaTs1 (first second of the Anglova CP1) scenario.

Routing Algorithm	User traffic [bps/node - msg/s/node]	Number of traversal	Jain's Fairness	RTT [s]	CR [%] (No Fading)	CR [%] (With Fading)
OLSRd2	3.6k - 1	435	0.27	0.6	88	59
	7.2k - 2	440	0.26	5	75	39
ODRb	3.6k - 1	582	0.4	0.8	99	36
121dB	7.2k - 2	582	0.4	5	79	36
GNN-t	3.6k - 1	599	0.37	0.8	99	35
	7.2k - 2	599	0.37	6	77	35

to find optimal solution. Future work should focus on devising more interesting rewards including, for example, traffic demand and slot allocations.

The reward is shown in Figure 8 for the GNN-t model during the first 9000 training steps; 3 min on a 2.5GHz machine. The GNN-t appears to be slow to explore and learn good solutions. Future work will attempt to discretize the weights, i.e., the actions to reduce the action space and to find a better set of nodes, and possibly edges, features. The chosen reward should also be changed since minimizing the maximum number of traversals cannot be reduced under 84 for the particular AnglovaTs1 scenario. Other metrics, such as fairness, might be used to define a better reward.

The Table 1 summarizes the comparison of the performance between GNN-t, ODRb and OLSRd2. The Completion Ratio (CR) of OLSRd2 is 88% because the routing protocol routes the packet through links with POR <= 100%. The improvement of the GNN-t is smaller at larger user traffic: 2msg/s/node (CR=77%) compared to ODRb (CR=79%). Longer routes decrease the performance of GNN-t at 2msg/s/node. The Jain's fairness index [41] is also computed for all three routing protocols. A value of 1 means the best possible fairness, and the smallest value is 1/24=0.04, for 24 nodes.

The effect of fading leads to a very poor CR=36% or 35% instead of CR=99%, without fading, for the two omniscient schemes ODRb and GNN-t. OLSRd2 appears more robust: CR=59% with fading instead of CR=88% without fading. Future work will focus on more robust routing to mitigate the effects of fading. The use of binary connectivity matrix to derive optimal routes will be reviewed.

Figure 9 shows the maximal number of traversals for ODRb and GNN-t during the most challenging portion, the first 300s, of the Anglova scenario. The scenario has disconnected networks, as shown in Figure 9, leading to few traversals and connected clusters. Reducing the maximal number of traversals in cluster network is difficult since two nodes act as gateway between groups. For the first few seconds, GNN-t discovers better routes leading to less traversals. ODRb is close to optimal making the GNN-t model struggle to minimize further the number of traversals. Figure 10 shows the total number of traversals. GNN-t has on average a larger total number of traversals compare to ODRb: alternative paths found by GNN-t have more hops than the shortest path, as expected.

Figure 11 shows the adjacency matrices of binary graphs and weight graphs for the first two seconds of the Anglova scenario. The binary graphs is computed from the pathloss of the topology with a pathloss limit of 121dB for POR of 100% in the no fading case. Between the first two seconds of the scenario, only 2 bidirectional link are lost (1-3) and (9-18): green dot, and another links is created (5-18): red dot in top right panel of Figure 11. The GNN-t weights for the first two seconds are shown in Figure 11 and the difference of the weights between the 1st and 2nd second are presented in the last plot. The difference in weights is unexpectedly high and demonstrate the need to fine-tune GNN-t training parameters.

The performances of the routing algorithms are assessed on the Anglova scenario and presented in the Figure 12. The first 300s of Anglova is the most challenging part of the scenario with disconnections. OLSRd2 has poor performance with fading (CR=45%).

ODR121dB decreases the CR=94% without fading to CR=73% with fading. The 121dB threshold for ODR121dB must be changed to account for fading. The minimum SINR required with fading for 90% POR is 40dB (ODR115dB) instead of 34dB (ODR121dB) for the non-fading case. The CR of ODR115dB increases from 73% (ODR121dB) to 83%. The improvement of ODR115dB demonstrates the efficiency of reducing the pathloss threshold from 121dB to 115dB for the fading case.

ODRb121dB and GNN-t have both similar performance with and without fading on the challenging section of Anglova because the binary adjacency matrix is based on the same 121dB pathloss threshold. The adjacency matrix should be adapted as in ODR115dB to be robust to fading. The ODR results shows that the GNN-t might have to take into account the pathloss matrix instead the binary adjacency matrix.

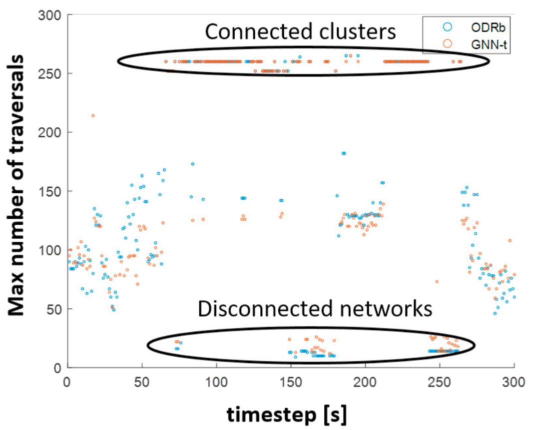

Fig. 9: The maximal number of traversals is presented for ODRb and GNN-t.

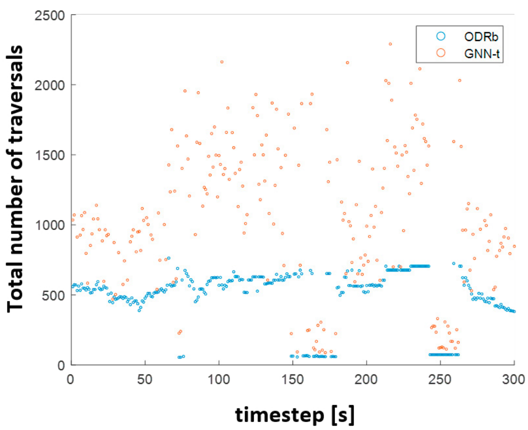

Fig. 10: The total number of traversals is presented for ODRb and GNN-t.

5. Conclusion

In this contribution, a proposed solution for improving the performance of the routing protocol in MANETs with a Graph Neural Network model was proposed based on the GNN-based Data Driven Routing (GDDR) approach [1] developed for fixed optical network. A modified version: GNN-t has been implemented to minimize the number of traversals in MANETs. In the AnglovaTs1 scenario: GNN-t learns how to reduce the maximum

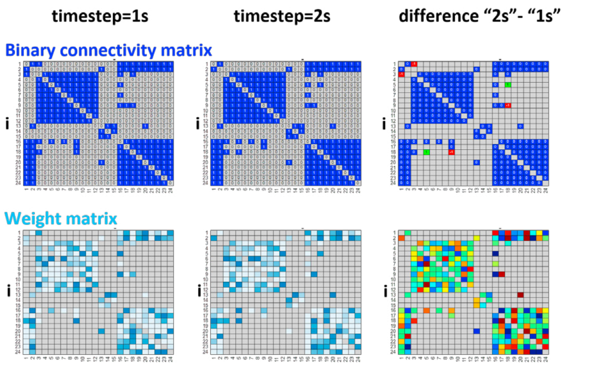

Fig. 11: The binary i to j dis(grey)-connection(blue) matrices (top left and middle matrices) and the ”fraction of hop” weight matrices are presented for the 1st and 2nd second of the Anglova CP1 scenario.

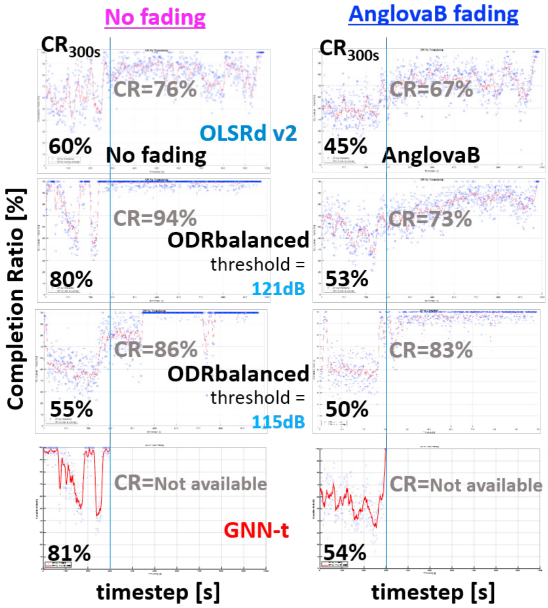

Fig. 12: The completion ratio (CR)-vs-time (dots) and the 10s-moving average completion ratio (red) of the 3 routing algorithms OSLRd2, ODRb(121dB and 115dB) and GNN-t are presented for Anglova.

number of traversals (84 traversals on node 13) compared to our best known algorithm: ODRb (leading to 90 traversals on 16). GNN-t finds alternative routes to avoid central nodes with longer routes. The agent performs similarly to ODRb with a completion ratio CR=99% and 77% (99% and 79% for ODRb) at 1msg/s/node and 2msg/s/node, respectively. Reducing the number of max(NTraversals), from 90 (ODRb) to 84 (GNN-t) does not bring significant benefit on the maximum bit rate before congestion. Reducing the number of traversals does not bring significant benefit to all performance metric: number of traversals, Jain’s fairness, RTT and CR. However, GNN-t successfully achieved the goal of reducing the number of traversals, therefore, we claimed that GNN-t remains a promising approach subject to devising more interesting observations, actions and rewards

Improving ODRb or GNN-t without the knowledge of the traffic appears difficult. Future improvements on the GNN-t algorithm will consider instantaneous or average user traffic. Predicting precisely the user traffic for tactical and emergency applications is a complicated task. The size of the queues could be used as feedback to the agent and/or the users could be requested to announce their traffic demand. Other improvements should focus on scheduling the radio resources to best use the traffic demand, the routing decisions or to compensate non-optimal routing.

Acknowledgment

Funding is provided by the armasuisse project 041-23.

References

[1] O. Hope, E. Yoneki, GDDR: GNN-based data-driven routing, Proceedings - International Conference on Distributed Computing Systems 2021-July (2021) 517–527. arXiv:2104.09919, doi:10.1109/ICDCS51616.2021.00056.
[2] olsr.org, Olsrd2 mediawiki, `http://www.olsr.org/mediawiki/`.
[3] olsr.org, Ff dat metric plugin, `http://www.olsr.org/mediawiki/index.php/FF_DAT_Metric_Plugin`.
[4] A. Laouiti, A. Plakoo, P. Minet, M. Badel, Experiments with OLSR Routing in a MANET (2006) 1–24.
[5] Y. Maret, F. Wagen, Preliminary Performance Benchmarking of OLSRd2 Using Emulated TDMA MANETs and an ODR, VTC-FALL2020 conference.

[6] Y. Maret, F. Wagen, M. Raza, J. Wang, N. Bessis, F. Legendre, Preliminary results of OLSR based MANET routing algorithms: OLSRd2-Qx reinforcement learning agents and ODRb, ICMCIS2021 conference.
[7] A. McAuley, K. Sinkar, L. Kant, C. Graff, M. Patel, Tuning of reinforcement learning parameters applied to OLSR using a cognitive network design tool, IEEE Wirel. Commun. Netw. Conf. WCNC (2012) 2786–2791doi:10.1109/WCNC.2012.6214275.
[8] P. Almasan, J. Suarez-Varela, A. Badia-Sampera, K. Rusek, P. Barlet-Ros, A. Cabellos-Aparicio, Deep reinforcement learning meets graph neural networks: exploring a routing optimization use casearXiv:1910.07421.
[9] K. Sawada, D. Kotani, Y. Okabe, Network routing optimization based on machine learning using graph networks robust against topology change (2020) 608–615doi:10.1109/ICOIN48656.2020.9016573.
[10] F. Wagen, V. Adalid, Y. Maret, Comparing OLSRd, d2 and -Q Using an Emulated 802.11 or TDMA MANET in a 2-ring Topology, ISNCC2020 conference.
[11] T. V. de Wiele, D. Warde-Farley, A. Mnih, V. Mnih, Q-learning in enormous action spaces via amortized approximate maximization (2020). arXiv:2001.08116.
[12] C. J. C. H. Watkins, P. Dayan, Q-learning, Machine Learning 8 (3-4) (1992) 279–292. doi:10.1007/bf00992698.
URL `https://link.springer.com/article/10.1007/BF00992698`
[13] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, R. Pascanu, Relational inductive biases, deep learning, and graph networks, 2020 International Conference on Information Networking (ICOIN) (2018) 608–615arXiv:1806.01261.
URL `http://arxiv.org/abs/1806.01261`
[14] deepmind, Graph nets, `https://github.com/deepmind/graph_nets` .
[15] Q. Sykora, M. Ren, R. Urtasun, Multi-agent routing value iteration network (2020). arXiv:2007.05096.
[16] K. Rusek, J. Suarez-Varela, P. Almasan, P. Barlet-Ros, A. Cabellos-Aparicio, Routenet: Leveraging graph neural networks for network modeling and optimization in sdn, IEEE Journal on Selected Areas in Communications 38 (10) (2020) 2260–2270. doi:10.1109/jsac.2020.3000405.
URL `http://dx.doi.org/10.1109/JSAC.2020.3000405`
[17] M. S. Mollel, A. I. Abubakar, M. Ozturk, S. F. Kaijage, M. Kisangiri, S. Hussain, M. A. Imran, Q. H. Abbasi, A survey of machine learning applications to handover management in 5g and beyond, IEEE Access 9 (2021) 45770–45802. doi:10.1109/ACCESS.2021.3067503.
[18] N. Vesselinova, R. Steinert, D. F. Perez-Ramirez, M. Boman, Learning combinatorial optimization on graphs: A survey with applications to networking, IEEE Access 8 (2020) 120388–120416. doi:10.1109/access.2020.3004964.
URL `http://dx.doi.org/10.1109/ACCESS.2020.3004964`
[19] Emane website, https://adjacentlink.com/emulation.html.
[20] T. Abbas, K. Sjöberg, J. Karedal, F. Tufvesson, A Measurement Based Shadow Fading Model for Vehicle-to-Vehicle Network Simulations, International Journal of Antennas and Propagation 2015. arXiv:1203.3370, doi:10.1155/2015/190607.
[21] U. Sterner, U. Uppman, On the robustness of olsr in a mobile tactical scenario in rural terrain, in: ICMCIS, 2017.
[22] Olsrv2d repository, https://github.com/OLSR/OONF.
[23] F. Babonneau, Solving the multicommodity flow problem with the analytic center cutting plane method (603).
[24] Y. H. Ezzeldin, M. Cardone, C. Fragouli, D. Tuninetti, Half-duplex routing is np-hard (2017) 89–96.
[25] A. Bley, Routing and Capacity Optimization for IP Networks, Operations Research Proceedings 2007 (2008) 9–16.
[26] M. Prates, P. H. Avelar, H. Lemos, L. C. Lamb, M. Y. Vardi, Learning to solve NP-Complete problems: A graph neural network for decision TSP, 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019 (2019) 4731–4738arXiv:1809.02721, doi:10.1609/aaai.v33i01.33014731.
[27] J. Jiang, C. Dun, T. Huang, Z. Lu, Graph convolutional reinforcement learningarXiv:1810.09202.
[28] Dgn repository, https://github.com/PKU-AI-Edge/DGN/.
[29] topology zoo.org, Zoo-topology dataset, `http://www.topology-zoo.org/dataset.html`.
[30] totem.run.montefiore.ulg.ac.be, Toolbox for traffic engineering methods, `http://totem.run.montefiore.ulg.ac.be/features.html`.
[31] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, P. Abbeel, Trust region policy optimization (2017). arXiv:1502.05477.
[32] hill a, a fork from baselines of openai, `https://github.com/hill-a/stable-baselines` .
[33] A. Valadarsky, M. Schapira, D. Shahaf, A. Tamar, Learning to route, HotNets 2017 - Proceedings of the 16th ACM Workshop on Hot Topics in Networks (2017) 185–191doi:10.1145/3152434.3152441.
[34] M. Seliuchenko, O. Lavriv, O. Panchenko, V. Pashkevych, Enhanced multi-commodity flow model for qos-aware routing in sdn, in: 2016 International Conference Radio Electronics Info Communications (UkrMiCo), 2016, pp. 1–3. doi:10.1109/UkrMiCo.2016.7739623.
[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms (2017). arXiv:1707.06347.
[36] Anglova scenario [cited 7.1.2019].
URL `https://anglova.net/`
[37] S. Ruffieux, C. Gisler, J.-F. Wagen, F. Buntschu, G. Bovet, Take - tactical ad-hoc network emulation, in: ICMCIS, 2018. doi:10.1109/ICMCIS.2018.8398728.
[38] A. Nikodemski, J.-F. Wagen, F. Buntschu, C. Gisler, G. Bovet, Reproducing Measured MANET Radio Performances Using the EMANE Framework, IEEE Commun. Mag. 56 (10). doi:10.1109/MCOM.2018.1800294.
[39] Predictake repository, https://gitlab.forge.hefr.ch/predictake.
[40] 802.11n packet error rate simulation for 2x2 tgn channel [cited 14.2.2022].
URL `https://www.mathworks.com/help/wlan/ug/802-11n-packet-error-rate-simulation-for-2x2-tgn-channel.html`
[41] M. Koppen, K. Ohnishi, M. Tsuru, Multi-jain fairness index of per-entity allocation features for fair and efficient allocation of network resources (2013) 841–846doi:10.1109/INCoS.2013.161.