Poster: Learning to Shine – Optimizing Glossy at Runtime with Reinforcement Learning

Valentin Poirot[†], Olaf Landsiedel[‡†]
[†] Chalmers University of Technology, Sweden
[‡] Kiel University, Germany
poirotv@chalmers.se, ol@informatik.uni-kiel.de

Abstract

Glossy is a dissemination protocol that allows a node to propagate information to the entire network through constructive interference. We present GLOSSAI, a new artificial intelligence-based version of Glossy. We use reinforcement learning to determine and update Glossy's parameters at runtime. Each node individually learns the best strategy to minimize energy consumption while maintaining high reliability. Furthermore, nodes can dynamically adapt their parameters to follow the dynamics of the medium.

1 Introduction

Context. Energy efficient communication is a cornerstone of low-power wireless networks. Nodes are often powered by batteries, have strong resource constraints, and must operate for an extended period of time without maintenance. These devices operate by performing local sensing or actuation, computation, and by exchanging information. Different types of traffic are common in low-power wireless networks: sensed data can be *collected* by a central node, which acts as a gateway; information can also be *disseminated* from one node to the entire network, for example to distribute a new network configuration; finally, more complex traffic, such as many-to-all broadcast, might also be present.

To consume as little energy as possible when communicating, numerous protocols have been designed. Each targets a specific class of traffic to perform as efficiently as possible: Glossy [2] offers efficient dissemination, while A^2 [1] is used for network-wide agreement, for instance. In order to minimize energy consumption, those protocols rely on specially defined rules, transmission policies, and parameters that were carefully handpicked and tuned to offer the best performance. As example, Glossy repeats a transmission N times before shutting down the radio, N being a parameter set by the end-user at the beginning of a new deployment. A high value for N will offer better performance (e.g., reliability) at the cost of more energy consumed, whereas minimizing N minimizes the power consumed, but also reduces the performance.

Challenges. In Glossy, the transmission policy is *static*, i.e., it is defined at startup and remains the same for the entire network's life. In contrast, the wireless medium is highly *dynamic*: temperature, mobility, and co-existing technologies can significantly affect its properties. To counteract these effects, Glossy deployments use a high N parameter. Performance and connectivity are maintained even under external interference, at the cost of an increased energy consumption.

Approach. We present GLOSSAI, a new version of Glossy that leverages the advances in artificial intelligence and machine learning to bring *adaptivity* to the dissemination protocol. GLOSSAI chooses the best N for each node at runtime to minimize energy consumption. Furthermore, it is able to react to the dynamics of the medium. We use Reinforcement Learning (RL), a technique that received a lot of attention lately by beating the world champion at the game of Go, and by learning how to beat top chess players in less than four hours.

2 Reinforcement-driven Glossy

Reinforcement learning. We motivate why Glossy can be represented as a RL problem. In RL, an *agent* tries to learn which *action* (or list of actions) achieves an unknown goal and maximizes an arbitrary *reward* function. An agent learns through trial-and-error, i.e., by acting on its *environment* and by *observing* the changes induced. This is different from

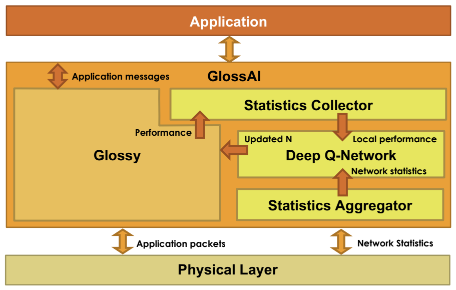

Figure 1. Architecture of GLOSSAI. Local statistics and statistics from other nodes in the network are periodically collected and used as an input to a deep Q-network, which outputs the updated N parameter for Glossy.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2019
25–27 February, Beijing, China
© 2019 Copyright is held by the authors.
Permission is granted for indexing in the ACM Digital Library
ISBN: 978-0-9949886-3-8

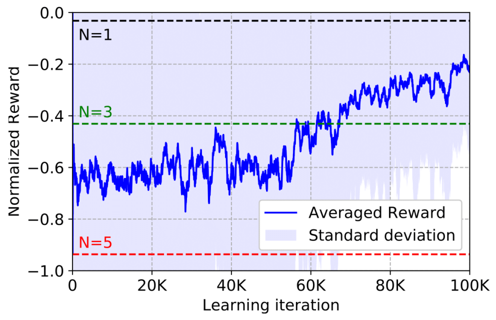

Figure 2. Rewards during the learning phase. A reward is an arbitrary scalar in [-1,0]. We train GLOSSAI over traces taken from Flocklab. Each iteration represents one update of N.

traditional supervised learning, in which some input-output tuples are known (i.e., we have training examples), and the goal is to generalize to unseen input.

Glossy follows a similar structure. Each node corresponds to one agent. A node wants to maximize performance, i.e., reliability, and wants to minimize energy consumption, i.e., the radio-on time. Our reward function is therefore rewarding a node if its radio-on time is low, while the reliability of the system stays high. Periodically, a node decides if it needs to update Glossy's N parameter, i.e., the number of time a received packet is retransmitted during a flood. Updating N will likely affect both the radio on-time and reliability of the flood. Therefore, the agent can observe how useful its action was, and compute a new reward.

Architecture. Fig. 1 presents the architecture of GLOSSAI. We build a new layer around Glossy, and introduce a control subsystem. We collect both local performance (i.e., radio-on time and reliability) and statistics from all other nodes periodically. We then use a deep Q-network to compute the new N parameter for Glossy.

Execution. GLOSSAI is divided into two phases: a *learning phase* and an *operational phase*.

Learning phase: the deep Q-network starts by choosing random actions and observe how well the system is performing. Periodically, the neural network is updated to reflect new discoveries. As time passes and as the agent builds up knowledge about its environment, it transitions from using randomness towards choosing the best action, i.e., the action that is more likely to give a higher reward. To improve stability during learning, all nodes are first trained together. A unique neural network is trained for the entire network over traces obtained from a testbed for many iterations. Once the learning converged to a correct behavior, we copy the trained neural network to all nodes, and then train each node independently on the testbed. This technique allows GLOSSAI to: **(a)** improve stability and guarantee convergence during the first step and **(b)** allows an near-optimal individual behavior with the individual learning step.

Operational phase: nodes do not rely on randomness anymore. Instead, each local N parameter is always locally computed based on the current state on the network, and on the knowledge of its dynamics. Periodically, nodes exchange their local performance with the rest of the network. Since each node learned an individual strategy, it converges to the optimal local N based on its location in the topology.

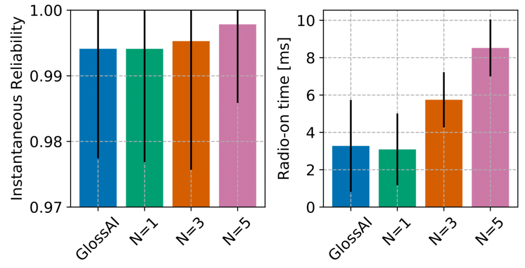

Figure 3. Comparing Glossy and GLOSSAI during the operational phase. The reward function influences the tradeoff between energy efficiency and reliability.

3 Preliminary Results

We base our work on the original implementation of Glossy, for Tmote Sky running Contiki. Fig. 2 presents GLOSSAI during its learning phase. The system is trained on traces containing over 6 000 floods from the Flocklab testbed (27 nodes). A single neural network is trained for all nodes. As the system builds knowledge of the network dynamics, it learns to reduce N to save energy. The standard deviation decreases as the probability of taking random action linearly decreases with time. After 100K iterations (i.e., trials), the system learned that under normal conditions (no external interference injected), N must be kept to a minimal value, likely 1 or 2, to minimize energy consumption.

Fig. 3 shows how GLOSSAI behaves during operational runtime. Because the system converges to a very small N for normal conditions, it obtains the same reliability as Glossy with $N = 1$. The higher radio-on time is mostly due to the convergence time of N towards 1, and the fact that GLOSSAI will increase N after dropped packets, until the reliability is stable again.

4 Challenges and Future Work

Learning how to optimize Glossy at runtime is no easy task. We list three main challenges that delineate our future work: **(a)** Individual learning is more prone to instability as it does not fulfill the assumptions behind guaranteed convergence of learning. Thus, special care must be given to individual learning to allow a stable and reproducible process. **(b)** Maintaining up-to-date information about the network's state can be an expensive operation, as Glossy does not provide feedback. The collection step can cause an important overhead if done frequently. **(c)** Executing a neural network can be an expensive computation step; the neural network must remain small and must only contain simple operations.

5 References

[1] B. Al Nahas, S. Duquennoy, and O. Landsiedel. Network-wide consensus utilizing the capture effect in low-power wireless networks. In *ACM SenSys*, 2017.

[2] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh. Efficient network flooding and time synchronization with Glossy. In *ACM/IEEE IPSN*, 2011.