Replication control strategy based on a simple game of life in opportunistic networks

Vittalis Ayu[1,2*], *Bambang* Soelistijanto[2], and *Yasintha Putri* Larasati[2]

[1]School of Computer Science, University of Nottingham, United Kingdom
[2]Department of Informatics, Sanata Dharma University, Indonesia

Abstract. Although flooding-based message dissemination in opportunistic mobile networks yields high delivery performance, the nodes' resources, such as energy and buffer, are rapidly depleted due to the enormous quantity of message replicas dispersed over the networks. This study aims to reduce the number of message replicas in the networks while maintaining an acceptable delivery rate. Inspired by Conway's Game of Life, which uses cellular automaton-based neighbor conditions to regulate the total population, we propose RiGoL, a replication control-based technique for determining how many neighbors hold the same messages. We utilize a counter to keep track of the number of neighbors and two thresholds, namely forward and drop thresholds. Our strategy works as follows: when a pair of nodes meet, the node checks to see if the peer has the same message. The counter is increased by one if the peer does not hold the same message. On the other hand, if the peer already has the message, the counter is decremented by one. Only when the counter exceeds the forward threshold, then the message forwarded to another node. In contrast, the message is deleted from the node's buffer if the counter value is less than the drop threshold. We conduct extensive simulations using ONE Simulator to evaluate our proposed strategy. The simulation results show that although RiGoL yields higher latency than Epidemic, RiGoL successfully reduces the message replication in the network and prolongs the hub node's lifetime.

1 Introduction

The emergence of mobile communication technology has accelerated the growth of mobile-based communication networks like Opportunistic Mobile Networks (OMNs) [1]. However, OMNs inherit the characteristics of Mobile Ad-Hoc Networks in which nodes can move dynamically and are intermittently connected [2]. As a consequence, a pairwise node's encounter becomes opportunistic. Therefore, in this networks, a store-carry-and-forward mechanism is used to facilitate the transmission of messages from the source node to the destination node. To facilitate the message forwarding process, routing protocols such as Epidemic [3] enable the nodes to relay the messages to every encountered node. Although this forwarding strategy yields high performance in message delivery, the number of message replication is enormous.

---

[*] Corresponding author: vittalis.ayu@nottingham.ac.uk

Our approach, however, aims to reduce message replication in the networks. We proposed RiGoL, a message replication control strategy, by measuring how many neighbors hold the (copy of) messages. RiGoL's mechanism is inspired by Game of Life (GoL), a classic cellular automaton game that models the growth and shrinkage of populations [4]. However, because in OMNs, each node lacks global knowledge of networks, we define RiGoL to enable nodes to adaptively adjust their message replication rate based on the local information from their peers. When a large number of peers have the same message, we assume that this message is already highly replicated in the networks. Therefore, we drop the messages to prevent further replication growth. In contrast, when many neighbors do not have a message, we assume that the message is not replicated enough in the networks; in this case, we aim to replicate and forward the message to other nodes to help the message reach its destination. This replication control strategy was subsequently implemented and simulated using ONE Simulator [5]. We compared RiGoL with Epidemic (with no replication control) and measured the total number of relayed messages, overhead ratio, and average latency to evaluate our approach. Furthermore, we conducted an investigation on the impact of RiGoL to the nodes lifetime (in terms of energy) within the networks.

The remaining sections are organized as follows: Section 2 discusses related works, while Section 3 presents the strategy design for RiGoL. The simulation results are then presented and discussed in Section 4. Section 5 summarizes the conclusion of this paper.

2 Related works

In OMNs, messages are replicated across the networks to reach their respective destination. More replication will increase the likelihood of delivery success and decrease delivery latency. Nonetheless, flooding-based routing strategy, such as Epidemic [3], imposes uncontrolled replication in the networks, which can eventually lead to traffic congestion and degrade networks performance [6]. Spray and Wait [7] is an alternative routing strategy that limits the number of allowed messages replication in the networks. Even though this method is better than Epidemic in reducing replication, the number of allowed replications is predetermined statically at the beginning of the experiment and cannot be modified on the fly. Therefore, it is necessary to determine how to adjust the message replication dynamically.

The classic example of cellular automata is Conway's Game of Life [4]. Conway defined a set of principles for cell evolution in a grid of squares so that a cell's state of life or death is determined by its surrounding environment. For instance, a cell is allowed to survive if it has one to two "alive" neighbours, but if it has at least three "alive" neighbours, it will die because the cell population is considered to be too dense. A dead cell, on the other hand, will become alive if it is surrounded by at least three more "dead" cells [8]. Figure 1 also shows how GoL could dynamically regulate population growth on the abstraction level of OMN nodes. The node in grey represents a living node, while the node in white represents a deceased node. In this example, we observe the evolution of node N2 as a function of its neighbours' states. In Figure 1(a), N2 is surrounded by deceased nodes (N1, N3, and N4); therefore, N2 must become alive to increase the number of living nodes. In contrast, as shown in Figure 1(b), N2 remains deceased if it encounters two "alive" nodes (N1 and N3). Figure 1(c) illustrates that if N2's neighbours are all alive, it is transformed into a deceased node because the population is considered dense enough.

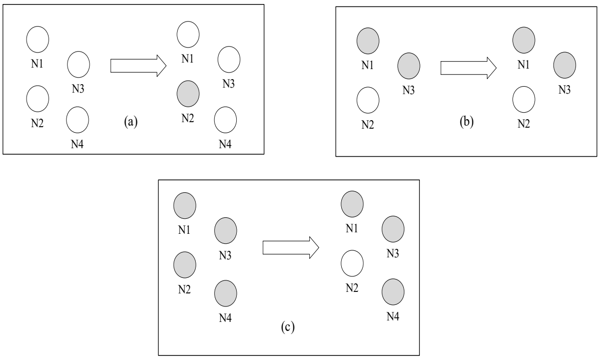

Fig. 1. Illustration of the works of GoL (a) node N2 become alive (b) node N2 stays dead (c) node N2 dies

Image processing and cryptography are among the applications of cellular automata, specifically Game of Life (GoL) [9]. In image and language recognition and classification, a GoL-based rule is used to detect associations from a set of samples. In cryptography, GoL is also used to regulate the effectiveness of a pseudo-random number generator utilised in the encryption process. In another study, Wen et al. [10] use timed cellular automata to optimise flight scheduling. Furthermore, Yang et al. [11] employ cellular automaton as a means to simulate traffic flow in the context of longitudinal driving and lane changing. Indeed, cellular automaton, specifically GoL, has been widely employed in diverse contexts. However, none of those studies examine GoL to reduce message replication in networks. In this study, we intend to design and implement GoL as a basis for RiGol in order to reduce the number of duplicate messages in networks.

3 Design of algorithms

GoL's set of rules influences the design of RiGoL. The primary objective of this approach is to limit the frequency of message dropping and forwarding by assessing if neighbouring nodes have already received identical messages. To prevent excessive replication, it is acceptable to remove the message from the node's buffer if it has already been received by numerous neighbours. In contrast, in cases when a certain number of neighbouring nodes do not have the message, the node is encouraged to duplicate the message and transmit it to its neighbouring nodes.

The algorithm is defined as follows: as depicted in Figure 2, a counter C is created to keep track of the number of identical messages in the buffers of other nodes. As an illustration, we set C_{M1} as a counter for message M1. When two nodes meet, they examine their peers' buffers for duplicate messages. If there is a match, the counter C is decremented; otherwise, the counter C is increased. Furthermore, the counter's lower and upper bounds are represented by two thresholds, T_{drop} and $T_{forward}$. When the value of the counter C is equal to or less than T_{drop}, the message is removed from the buffer. When the value of the counter C is equal to or greater than $T_{forward}$, the message is sent to the relay node.

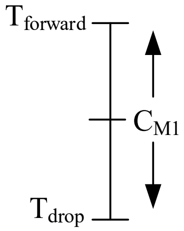

Fig. 2. Illustration of M1's counter C_{M1} dan two thresholds ($T_{forward}$ and T_{drop}).

Figure 3 depicts the evolution of the messages counter at node P in the RiGoL mechanism. Figure 3 depicts the chronological encounter of a paired node over three blocks (specifically, node P encountering node Q, node V, and node X). The message counters are initialised to zero. If an identical message exists in the peer's buffer, the counter for messages is incremented. If the condition is not met, the counter is decreased.

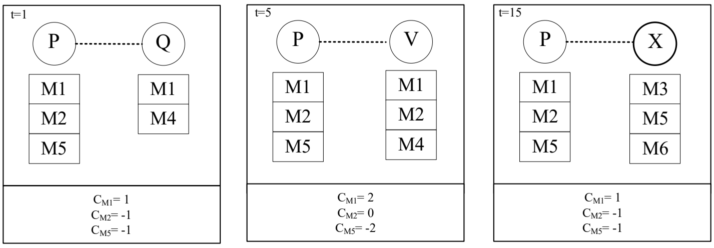

Fig. 3. Illustration of message counter's calculation in node P.

In this section, we will now introduce the RiGoL mechanism, which is outlined in Algorithm 1. The primary goals are to update the message counter and decide whether a message will be forwarded or deleted. We denote node i as N_i and node peer as N_j in the first phase of RiGoL's method. Every message counter should be updated when N_i and N_j meets. Afterwards, N_i will then treat each message in its buffer based on its counter value. The message will be replicated and sent to the peer if the counter value equals or exceeds the forwarding threshold $T_{forward}$. The message is removed from the buffer if the counter values are equal to or less than T_{drop}. Otherwise, the message is not copied and remains in the buffer.

ALGORITHM 1. Replication Strategy based on Game of Life (RiGoL)

```
When $N_i$encounter $N_j$, do
/*Update message counter*/
  for each $m_i$ in buffer $N_i$ do
    if $m_i$ exist in buffer $N_j$ then
      $C_{mi}$++ else $C_{mi}$--
    end if
  end for

/* Decision of message forwarding or dropping */
  for every message counter $C_{mi}$ do
    if $C_{mi} \geq T_{forward}$
      then forward message $m_i$ to $N_j$
      else
        if $C_{mi} \leq T_{drop}$
          then drop message $m_i$ from buffer $N_i$
          else
            do nothing
        end if
    end if
  end for
```

4 Results and discussion

The RiGoL algorithm was evaluated using the Java-based ONE Simulator to simulate OMNs [5]. We employ three performance indicators to evaluate the efficacy of our strategy for reducing message replication:: total relayed messages, overhead ratio, and average delay. In addition, we investigate the effect of RiGoL's implementation on the energy consumption of nodes.

In this simulation, we present RiGol at five distinct threshold levels denoted by T(T_{forward}, T_{drop}) and Epidemic [3] as a benchmark protocol in this simulation. This simulation uses the Haggle-3 Infocom 5 dataset [12] as connectivity traces to represent the actual human encounter patterns in a conference setting.

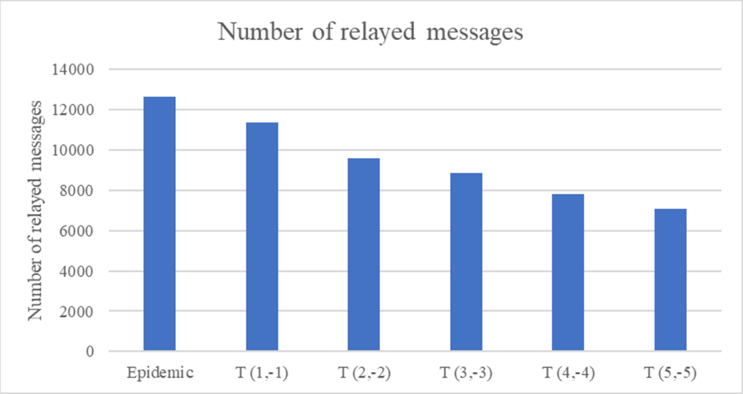

Fig. 4. Number of relayed messages.

Figure 4 depicts the total number of messages relayed by Epidemic and RiGoL protocols over different threshold values. It is evident with the Epidemic protocol, nodes transmit an enormous amount of messages. In contrast, RiGoL possesses the capacity to decrease the quantity of relayed messages. In addition, a higher RiGoL threshold could further reduce replication. RiGoL with $T_{forward}$=5 and T_{drop}=-5 generates the smallest number of relayed messages, as shown in Figure 4. RiGoL also outperforms Epidemic in terms of overhead ratio. As shown in Figure 5, a higher RiGoL threshold decrease the network's overhead ratio.

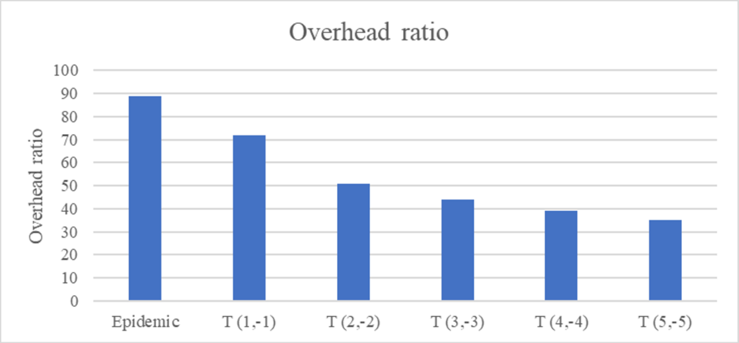

Fig. 5. Overhead ratio.

In Figures 6, however, Epidemic outperforms RiGoL in terms of average latency. In Epidemic, a significant proportion of networks replicate and store messages, thereby accelerating the process of message dissemination. In contrast, RiGoL limits the replication rate, causing only a small fraction of nodes to be able to replicate the messages, thus delaying message delivery (as indicated by a high average latency value).

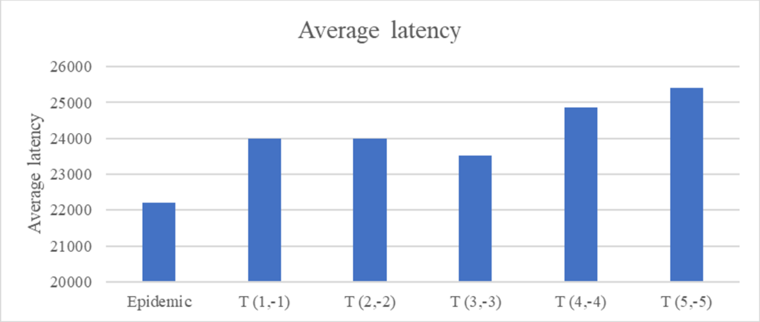

Fig. 6. Average latency.

Subsequently, we examined the influence of RiGoL implementation on energy consumption. A sample is obtained from node id 34, which is identified as a hub node in the Haggle-3 Infocomm 5 dataset. We presume that each node consumes a certain amount of energy whenever it transmits a message or discovers its neighbours. In this study, however, we assume that the energy required for transmit is greater than that required for neighbour discovery.

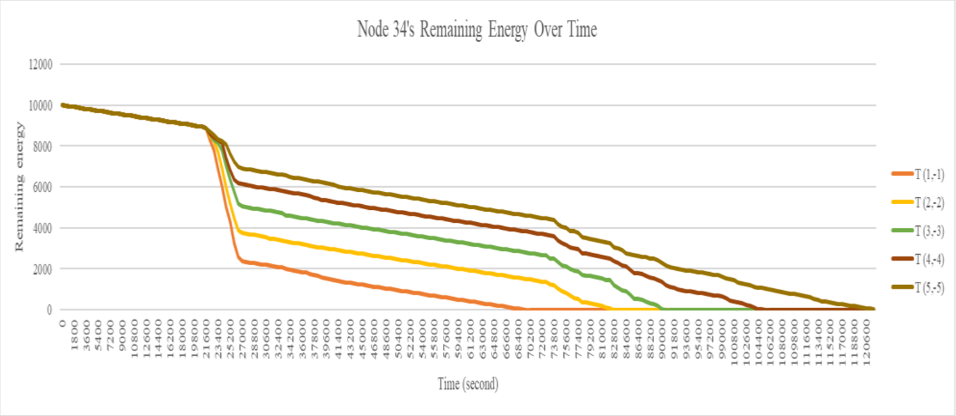

Fig. 7. Node's 34 remaining energy.

As shown in Figure 7, RiGoL outperforms Epidemic in terms of conserving energy for node id 34. Furthermore, as the RiGoL threshold increases, node id 34 exhibits an extended lifespan inside the networks. This phenomenon can be attributed to the implementation of RiGoL, which effectively reduces the number of duplicate messages inside networks. Consequently, the quantity of messages that nodes are required to relay or transmit has considerably decreased, resulting in energy conservation for this node.

5 Conclusions

This paper introduces RiGoL, a replication technique that reduces message replication in OMNs based on the Game of Life. We compare the proposed algorithm to Epidemic as a benchmark. Our findings show that the Epidemic outperforms RiGoL in terms of average latency. RiGoL, on the other hand, is more effective than Epidemic at reducing message replication in networks, as indicated by a smaller number of relayed messages and lower overhead ratio. Finally, we compare various RiGoL thresholds $T(T_{\text{forward}}, T_{\text{drop}})$. The maximum stated threshold T(5, -5) produces fewer replicated messages than the lower threshold T(1,-1) because nodes have to encounter more nodes without the message in order to duplicate it. Furthermore, the implementation of RiGoL also enhances the lifetime of node in the networks due to the decreased demand for message transfer. The decrease in message transmission reduces the individual node's energy consumption and extends the hub node's lifetime in the networks.

References

1. S. Misra, B. Kumar, S. Sujata, B. K. Saha, and S. Pal, *Opportunistic Mobile Networks* (Springer International Publishing, Switzerland, 2016)
2. C. Boldrini and A. Passarella, in *Mob. Ad Hoc Netw. Cut. Edge Dir. Second Ed.* (2013)
3. D. Vahdat, Amin; Becker, *Epidemic Routing for Partially-Connected Ad Hoc Networks* (Durham, 2000)
4. A. Adamatzky, Game Life Cell. Autom. 1 (2010)
5. A. Keränen, J. Ott, and T. Kärkkäinen, in *Proc. 2nd Int. Conf. Simul. Tools Tech.* (ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, BEL, 2009)

6. A. Grundy and M. Radenkovic, 2010 IEEE 6th Int. Conf. Wirel. Mob. Comput. Netw. Commun. WiMob'2010 324 (2010)
7. T. Spyropoulos, K. Psounis, and C. S. Raghavendra, Proc. ACM SIGCOMM 2005 Work. Delay-Tolerant Networking, WDTN 2005 252 (2005)
8. S. Annadurai, 2021 6th Int. Conf. Recent Trends Electron. Information, Commun. Technol. RTEICT 2021 829 (2021)
9. M. Ghosh, R. Kumar, M. Saha, and B. K. Sikdar, Proc. - 2018 IEEE Int. Conf. Autom. Control Intell. Syst. I2CACIS 2018 52 (2019)
10. F. K. Wen, G. W. Chung, and G. K. Hoon, IEEE Int. Conf. Artif. Intell. Eng. Technol. IICAIET 2020 1 (2020)
11. L. Yang, J. Zheng, Y. Cheng, and B. Ran, Phys. A Stat. Mech. Its Appl. **535**, 122277 (2019)
12. J. Scott, J. Crowcroft, P. Hui, and C. Diot, in *WONS 2006 Third Annu. Conf. Wirel. On-Demand Netw. Syst. Serv.* (2006), pp. 78–86