ASIAN JOURNAL OF INFORMATION TECHNOLOGY
Journal Home Page: https://medwelljournals.com/journalhome.php?jid=1682-3915

Modelling a Gossip-based Protocol for Enhanced Dynamic Routing

Elliot Mbunge, Stephen G. Fashoto and Simphiwe Dlamini
Department of Computer Science, Faculty of Science and Engineering, University of Swaziland, Kwaluseni, Swaziland

Key words: Disseminating, properties, algorithm, knowledge, routing

Corresponding Author:
Elliot Mbunge
Department of Computer Science, Faculty of Science and Engineering, University of Swaziland, Kwaluseni, Swaziland

Page No.: 203-206
Volume: 18, Issue 8, 2019
ISSN: 1682-3915
Asian Journal of Information Technology

Abstract: Gossip-based protocols for group communication have attractive scalability and reliability properties. The gossip schemes studied so far typically assume that each group member has full knowledge of the global membership and chooses gossip targets uniformly at random. In this study, we present a design of a routing algorithm that has the properties of gossip style of disseminating link state updates with the intention of making it faster than the current algorithms.

INTRODUCTION

Technology around the world has been moving in one direction like the way we experience time, one of the greatest factors that have propelled this is communication. Networks have played a very vital role, users get to send and receive messages across continents within a short space of time. The never ending the need for proper networking means is currently forcing some parts of the world to migrate to IP v6 because of the unavailability of IPv4 addresses. China and India with 36% of the world's population between them have only 8% of the IPv4 address space summing up to only 9.75 million IPv4 addresses. These changing networking requirements have been firmly supported by dynamic routing and its evolving protocols to meet the demands. Development of high performance and reliable networks has been influenced by factors like the latter, sparking an evolution. Different kinds of networks are being explored and the sudden emergence of networks that can be deployed anywhere with minimal infrastructural costs and highly dynamic, meaning that the topology changes anytime and nodes are added and removed spontaneously. This will require very robust algorithms and the gossip-based ones seem to be suited for the task. The current protocols we deem efficient cannot sustain us for eternity as they have also replaced previous versions. Dynamic routing can be defined as an adaptive link state routing whereby the algorithms used to route the data packets change their routing decisions in response to changes in both topology and traffic on a network (Tanenbaum and Wetherall, 2011). Link state routing algorithms are of a new age or are the latest and currently being used for routing of packets in a network, the general idea behind is simple and they propel a router to do the following:

- Discover neighbors and know their network address
- Calibrate the distance from router to each of these neighbors
- Create a data packet with all information gathered in previous steps
- Send the data packet in order to receive packets from other routers
- Compute the shortest path to every other router

With the above successfully implemented, a router knows the whole topology of the network in real time, the router is aware of changes that occur and finds alternative routes for data packets if necessary.

In static routing, all data packet routes are known beforehand and they do not change unless it is necessary and the paths are re-configured after that change in the network topology has been made. In dynamic routing, the packet routes are also known beforehand as these algorithms map the network topology, the key difference is that packets here are able to take different routes to the same destination, in any case, the network structure changes or there is traffic in the usual path a packet to a certain destination takes.

In dynamic routing, there is a huge demand for real-time communication of the routers in order for algorithms to make decisions based on the latest feed from routers in the network. Networks are getting larger it becomes more and more difficult to route packets dynamically. If all nodes are not using the exact same map in real-time communication, a routing loop occurs. It is also worth mentioning that in a large network, convergence is not easily achieved.

The aim of the study is to develop a model of a gossip-based link state dynamic routing protocol that is able to disseminate information on real-time between routers in order to increase the efficiency of dynamic routing while reducing packet loss and routing loops.

Literature review

Dynamic routing: Dynamic routing protocols have been used in networks since the early 1980's. The first routing protocols to be developed were Distance Vector routing protocols, each router maintains a table with a known distance to each destination and also has a link that will be used to get there. The distance is defined in terms of hop count and direction, it is feasible to say that distance vector protocols used the routers in the networks as signposts as each router only knew the distance to a destination as they did not have a map of the network topology. Distance Vector routing protocols had a shortcoming called the 'count infinity problem' thus we integrated to Link State routing protocols (Tanenbaum and Wetherall, 2011). Unlike distance vector routing protocols, these allowed a router to create a complete topology of the network. All routers possess a map of the network, updates are only sent when there is a change in the network topology.

Convergence: Convergence is achieved when all routing tables in routers are in a state of consistency, it is one of our primary goals to have an impressive convergence time in dynamic routing as the network gets larger or the topology changes and routing tables need to be updated. As a way to achieve feasible convergence time, the update must send throughout the entire network in a very fast rate in order to maintain consistent and efficient convergence standard.

Flooding in link state protocols: Generally, networks are not fully connected, flooding is used for Link State Updates (LSU) (Cheung, 1997). Flooding is a technique used to distribute data packets across networks, every incoming packet is sent out on every outgoing line except the one it arrived on and ultimately, flooding the network.

The gossip style of information dissemination: Information spreads in society at great speeds reaching almost everyone without a central coordinator, spreading information in gossip style has been inspired by epidemics, human gossip and social networks. Anyone can start a rumor but none can stop one, this is how the gossip-based protocol is proposed to work. A node is periodically selected at random from a set of nodes and exchanges the information, nodes receiving the information do the same (Kumar and Pahuja, 2014). The message here is sent only to selected nodes. In this study, we considered three important states in modelling a gossip-based model to disseminate information over the network namely: susceptible, infectious and removed. Susceptible means before the node has caught the disease it is susceptible to be infected by its neighbors. Infectious means once the node has caught the disease, it is infectious and has some probability of infecting each of its susceptible neighbors. Lastly, removed means after a particular node has experienced the full infectious period, it is removed from consideration since it no longer poses a threat of future infection.

MATERIALS AND METHODS

Types of gossip style of information dissemination

Anti-entropy (SI Model): In this model, a node is always susceptible or infective. Infective node shares information in every cycle, there is no termination and it sends an unbounded number of messages (Lopez, 2016). The algorithm in this model makes use of Boolean parameters we will call push and pull (Jelasity, 2011).

PUSH: Infective nodes sending to susceptible nodes.

PULL: All nodes are continuously pulling for updates as they cannot know the new update in advance. The nodes alternate between Susceptible (S) and Infectious (I) states.

We describe the SI model in the following way:

- Initially, some nodes are in the I state, and all others are in the S state
- Each node v that enters the I state remains infectious for a fixed number of steps t_1
- During each of these t_1 steps, v has a probability p of passing the disease to each of its susceptible neighbors

After the t_2 steps, node v is no longer infectious and it returns to the S state.

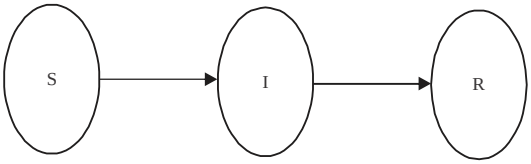

Fig. 1: Gossip-Based SIR Model

Rumor mongering (SIR Model): A node loops in a Susceptible (S), Infectious (I) and Removed (R) cycle and exhibits the following:

- Initially, some nodes are in the I state and all others in the S state
- Each node v that enters the I state remains infectious for a fixed number of steps t_1
- During each of these t_1 steps, v has probability p of passing the disease to each of its susceptible neighbors
- After t1 step, node v is in state R, it is no longer in I or S and can no longer catch or transmit the disease

This model is more frequent than the SI's cycles as they demand lesser resources. Rumor mongering spreads updates fast with low traffic network (Lopez, 2016). SIR Model tackles the termination problem as we want it to be optimal which makes the optimality highly involve the nodes receiving the Link State Update.

Gossip-based algorithm modelling: Our study makes basic and arbitrary initial conditions in modelling the gossip algorithm on the mathematical model of the algorithm in the pseudocode. The aim of the study is to model a gossip algorithm for disseminating link state updates, the researcher bases the algorithm on the SIR model with the following assumptions:

- The number of nodes is fixed, there are N nodes, where N = S+I+R
- A node is in one of three states, susceptible, infectious and removed

A node in the S state is susceptible to getting an update from other nodes, a node in I state has received an update and it is 'sharing' it to other nodes randomly. In R state, the node has received an update and infected it to other nodes as required.

In Fig. 1, a node I with an update does the following:

- Node I with an update is in I state
- Node I picks a neighboring node S at random
- Node I sends the update to node S if node S is in S state
- Node I randomly picks neighbors until it is done, it then enters the R state

On receiving the update, node S does the following:

- Node S checks whether the received update is there or not, if it is there, the received update is discarded

In this study, a compartmental form of mathematical modelling is adopted to construct differential equations that showcase the behavior of the nodes between the three states over time. In susceptible nodes, the rate at which new infections occur is:

$$\text{Rate}(I) = \beta SI \tag{1}$$

For some $\beta>0$. When a node is infected with the update, it changes states from S to I. In this model, removed nodes enter the S state in order to receive new updates, the rate at which R state nodes change to S state is given by:

$$\text{Rate}(S) = \alpha R \tag{2}$$

Therefore, our first differential equation for the Susceptible (S) state is written as:

$$\frac{dS}{dt} = -\beta SI + \alpha R \tag{3}$$

Infectious nodes have the same rate at which infections occur added to them. The infectious nodes also change state at the rate:

$$\text{Rate}(I) = \gamma I \tag{4}$$

Where, γ is a constant. Therefore, the second differential equation for the Infectious (I) state is written as:

$$\frac{dI}{dt} = \beta SI - \gamma I \tag{5}$$

The third equation shows the rate at which I nodes enter the R state with the previously defined conditions still holding, hence, the last differential equation for the Removed (R) state is written as:

$$\frac{dR}{dt} = \gamma I - \alpha R \tag{6}$$

The solutions of these equations in the study use Runge Kutta 4 equations to get numerical solutions that were used to study the behavior of the nodes over time in different states over time.

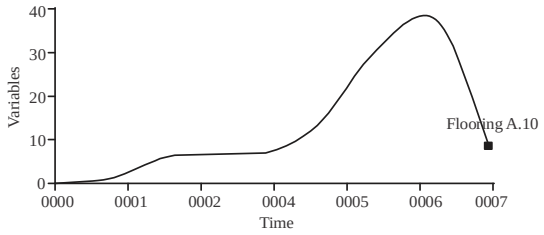

Fig. 2: Flooding %CPU usage vs. time (seconds)

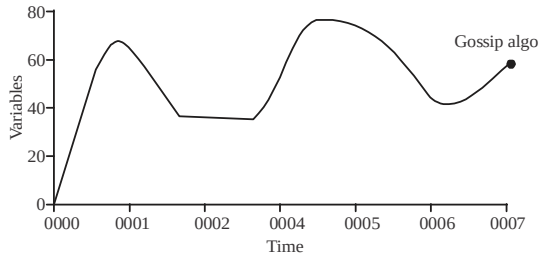

Fig. 3: Gossip %CPU usage vs. time (seconds)

RESULTS AND DISCUSSION

Implementation and results analysis: The development tools used to implement the gossip and flooding algorithm are R, Mathcad and Java. The study evaluated the modelled gossip algorithm against the flooding algorithm. In three nodes, where one node enters I state and gossips with the other two that it is adjacent with, Fig. 2 and 3 evidently show that gossip uses up a lot of percentage CPU usage over time than the flooding.

The gossip-based algorithm has a faster execution time than flooding as disseminating the information to a neighbor and in specificity has beaten disseminating redundantly across the whole network. In the three nodes in this study, we understands that flooding disseminates without rigorous computation that gossip goes through of finding the state of the nodes hence having higher CPU percentage usage. The gossip style executed faster, meaning it spread or reached nodes and completed the same task assigned to flooding in lesser time, despite higher CPU usage, having a target rather than spreading randomly made it execute faster.

CONCLUSION

Flooding of LSU packets across a network has had notable disadvantages, costly bandwidth and duplication of packets that waste resources in the network. The gossip style of disseminating the LSU packets offered a faster and less demanding strategy in terms of resources and time. Instead of spreading dozens of packets across a network with the hope that they will reach the destination, the researcher selects nodes to randomly spread the update in a way that a rumor spreads among people. In theory, routing loops occurring will be lessened as the researcher would not have redundant duplicate packets circulating in the network an LSU packet always has a specific destination in the gossip which is not the case in flooding.

REFERENCES

Cheung, S., 1997. An efficient message authentication scheme for link state routing. Proceedings 13th Annual Computer Security Applications Conference, December 8-12, 1997, IEEE, San Diego, California, USA., pp: 90-98.

Jelasity, M., 2011. Gossip. In: Self-Organising Software: From Natural to Artificial Adaptation, Natural Computing Series, Serugendo, G.D.M., M.P. Gleizes and A. Karageorgos (Eds.). Springer, Berlin, Germany, pp: 139-162.

Kumar, A. and S. Pahuja, 2014. A comparative study of flooding protocol and gossiping protocol in WSN. Int. J. Comput. Technol. Appl., 5: 797-800.

Lopez, E., 2016. Continuous learning: Introduction to gossip. WordPress, Automattic Company, San Francisco, California, USA.

Tanenbaum, A.S. and D.J. Wetherall, 2011. Computer Networks. 5th Edn., Pearson Education, Boston, Massachusetts, USA., ISBN: 9780132126953, Pages: 933.