Gossip-based Aggregation blah

Niklas Semmler, Jiannan Guo
KTH Royal Institute of Technology (Sweden)

Abstract

Motivated by increase of peer-to-peer network and *ad hoc* sensor networks, we study distributed algorithm, also known as gossip algorithm, which is for exchanging information and message dissemination. As aggregating network information and enabling local access to this information are crucial to manage a large scale networks, gossip-based aggregation facilitates management of networks in a completely decentralized manner. This study focuses on the extracting the relation between convergence speed and connectivity of graph, addressing several real network topologies.

Keywords
Gossip-based aggregation, performance

[10]

Contents

1 Introduction

1.1 Background

Large scale computer networks have been an uptrend in these years and will require a more scalable and reliable solution.
Traditionally, with periodically poll and centralized management system, some variables of the network, such as throughput, workload and bandwidth, are gathered through the whole network. Apart from several apparent drawbacks such as single point failure and performance bottleneck, centralized management approach usually cannot fulfill the need of mission-critical business processes, which will often need to deal with load changes and failure [11]. Distributed algorithms are utilized to tackle these problems by propagating messages through the whole network in completely decentralized manner. Another advantage along with the solution is the local accessibility of global information [3].
Gossip-based algorithms, also known as epidemic-style techniques [2], are used to deal with drawback of deterministic algorithms in exchange of certainties. On another hand, they are structureless algorithms, comparing to some other distributed protocols such as Echo algorithm proposed by Segell [9]. Although, gossip-based algorithms could be combined with tree-based algorithms to reach an optimization for certain topologies [6]. Utilizing this algorithm, aggregation could be performed to collect data through the whole network and compute global information on every node. Underlying topology has been proved by many works to have a strong impact on the performance of gossip-based algorithm [7] [3], although little works exist regarding running gossip-based aggregation algorithm in real network topologies. This study would focus on investigating the divergence of performance in different real network topologies, namely Reuna, BREN, Geant, Iij [5].

2 Chapter 2

2.1 Related work

Motivated by peer-to-peer and *ad hoc* networks, a considerable number of studies have been done regarding gossip-based algorithms. Convergence and upper bound consensus time have been proved by J.Lavaei and R.Murray in [7]. Analytical methods and simulations have been utilized to discuss the relations between performance of gossip protocols and topology of network, namely randomness, connectivity etc. As a subcategory of distributed algorithm, different models, such as synchronous and asynchronous models, with or without churn, are discussed in [8]. The optimization of parameters of an asynchronous randomized gossip algorithm for fasted convergence is proved to be semi-definite problem [1].
This study is mainly based on the work of M.Jelasity, A.Montresor and O.Babaoglu [3], focusing on existing static networks of different topologies [5]

2.2 Gossip-based aggregation

A simple implementation of synchronous gossip-based aggregation algorithm, inspired by [3] can be illustrated by following pseudo code,

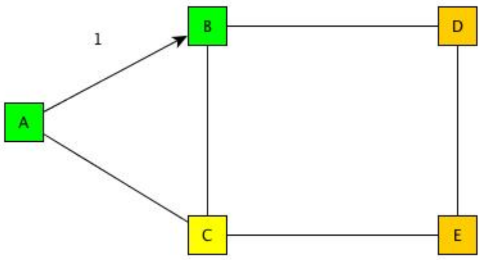
```
ActiveGossipThread
    for each consecutive *delta* time units at randomly picked time; do
        q <- GET_NEIGBOUR()
        send state_p to q
        state_q <- receive(q)
        state_p <- UPDATE(state_p, state_q)

PassiveGossipThread
    while true:
    state_q <- receive(*)
    send state_p to sender(state_q)
    state_p <- UPDATE(state_p, state_q)
```

For better understanding of aggregation mechanism, we assume a graph as in Figure

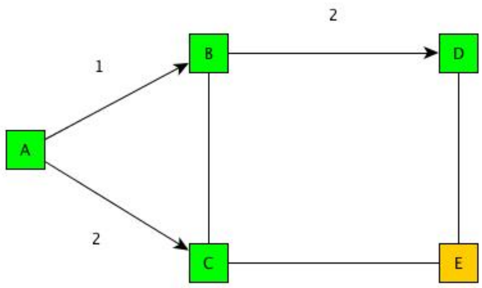
Although convergence is proved and expected convergence time can be estimated by probability density function for a certain topology [7], a drawback of probabilistic algorithm comparing to deterministic algorithm is reliability [8]. The value can only be considered as true result at a certain probability. To put this protocol into practice, some extra procedures need to be added. In this study, we leave out the test of correctness inside implementation but try to obtain a empirical criteria according to the result of experiments.

2.3 Impact of topology toward the performance of aggregation

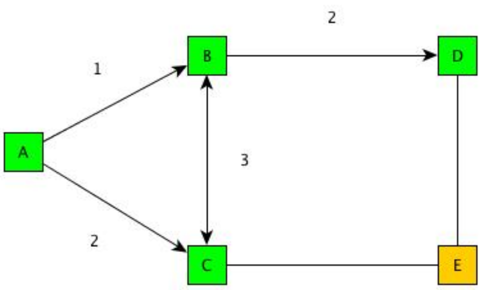
A plenty of methods exist to describe overlay topologies of a network, and different representations are used to describe properties of a topology. On the other hand, the performance of gossip-based aggregation is also abstracted differently for specific purpose. In this study, we firstly focus on investigating the impact of various number of links with the number of nodes unchanged over the performance of gossip-based aggregation algorithm. Then we try to extract proper parameters representing a unweighted undirected connected graph through applying the definition of entropy presented by [4] and [10] and then we apply inductive research method [?] to address the relation between entropy and convergence time.



Beginning of message dissemination (a)



Passing the message (b)



Probable redundant message (c)

Figure 1: Gossip algorithm evolution cycles

Name of Network	Year	Country	# of Links	Entropy1	Entropy2
Reuna	2010	Chile	36	202.0476	0.5164084
BREN	2010	Bulgaria	38	210.2229	0.6214745
Geant	2010	Europe	58	277.3282	1.530375
Iij	2010	Japen, USA	66	307.2482	1.553654

Table 1: Topologies under investigation

2.4 Applying to real network

In order to apply control variable experiment method, we based our experiment on 4 real network topologies with same number of nodes (37) and different number of links, as showed in Table 2.4. Entropies are also calculated by applying methods presented by [4] (referred in the table as Entropy1) [10] (referred in the table as Entropy2).

2.5 Hypothesis

Hypothesis 1. In a given network topology $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$, the number of links (edges) N_E and convergence time t are positively correlated, with number of nodes (vertices) N_V unchanged.
Hypothesis 2. In a given network topology $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$, the entropy of this graph S and convergence time t are positively correlated.

2.6 Research methodology and methods

We are creating a positivistic investigation grounded in quantitative experiments. With induction in the form of statistical inference, we aim to predict the behavior of the system under question in the boundary of our experimental research.

We clearly distinguish ourselves from realism, as we use network graphs inspired by real networks, but leave it to other research to prove that the experiment is representative of those real networks.
Since we will investigate several existing networks with the same number of nodes (constant) but different number of links (variable), we choose experimental research as our research method.
We will apply inductive reasoning research approach to verify the hypothesis. Since our unique contribution is an investigation of real network rather than a theoretical proof, deductive reasoning research approach won?t fit our goal properly. We will run our experiment multiple times to collect data and based on the observation of these phenomena we can obtain our conclusion.

3 Chapter 3

Implementation * Topology Zoo * Autonetkit * Emulator * Python Daemon
graphml -> configs -> uml machines

3.1 Simulation/Emulation of gossip-based aggregation

To evaluate the value of the gossip-based aggregation it is necessary to see it in the wild of a real network environment. Unfortunately the setup of a real network is to time and cost intensive. Hence the next best thing was chosen: A network emulation. In a network simulation the network is simplified so as to illuminate a certain set of behaviours. (REF ns) An emulation in contrast consists of all the elements of the real network only that the resources are not "real" but "virtual". While for example an emulated switch may process network traffic in the very same way as a switch in your office, it could be implemented as a linux process. In this research the netkit emulation framework (itself a wrapper to user mode linux) was chosen. Netkit (REF Netkit) creates virtual linux machines from a set of configuration files. These virtual machines are automatically connected through usermode processes called "uml_switch". It is possible to logon to the machines via ssh and configure them just like any other linux machines. This allows great freedom to try and test software which otherwise would be hard to implement. The initial set of configuration files were created with the tool Autonetkit using network graphs from the "Topology Zoo". The "Topology Zoo" is a collection of around two hundred fifty graphs of telecommunication networks. The graphs are open for use and give a good idea on the reality of communication networks. Simon Knight, one of the founder of the topology zoo has developed the tool Autonetkit. It takes a graph xml file (graphml) and constructs over a series of abstractions configuration files from OSPF to DNS. For the purpose of this experiment the number of assumptions are created to take only the relevant creating a number of assumptions an emulation In contrast to a simulation a network emulation contains

3.2 Topology Zoo

* "two hundred and fifty networks in the Zoo" * communication networks of real telecommunication providers * gallery and graphml files * processed to remove multiple links between nodes (simplification)

3.3 Autonetkit

* By Simon Knight, also worked on topology zoo * Translates graph file to linux configurations files (Qugga, Scripts, ...) * multiple layers of abstraction * we customized this so that every node stores a list of all it's neighbours and corresponding IP addresses

3.4 Implementation of gossip-based aggregation protocol in virtual networks

3.4.1 Netkit & UML

* Netkit a framework to do network experiments * What is an emulation * Uses User Mode Machines, the linux kernel is started as a userlevel process * Combines kernel with a filesystem used mostly for pedagogic purposes * network is created by using processes as switches between the linux kernel processses

pseudo code -> python scripts -> csv -> R scripts -> graphs

3.4.2 Gossip Daemon

* written in python * based on a modular structure of socket, epoch and state modules * difficulties in combining multithreading with network

3.4.3 R scripts

...

4 Future works

4.1 More comprehensive emulation

Since we assume an undirected unweighted static topology, some other crucial properties of a real network are not reflected in our study, such as link speed and congestions. Although, in real backbone networks, node leaving and joining are not likely to happen, but packet loss and delay could be modeled into churn mechanism and should be considered in future work.

4.2 Relation between the performance and other properties of graph

In [3], convergence factor $E(2^{-\phi})$ is used to determine convergence time, smaller convergence factor results in faster convergence. The Watts-Strogatz is used to model the topology of overlay network, indicating randomness as an independent variable [12]. Thus, a function is derived, with randomness parameter β as input and convergence factor $E(2^{-\phi})$ as output.
Although, to apply this model in a real network, reverse procedures need to be done to determine randomness parameter β of a given graph $\mathcal{G} = \{\mathcal{E}, \mathcal{V}\}$. Mapping this back to the study of [3] can give a better understanding how topology impact convergence time.
Other than this factor, graph clustering [?], max-flow and min-cut [?] could also be derived from these graphs and relationship between them and gossip algorithm performance can be further discussed.

References

[1] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Analysis and optimization of randomized gossip algorithms. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 5, pages 5310–5315 Vol.5, 2004.

[2] Indranil Gupta, A-M Kermarrec, and Ayalvadi J Ganesh. Efficient and adaptive epidemic-style protocols for reliable and scalable multicast. *Parallel and Distributed Systems, IEEE Transactions on*, 17(7):593–605, 2006.

[3] Márk Jelasity, Alberto Montresor, and Ozalp Babaoglu. Gossip-based aggregation in large dynamic networks. *ACM Transactions on Computer Systems (TOCS)*, 23(3):219–252, 2005.

[4] Li Ji, Wang Bing-Hong, Wang Wen-Xu, and Zhou Tao. Network entropy based on topology configuration and its computation to random networks. *Chinese Physics Letters*, 25(11):4177, 2008.

[5] Simon Knight, Hung X. Nguyen, Nick Falkner, Rhys Bowden, and Matthew Roughan. The internet topology zoo. *Selected Areas in Communications, IEEE Journal on*, 29(9):1765–1775, 2011.

[6] Pradeep Kyasanur, Romit Roy Choudhury, and Indranil Gupta. Smart gossip: An adaptive gossip-based broadcasting service for sensor networks. In *MASS*, pages 91–100. IEEE, 2006.

[7] J. Lavaei and R.M. Murray. Quantized consensus by means of gossip algorithm. *Automatic Control, IEEE Transactions on*, 57(1):19–32, 2012.

[8] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1996.

[9] Adrian Segall and Inder S. Gopal. Distributed name assignment in computer networks. *Computer Networks*, 17:127–139, 1989.

[10] Ricard V. Sole and Sergi Valverde. Information theory of complex networks: On evolution and architectural constraints. In *In*, pages 189–210. Springer-Verlag, 2004.

[11] Rolf Stadler. Protocols for distributed management. Technical Report 2012:028, KTH, Communication Networks, 2012. QC 20120604.

[12] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.