Review of the paper: “EPIDEMIC ALGORITHMS FOR REPLICATED DATABASE MAINTENANCE by Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry”

Reviewer: Varun Kesharaju

I. Comprehension
    A. Clearinghouse Servers on the Xerox Corporate Internet are a cluster of several hundreds across the globe connected with ethernets and gateways.
    B. Traditional direct mail and anti-entropy methods to maintain consistency across domains of information have failed. Anti-entropy algorithm ran on each domain each night, generating traffic which was way beyond the handling capacity of the network.
    C. In addition to that, the remailing step followed anti-entropy, to send the correct database value due to conflicts that arose between servers which participated in the anti-entropy step.
    D. Since remailing was very inefficient, that was disabled. Yet, critical links would be overloaded by anti-entropy.
    E. This paper addresses the challenge of maintaining consistency in replicated databases across many sites in large networks and attempts to reduce the network load imposed by the Clearinghouse update process by exploring spatial distributions and rumor mongering algorithms.
    F. Analogies and results from the theory of epidemics were introduced as simple yet robust communication systems to improve performance and consistency.
    G. Effective flavors of anti-entropy algorithm and rumor mongering method were developed based on mathematical analysis of simulations and practical implementations.
    H. It was concluded that randomised epidemic based algorithms were more effective in maintaining replicated database

consistency while reducing network traffic when compared to deterministic algorithms.

I. The methods introduced are scalable. By using certain spatial distribution for selecting anti-entropy patterns, the average traffic per link and traffic on certain busy links was reduced significantly.

II. Critique

A. This paper shows the robustness of anti-entropy and rumor mongering epidemic based approaches. I think this is only true when the network traffic and load distribution among the servers are uniform. Heterogeneous network conditions which include variable bandwidth, node failures and downtime due to heavy load have to be taken in consideration. The *ResolveDifference* step during Anti-entropy, which runs between each pair of randomly chosen sites can be inefficient when the chosen site s’ is temporarily under heavy load due to some other user activity. It would be beneficial to resolve differences with this site when the load decreases, which might in fact be faster. Else we would be bombarding the site and might potentially bring it down, while there might be other underutilised nodes that are not updated. Poorly connected node-to-node links would also affect the runtime of the update process. Topology was not considered during the discussion of epidemic approaches for replicated data consistency. I believe that the random algorithms discussed have to be implemented with capability to adapt according to the dynamically changing network conditions.

B. The paper discusses using latest computed checksums to compare the full database once the recent update vector list is exchanged. I believe that this process can be further broken down. Partition the data and maintain checksums for each of the partitions. If checksums related to a particular partition mismatch, we can narrow down the list of tables and data segments where there is inconsistency and update only those. This method might work best when we have independent partitions(on which checksums are calculated) on groups of tables that are dependent. This proposal might increase the compare traffic and meta data exchanged, but would significantly decrease the update traffic.

C. Anyway such checksum strategy works only when there are moderate write updates. Exchanging update vectors would be very inefficient when there is heavy write traffic. Some other optimization techniques should be adapted dynamically when the size of the update vector increases above some threshold or when it is easy to determine that there are a huge number of write updates within a time frame.

III. Synthesis

A. Machine Learning for Intelligent Partner Selection

1. Motivation and proposal:

a) In a probabilistic approach, there might be cases when a pair of sites could be selected again even though there are other sites that await update. State of the node is not considered during the selection process which could result in inefficiencies by generating redundant data exchanges and network traffic.

b) Uniform and probabilistic methods do not account for dynamic changes in topology and propagation delays, which are hard to ignore in a real-world large replicated database.

2. Proposal and contributions:

a) I think the random selection of sites during the update process in both rumor mongering and anti-entropy steps could be significantly improved by using machine learning techniques. Intelligently selecting nodes reduces unfruitful exchanges within the network.

b) Idea is to predict nodes that are most likely to be out of sync and prioritize them for updates, while considering the time that it takes to complete the update.

c) Scalability enhances as the system becomes more adaptive.

3. Methodology:
    a) Training a model based on historical update pattern, point-to-point bandwidth availability, node status in terms of both network connectivity and load, propagation delays, and network topology as few features would provide partners optimally which in turn speeds up update convergence.
    b) Selection of the optimal machine learning model has to be tested out with multiple simulations and comparisons iteratively.