2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Evaluating the Usefulness of the Metropolis Algorithm for Overlay Network Optimization

Tatsushi Takamura, Tatsuhiro Tsuchiya and Tohru Kikuno

†Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871 Japan
Email: t-takamr@ist.osaka-u.ac.jp, t-tutiya@ist.osaka-u.ac.jp, kikuno@ist.osaka-u.ac.jp

Abstract—Some algorithms adopt the Metropolis algorithm to optimize the communication cost and fault tolerance of an overlay network. The intended advantage of using the Metropolis algorithm is the avoidance of getting trapped in local optima; however there has been no convincing evidence. This paper studies one such algorithm to examine it.

1. Introduction

Peer-to-peer systems use an *overlay network*, a virtual network built on top of the physical network, to route messages to destination. Overlay networks can be classified into two types: *structured* and *unstructured*. Structured overlay networks are tightly controlled and constructed by a coordination mechanism, such as a distributed hash table (DHT). Structured overlays can efficiently search data and disseminate messages but exhibit relatively low robustness to node failures or frequent churn.

Unstructured overlay networks maintain only the list of addresses of neighbor nodes at each node and rely on no rigid mathematical structure. They are typically constructed without taking the physical network topology into consideration. This may cause a significant mismatch between the physical network topology and the overlay network topology, which in turn results in a large volume of redundant traffic [1].

To solve the topology mismatch problem, some algorithms adopt to optimize the overlay topology: Using the Metropolis algorithm, these algorithms iteratively reshape the overlay topology so that the mismatch can be gradually reduced [2, 3].

The Metropolis algorithm is originally used in Monte Carlo simulations to obtain random samples from a probability distribution. The idea is to perform random walk in the solution space so that the points on the walk are distributed according to the required probability distribution.

Its adaptation to topology optimization replaces random walk with a sequence of random, small topology changes. Each node in a network periodically initiates a local topology change. If the change leads to a better network topology, then the change will be accepted and performed. On the other hand, even if the change results in a worse topology, the change will be accepted with some probability to avoid getting stuck at local optima.

The simulation results presented in [2, 3] report that the algorithms adopting the Metropolis algorithm exhibit good performance. However there has been little evidence of the benefit of using the Metropolis algorithm. For example, it has been unclear whether the Metropolis algorithm performs better than simple hill-climbing which takes no consideration in avoiding trapping into local optima. To overcome this lack, this paper evaluates the effects of using Metropolis scheme on the performance and resiliency of an overlay. Specifically we focus on Localiser [2], one of the overlay construction algorithms adopting the Metropolis algorithm.

2. Unstructured Overlay Networks

We model the overlay network topology as a undirected graph $G = (V, E)$ with a vertex set V and an arc set E. A vertex corresponds to a node and an arc corresponds to a link. We let n denote the number of nodes; i.e., $n = |V|$. The physical proximity is modeled by a communication cost function $c : V \times V \rightarrow \mathbb{R}$. $c(i, j)$ represents the communication cost between nodes i and j. We assume $c(i, j) = c(j, i) > 0$ for any $i, j (\neq i) \in V$.

We evaluate the overlay network with respect to two properties.

- Locality awareness

  A large amount of traffic caused by peer-to-peer applications is routed along the path in the overlay network. In order to avoid overloading the network and to reduce message latency, the overlay should reflect the underlying network topology. Specifically, a node should have neighbors that are close to it in the physical network.

- Degree distribution

  The resiliency of the overlay network critically depends on the distribution of node degrees. The failure of a node with a smaller degree is more likely to break the connectedness of the network. Thus all nodes should have the same degree, unless node failures do not occur uniformly randomly.

3. Localiser

Localiser is the algorithm proposed by Massoulié et al. It relies on the Metropolis scheme to optimize the communication cost and fault tolerance of an overlay network [2].

The basic idea behind the Metropolis scheme is as follows: This is a simple iterative scheme for minimizing a function f on a domain D. Starting from any point $x \in D$, a move to a nearby point y is proposed. The move is then accepted with some probability which decreases as $f(y)-f(x)$ increases. The reason for allowing moves that may increase f is to avoid getting trapped in local optima. If the move is rejected, then the new point remains at x. The propose-accept process is repeated from the new point.

The global cost of an overlay network topology is evaluated by the following function.

$$f(G) = w \sum_{i \in V(G)} d_i^2 + \sum_{(i,j) \in E(G)} c(i, j)$$

where w is a weight parameter, d_i is the degree of node i and $c(i, j)$ is the communication cost between node i and node j.

In Localiser, each move corresponds to a local topology change, as illustrated in Figure 1. The topology change is iteratively initiated by each node. The outline of the algorithm for local topology change is as follows:

1. Choose two of its neighbors, say node j and node k at random, and measure the communication cost $c(i, j)$ and $c(i, k)$.

2. Send messages to node j and node k to request d_j and d_k. Nodes j and k send back respectively d_j and d_k. In addition, node j sends back its estimate of $c(j, k)$.

3. Evaluate the cost change resulted from replacing link (i, j) with (j, k) as depicted in Figure 1. The cost change Δf can be calculated locally by the following expression:

   $\Delta f = 2w(d_k - d_i + 1) + c(j, k) - c(i, j)$

4. Perform the replacement of the links with probability $p = \min\left(\left(e^{-\Delta f/T} \frac{d_i(d_i-1)}{d_k(d_k+1)}\right), 1\right)$.

Note that the number of links is conserved, because the algorithm only moves links.

Here the key parameter is T, which is used to compute the probability of accepting a “bad” move. This parameter is called temperature and is used to specify a trade-off between accuracy and speed of convergence. That is, if the temperature is high, then bad moves are accepted with higher probability, resulting in high capability of local optima avoidance. On the other hand, if the temperature is low, then an equilibrium is reached fast.

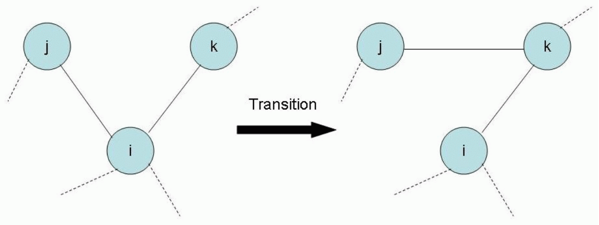

Figure 1: Localiser algorithm

4. Simulations

In this section we present detailed simulation results using PeerSim, a cycle based simulator. We assume that every node initiates a local topology change in each cycle. We evaluate Localiser in terms of the following three criteria: (1) average communication cost, (2) degree distribution, and (3) resilience to node failures. For each combination of parameter values, we perform 10 runs and average the obtained values.

4.1. Network Model

To obtain the communication cost function c, we construct physical two network topologies using the Waxman model [4] and the Transit-Stub model [5]. Each topology is composed of 100 routers to which end nodes are randomly attached.

We assign to every physical link 50ms as communication cost. The communication cost between two nodes in the overlay is then set to the summation of the communication cost along a physical path between the two nodes. If there are multiple paths, then the shortest one is selected to compute the communication cost.

We use Scamp [6] to build the initial topology of the overlay network. Scamp is a fully decentralized self-organizing membership protocol for building unstructured overlay networks.

4.2. Average Communication Cost

Figures 2 to 5 present the average communication cost between any pair of adjacent nodes. Figures 2 and 4 show the results for the network with 5000 nodes. Figures 3 and 5 present those for the network with 10000 nodes. The horizontal axis represents cycles, while the vertical axis represents the average communication cost between two neighbors. Different curves correspond to different temperature values T.

As can be seen in these figures, smaller T leads to fast convergence. Importantly, no case was observed where the network was trapped with local optima, even if T was very small. That is, smaller T always resulted in smaller communication cost. No clear qualitative difference is observed between the Waxman model and the Transit-Stub model and between different network sizes.

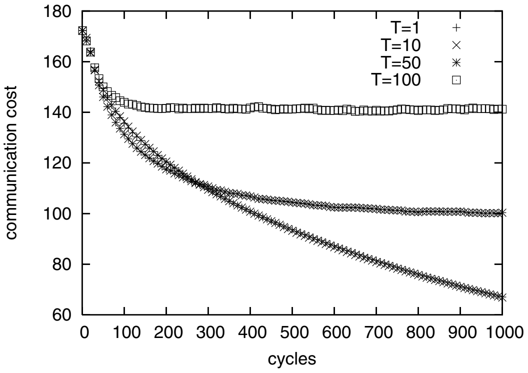

Figure 2: Average Communication Cost (Waxman, n = 5000)

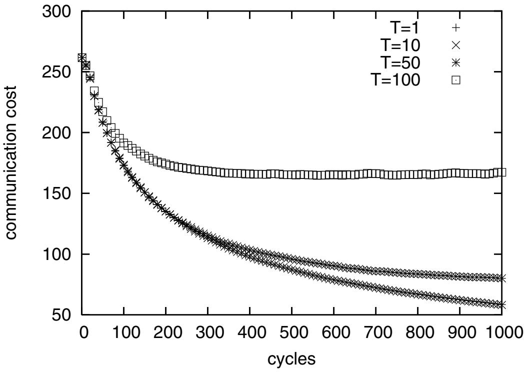

Figure 4: Average Communication Cost (Transit-Stub, n = 5000)

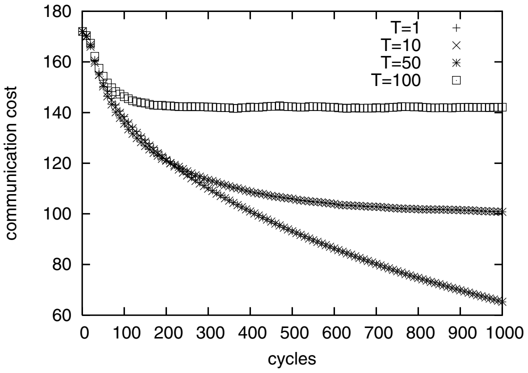

Figure 3: Average Communication Cost (Waxman, n = 10000)

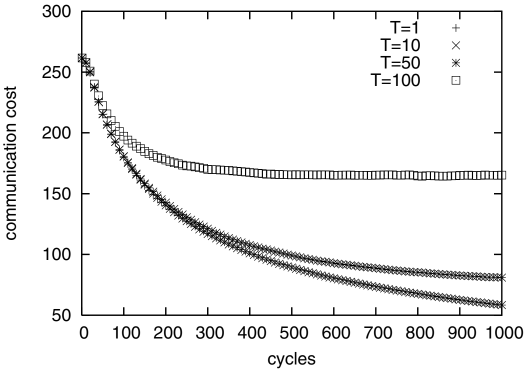

Figure 5: Average Communication Cost (Transit-Stub, n = 10000)

4.3. Degree Distribution

Here we present how uniform the degree distribution will be when Localiser is used. We compare node degree distribution between the initial overlay network and the optimized topology obtained when 300 cycles elapse.

Figure 6 shows the degree distribution for the Waxman model, while Figure 7 shows that for the Transit-Stub model. The number of nodes is set to 5000. These graphs have degree on the horizontal axis and the number of nodes with a particular degree on the vertical axis.

We varied the value of T but the results obtained were almost the same. The results shown in the figures were obtained when $T = 1$.

As can be clearly seen, node degrees concentrate near the average after 300 cycles. No clear difference is observed between the two physical network models.

4.4. Resilience to Failures

In order to assess the resilience to failures, we measure the number of network partitions in the presence of random node failures. In graph theoretic terms, a network partition is a connect component of the graph G with failed nodes and their incident links removed.

We compare the initial overlay network and the one obtained when 300 cycles elapse.

Figures 8 and 9 depict the results obtained when the Waxman model and the Transit-Stub model are used. The horizontal axis represents the ratio of failed node. The vertical axis represents the number of network partitions. The number of nodes is set to 10,000.

Again, we observed that the value of T has almost no effect on the results. Thus we only show the result obtained when $T = 1$.

The improvement in resilience is substantial in both models. For example, even if 50% nodes have failed, the optimized network does not loose the connectedness.

5. Conclusion

In this paper, we conducted simulation experiments to evaluate the effects of using the Metropolis algorithm in the context of overlay topology optimization. We focused on the Localiser algorithm which adopts the Metropolis algorithm for iterative topology reshaping.

To see the effects, we varied the value of T, the temperature parameter that dictates the trade-off between convergence speed and local optima avoidance. The results

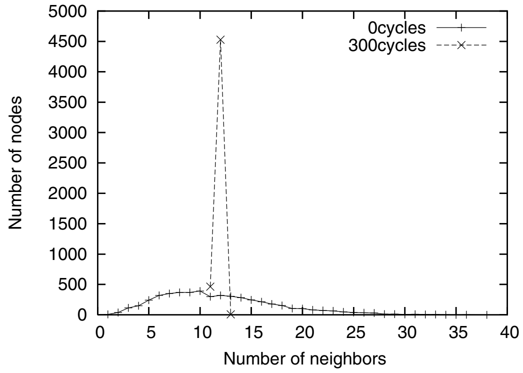

Figure 6: Degree Distribution (Waxman, n = 5000)

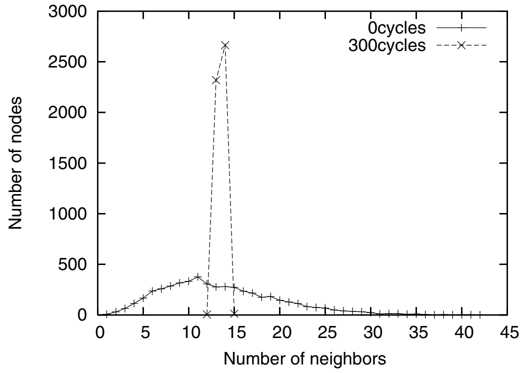

Figure 7: Degree Distribution (Transit-Stub, n = 5000)

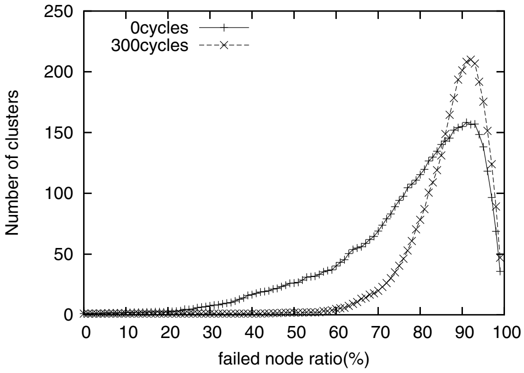

Figure 8: Resilience to Failures (Waxman, n = 5000)

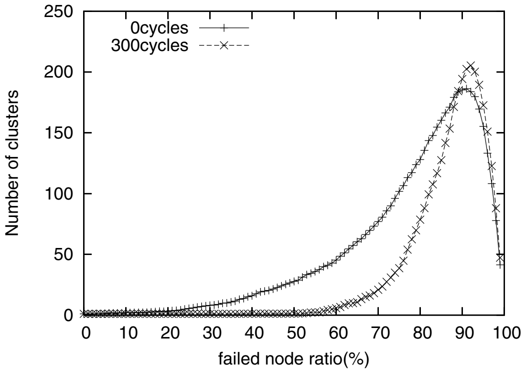

Figure 9: Resilience to Failures (Transit-Stub, n = 5000)

obtained indicated that smaller T always leads to a faster convergence to a state where the average communication cost is low. More importantly, no single case was observed where the optimization is trapped in local optima. This can be explained by the property of the Localiser algorithm: In the algorithm, every node can perform small topology change. Because of the large number of these possible moves, the network always has a good chance of selecting a move that improves the topology.

We leave as future work to see if the obtained results can be generalized to other algorithms that use the Metropolis algorithm, such as the SAP2P protocol [3]. To this end, we plan to conduct further simulation studies.

Acknowledgments

This work was supported in part by the MEXT Global COE program (Center of Excellence for Founding Ambient Information Society Infrastructure).

References

[1] M. Ripeanu, I. Foster, and A. Iamnitchi, “Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design,” IEEE Internet Computing, vol.6, no.1, pp.50–57, 2002.

[2] L. Massoulié, A.M. Kermarrec, and A.J. Ganesh, “Network awareness and failure resilience in self-organising overlay networks,” In Proc. of 22nd Symposium on Reliable Distributed Systems (SRDS 2003), pp.47–55, 2003.

[3] Z. Li, Z. Zhu, Z. Li, and G. Xie, “SAP2P: Self-adaptive and Locality-aware P2P Membership Protocol for Heterogeneous systems,” In Proc. of 16th Euromicro Conf. on Parallel, Distributed and Network-Based Processing (PDP 2008), pp.229–236, 2008.

[4] B.M. Waxman, “Routing of multipoint connections,” IEEE Journal on Selected Areas in Communications, vol.6, no.9, pp.1617–1622, 1988.

[5] E.W. Zegura, K.L. Calvert, and S. Bhattacharjee, “How to model an internetwork,” In Proc. of IEEE INFOCOM, pp.594–602, 1996.

[6] A.J. Ganesh, A.-M. Kermarrec, and L. Massoulié, “Peer-to-peer membership management for gossip-based protocols,” IEEE Trans. Comput., vol.52, no.2, pp.139–149, 2003.