

Performance evaluation of two Self-Adaptive Routing Algorithms in Mesh Networks

Miguel Angel Rojas González

Faculty of Mathematics and Computer Science
FernUniversität in Hagen
Universitätsstraße 27, PRG
58084 Hagen, Germany
miguel.rojas@fernuni-hagen.de

Abstract: Following the seminal work of Unger et al. based on building mesh-like structures on top of a P2P network, we introduced an improved version of *Compass* - a routing and load balancing algorithm based on direction-scopes for message delivery in such kind of networks. In this paper we evaluate and compare the performance of this scope-based Compass against the self-balanced and self-adaptive routing algorithm named *ColorANT*. Moreover, we compare the Compass performance against two routing algorithms: Flooding and a constrained version of Hotpotato.

1 Introduction

Unger et al. in [BCS⁺07] introduced an adaptive routing algorithm for passive RFID tag information, monitored from any internet-connected tag reader. This algorithm is based on building mesh-like structures on top of a P2P-connected network of the manufacturer's servers. In such kinds of mesh network a node is identified by its X and Y coordinate and references up to eight neighbors (one for each orientation: down, up, right, left, as well as up-right, up-left, down-right and down-left). Due to the nature of the mesh structure (code-named *UTH*¹) each node finds the closest neighbors through a fitness function. This function is based on the Euclidian distance. UTH mesh-based networks offer not only high stability and high speed of reaction by node-jittering in very large networks, but also more efficient and scalable than structured systems.

Several papers deal with online routing and related problems in geometric settings. Bose and Morin [BM99] classify routing algorithms based on their use of memory and/or randomization. In both cases, the message forwarding is based only on the coordinates of the node where the message is located. Bose and Morin study also a randomized and memoryless routing algorithm that works for any triangulation. Kalyanasundaram and Pruhs propose a 16-competitive algorithm to explore any plane graph. If the ratio between the

¹UTH denotes the abbreviation for the "Udon Thani Airport Thailand", where the author invented the building mesh algorithm, while waiting.

length of any path found by algorithm A and the shortest path, is less than K, then A is called K-competitive. Cucka et al. [CNR96] have shown that greedy routing algorithm performs better than Compass routing algorithm ([BM99], [KSU99]) on random graphs, but does not do as well on Delaunay triangulations. Nevertheless, Compass is slightly more efficient in any case if the number of edges traversed is considered. In [BBC⁺00], Bose et al. show that, using Euclidean metric, there is no routing algorithm competitive for all triangulations. They also proved, that no routing algorithm exists that uses the number of traversed edges which are competitive for all Delaunay or greedy triangulations.

1.1 Motivation

In [RC07], we introduced an improved Compass routing algorithm. It delivers messages in the above mentioned mesh-like structure using on each hop the direction closest to the destination node. By using a range of possible directions that point to the destination, the so called *Scope*, Compass avoids paths under congestion conditions.

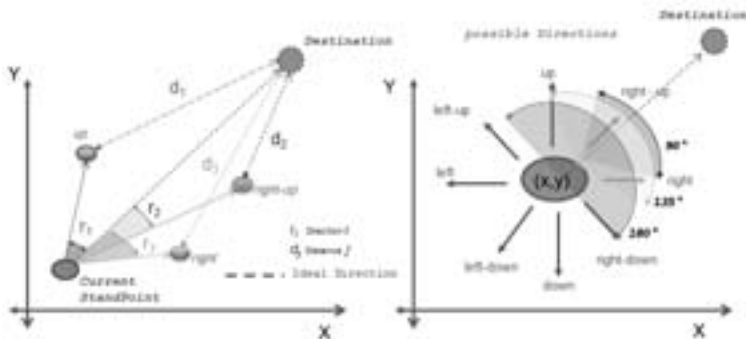


Figure 1: Compass mechanism performed on each peer. Left side shows the Euclidian distance and the angle between the closest neighbors with the ideal direction. Right side depicts the eight neighbors of a peer on the UTH overlay network. Furthermore it shows three different scopes: 90, 135 and 180 grades.

One of the open issues in routing algorithms is the *Hotspot* problem. A peer becomes Hotspot if it forwards messages with a higher frequency than other peers. Hence, a hotspot-peer induces quickly to congestion conditions. In this paper, we evaluate experimentally the performance of this improved Compass and prove its load balancing capability.

In [RUC07], a self adaptive and self-balanced routing algorithm was introduced for message delivery in large unstructured P2P networks. ColorANT adopts Ant Systems as introduced by Dorigo, Maniezzo and Colorini [FBP95]. It uses colored pheromones to look for alternative paths near to the optimal route in case of congestion conditions . As described

in [RC07] and [LU09], the colored routes marked by ColorANT can offer higher availability, coverage and up-to-dateness than the known best local routes approaches. Based on this, this paper aims also to evaluate the high availability and load balancing capability shown by ColorANT in very large networks.

In this paper we compare also the efficiency of two simple routing algorithms: Flooding and HotPotato in UTH networks. The HotPotato algorithm represents a general cacheless routing algorithm, which means the nodes of a network have no buffer to store temporal messages before they reach their destination. In HotPotato, each message that is routed is simply forwarded until it arrives its destination or it reaches its TTL (*time to live*) -then the message dies. Note that this procedure lacks any intelligence. On the other hand, there are many different implementations of the Hotpotato algorithm [FBP95], [TSG08], [HPL01], even buffered ones, when network limitations are considered. For comparison purposes, we modified the implementation of the HotPotato to be used in UTH overlay networks, keeping a list of each node load and its neighbors. It then equally distributes the incoming messages between half of the neighbors located close to the ideal direction of the destination on the mesh. By Flooding algorithm, each node replicates and forwards incoming messages to its neighbors until the message reaches its destination. Nevertheless, the forwarding process is limited by a TTL to avoid network saturation. Due to simplicity, flooding is being used even in many strategies for message delivery in unstructured P2P networks.

2 Performance Evaluation

2.1 Simulation Environment

Simulation Environment P2PNetSim version 2.5 was used for the above mentioned algorithms' performance evaluation. P2PNetSim is a distributable tool for simulating IP based network infrastructures and in particular for simulating large P2P Systems [Col06]. The analysis of simulation results can be classified into two basic groups: small networks (with 256, 512 and 1024 nodes) and large networks (dealing with 2048, 4096 and 8192 nodes). Each simulation scenario was executed 16 times on a cluster with 32 nodes AMD 64 bits Quad-Core 2.6 GHz, 4 GB RAM and 500 TB of storage. The platform used was JDK 1.6 u 18 under Linux Suse 12. The figures in this paper show the average of those 16 experiments according to each scenario.

2.2 Performance Measurements

In this section, we present several important performance measurements whose results will be compared by simulations in Section 3.

Number of steps: determines how many peers were visited starting from the sender until reaching the destination. The evaluation of this factor uses the average value of all delivered messages.

Network usage: This is a measurement for the distribution of used paths on the network. A marginal value means that many potential paths remain unused. This increases of course the probability of congestion over most used routes. The higher the network usage, the higher load balancing by message delivery. The higher value for network usage is 100%.

Amount of overloaded peers: describes how many peers in a given simulation step cannot accept incoming messages anymore, because their queue is full. The evaluation of this factor is calculated during the whole simulation.

Delay: This factor specifies how long a message stays in the queue before it is processed.

Latency: defines the elapsed time from a message is sent until it's delivered. This factor includes the delay experimented on each peer before the message reach his destination.

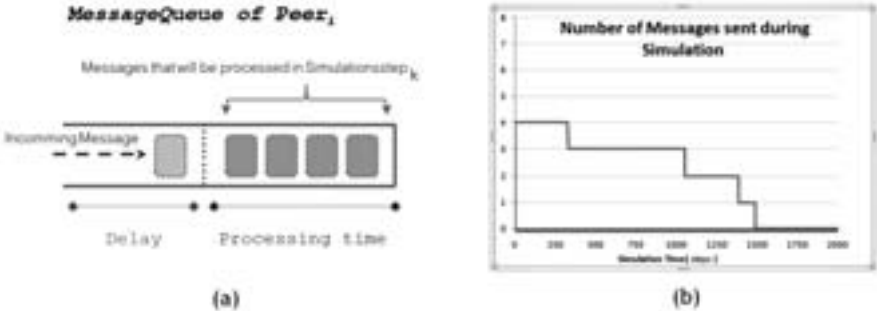


Figure 2: (a) Representation of delay and processing time of message on a peer. (b) Power Law function on number of messages to be routed by each simulation.

3 Simulation Model and Results

As depicted in figure 2 the number of messages generated follows a Power Law function. Each simulation runs 2000 steps and during the last 500 steps, no new messages are generated. This time period is used for processing even the messages that are on the way to their destination. Due to comparison constraints and duration of simulation time, the number of peers generating messages was 4 and 8. The average link length of any pair source-destination peers was $2L$, where L is the square root value of N (network size). The selection of these peers was random. Moreover the TTL for messages was set to $3L$.

3.1 Evaluation of average hops by message delivery and network usage

As can be appreciated in figure 3, Compass and ColorANT clearly need less steps for message delivery than HotPotato and Flooding in small networks. Note that HotPotato requires the highest values (or maximum number) of hops. The reason behind is that HotPotato does not use a smart strategy for message forwarding. In large networks, Flooding shows not only the lower averages of hops but also exhibits the worst percentages of network usage. Additionally, Compass maintains around 100% of network usage. ColorANT presents similar high network usage-, around 80% on networks with 256 peers. This value decreases softly when the networks grow. The network usage in ColorANT reaches approximately 75% with 8192 peers. This is an enhancement against HotPotato from 5% to 75%. In large networks, Compass takes advantage of around 100% of network usage.

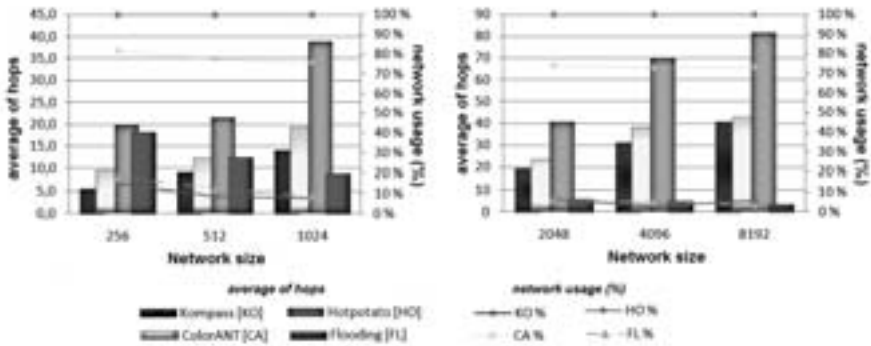


Figure 3: Evaluation of hops needed by message delivery and the network usage. The figure shows the evaluation of Compass in small networks (left side) as well as in large networks (right side)

3.2 Evaluation of the message queue length and number of overloaded peers

Figure 4 shows the simulation results regarding the average length of the message queue of all peers and the percentage of overloaded peers in small and large networks (left and right side respectively). Flooding reveals high values of queue length due to the replication of messages by forwarding. On the other side, HotPotato shows the shortest queues since this mechanism forwards the messages immediately and does not make any copies of the original message. Regarding the number of overloaded peers, ColorANT, HotPotato and Flooding have a similar behavior. This overloading factor lies between 80% and 90%. Compass emphasizes however with values between 30% and 35%. Note additionally that Compass behaves similarly in both small and large networks.

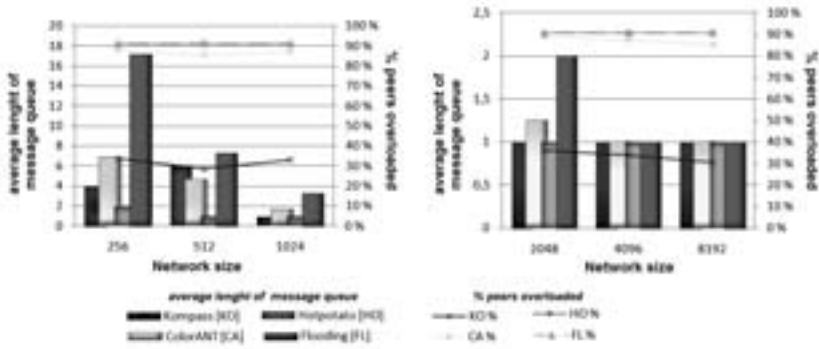


Figure 4: Evaluation of the message queue length and number of overloaded peers

3.3 Evaluation and Comparison of delay and number of hops by message delivering

Figure 5 represents the distribution of average delay and the number of hops in delivering messages of different sizes. In Flooding, a high value is observed for the delay which decreases as the network grows. The number of hops in Flooding is small although it increases lightly with the network size. Note that this value is maintained under 50 hops. This behavior is reflected by Compass and ColorANT as well. Nevertheless, in HotPotato, the number of hops increases strongly. The average delay for all algorithms, except Flooding, remains under 8 simulation steps for messages of 1KB. Compass exhibits a delay close to zero, which is remarkable. A comparable behavior is observed with messages of size of 2KB, 3KB and 4KB. The number of hops is maintained although the delay increases proportionally with the message size, as expected.

4 Conclusions and Future Work

In this paper, we evaluated the performance of two self-adaptive routing algorithms in UTH Overlay Networks. As it was demonstrated, Compass and ColorANT deliver messages with low delay values. In the same way, these algorithms are highly comparable with Flooding as these mechanisms maintain few hops for message delivery. The average delay in most cases remains marginal. Currently, we are evaluating the use of colored ants by the scope-based Compass algorithm. We believe the ants will increase the Compass performance during triangulation situations. In near future, we will evaluate the performance of this routing algorithm cooperation in larger networks that reflect the small-world principle.

Acknowledgements. The authors would like to thank to the University of Dortmund and the FernUniversitaet in Hagen for the facilities to run the simulations on their clusters.

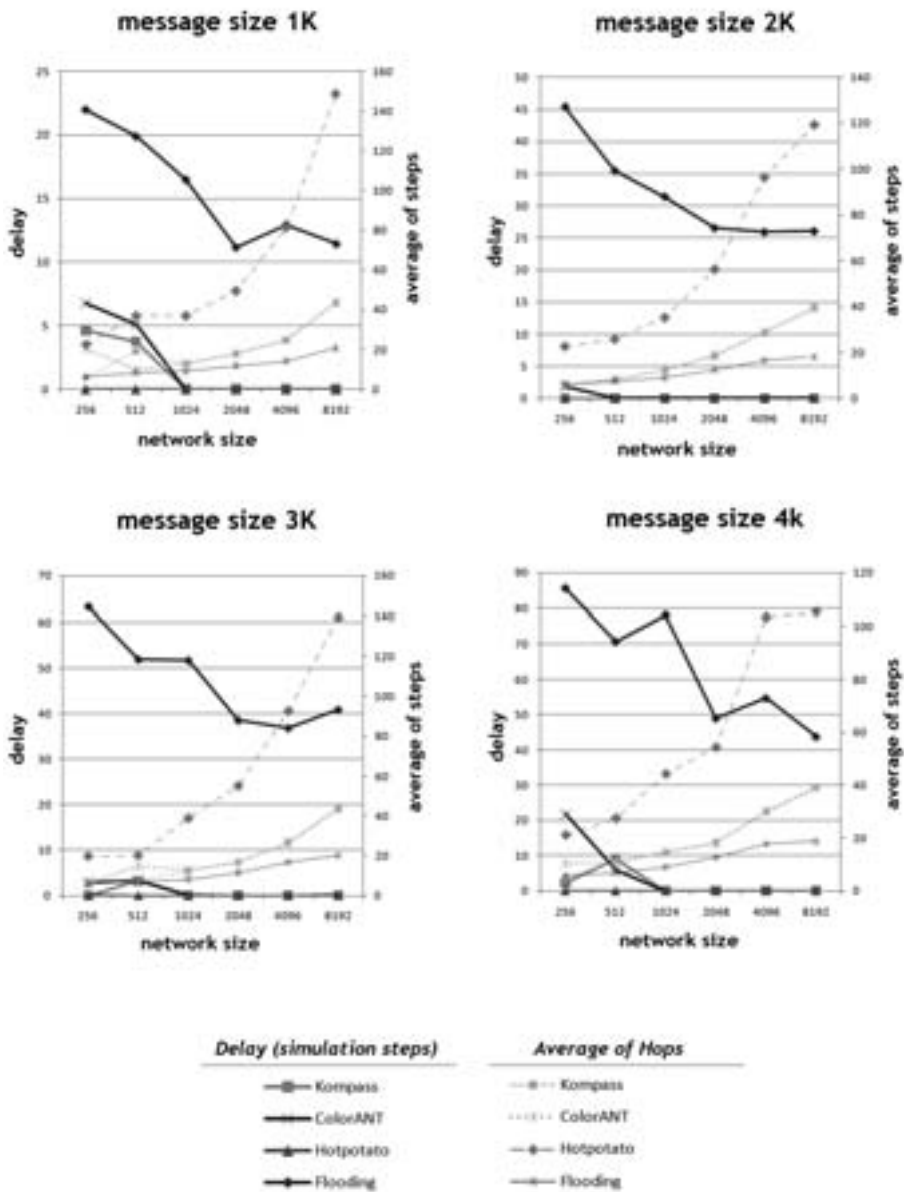


Figure 5: Evaluation and comparison of delay (in simulation steps) and number of hops by message delivery

References

- [BBC⁺00] P. Bose, A. Brodnik, S. Carlsson, E. Demaine, R. Fleischer, A. Lopez, P. Morin, and J. Munro. Online Routing in Convex Subdivisions. In *International Journal of Pattern Recognition and Artificial Intelligence*, pages 429–446, Springer LNCS, 2000.
- [BCS⁺07] D. Berg, H. Coltzau, P. Sukjit, H. Unger, and J. GM. Nicolaysen. Passive RFID tag Processing using a P2P architecture. In *Malaysian Sw. Engineering Conference*, 2007.
- [BM99] P. Bose and P. Morin. Online Routing in Triangulations. In *Tenth International Symposium on Algorithms and Computation ISAAC 99*, pages 113–122, 1999.
- [CNR96] P. Cucka, N.S. Netanyahu, and A. Rosenfeld. Learning in Navigation: Goal finding in graphs. In *International Journal of Pattern Recognition and Artificial Intelligence*, pages 429–446, 1996.
- [Col06] H. Coltzau. Specification And Implementation Of A Parallel P2P Network Simulation Environment. Diploma thesis, Universität Rostock, 2006.
- [FBP95] F. Forghieri, A. Bononi, and Paul R. Prucnal. Analysis and Comparison of Hot-Potato and Single-Buffer Deflection Routing in Very High Bit Rate Optical Mesh Networks. In *IEEE Transactions on Communications*, 1995.
- [HPL01] R. Honkanen, M. Penttonen, and V. Leppänen. Hot-Potato Routing Algorithms for Sparse Optical Torus. In *International Conference on Parallel Processing Workshops (ICPPW'01)*, 2001.
- [KSU99] E. Kranakis, H. Singh, and J. Urrutia. Compass routing on geometric networks. In *11th Canadian Conference on Computational Geometry CCCG 99*, 1999.
- [LU09] L. Lertsuwanakul and H. Unger. An Improved Greedy Routing Algorithm for Grid using Pheromone-Based Landmarks. In *World Academy of Science, Engineering and Technology*, Indonesia, 2009.
- [RC07] M. Rojas and H. Coltzau. 'Compass', A Routing and load balancing algorithm in UTH Overlay Networks for largescale P2P Networks. In *IADIS WWW/Internet 2008 Conference*, Germany, 2007.
- [RUC07] M. Rojas, H. Unger, and H. Coltzau. Self-Balanced and Self-Adaptive Routes in Unstructured P2P Networks. In *Second International Conference on Systems (ICONS'07)*, Martinique, France, 2007.
- [TSG08] R. Teixeira, A. Shaikh, and T. Griffin. Impact of Hot-Potato Routing Changes in IP Networks. In *IEEE Press Piscataway*, NJ, USA Volume 16, Issue 6, 2008.