Multiple Criteria Routing Approaches in Mesh Overlay Networks

Lada-On Lertsuwanakul
Communication Networks
FernUniversiät in Hagen
Hagen, Germany
lada-on.lertsuwanakul@fernuni-hagen.de

Hauke Coltzau
Communication Networks
FernUniversiät in Hagen
Hagen, Germany
hauke.coltzau@fernuni-hagen.de

Herwig Unger
Communication Networks
FernUniversiät in Hagen
Hagen, Germany
herwig.unger@fernuni-hagen.de

Abstract **— The multi-constrained optimal path problem is one of the main issues of Quality-of-Service (QoS) routing, which consists in finding a route between two nodes that meets a series of QoS requirements such as overall delay time, maximum acceptable packet loss ratio, and others. With the aim to improve the QoS routing by considering buffer stages as well as remaining distance to the target, three adaptive routing algorithms in grid-like P2P overlays are presented in this paper: an adaptive probability function, a weighted decision function and a fuzzy-logic approach. In all proposed algorithms, a thermal field is used to communicate the buffer utilization over the network. By means of simulations it is shown that the weighted decision function as well as the fuzzy-logic approach show very good performance according to message losses and overall routing time in both low and high-congestion traffic scenarios. Additionally, all approaches are able to balance the network load and therefore effectively avoid message losses.**

Keywords- multi-constrained decision making, routing algorithm, overlay networks, buffer utilization.

I. INTRODUCTION

The main advantage of structured Peer-to-Peer overlay networks lies in their ability to distribute arbitrary contents over a dynamically changing number of participants and still provide efficient lookup mechanisms. Additionally, such overlays usually provide robust routing architectures, redundant storage and – though more seldom – distributed implementations of trust and authentication mechanisms that avoid single points of attacks and failures.

Unfortunately, in some overlays as e.g. in CAN and Grid-like structures, the routing process can cause single peers to have a high message load, since each may have a central or otherwise crucial position in the network so that a lot of messages are routed to or through it. This problem is enforced, whenever a peer manages content that is accessed by a lot of users in the whole network. The peers around such hot-spots are inherently exposed to higher routing load, since a lot of messages need to be routed to and from the hot-spot. Whereas all messages that are targeted to a hot-spot or its surrounding nodes necessarily have to be routed into the overloaded region, other messages should be routed around it. This not only avoids additional load and possible resulting message losses for the already stressed region, but also decreases and therefore optimizes the delay time for the redirected message. On the other hand, the alternative routes should still have a minimum number of hops to make sure, no messages are lost due to TTL expiries.

To increase the robustness and provide some load-balancing, we therefore propose a routing algorithm for Peer-to-Peer overlays that is able to dynamically route messages around over- or highly loaded peers and regions. To find the fastest, but not necessarily shortest path to the requested target and avoid message losses at the same time, each peer does not only take the target direction into account, but also the buffer levels of its direct neighbors that may be involved into the routing process. To propagate each peer's buffer levels into its neighborhood, a thermal field approach is used.

Such kind of routing problems are generally referenced to as finding a multiple-constraint optimal path (MCOP). The constraints, as e.g. overall routing delay, the maximum number of hops or transfer rate, usually are entailed by application-specific quality-of-service (QoS) requirements.

A multi-criteria decision function is needed to find an appropriate tradeoff between distance and load. Since this function is crucial for the effectiveness of the routing algorithm, we propose and compare three different approaches. Those are: (i) an adaptive probability function, (ii) a weighted decision function as presented in [1], and (iii) a fuzzy-logic approach, which provides a mathematical model for dealing with imprecision and uncertainty as given in common traffic situations in today's communication networks.

The rest of this article is organized as follows: in section II, a short overview about related work is given. Section III discusses both the thermal field approach as well as the three different decision making mechanisms. Section IV shows the simulation of the different decision functions and discusses the results. Section V concludes this article and gives an outlook on future works.

II. Related Work

A. QoS Routing

The multi-constrained optimal path problem (MCOP) is related to the issue of Quality of Service (QoS) routing, which consists in finding a route between two nodes that meets a series of QoS requirements such as overall delay time, maximum acceptable packet loss ratio, and others. Although the utilization of a routing node's message buffers is an indicator for that node's load, current approaches mainly consider available bandwidth or the remaining hop-count for the decision making algorithm [2-4]. Only few approaches, like [5], take buffer utilization into account.

The *Fuzzy Logic Ant based Routing* (FLAR, [23]) is a routing algorithm based on ants, which was enhanced by fuzzy logic. The messages are forwarded according to information gathered from forward and backward ants which dynamically update the routing tables on each node during message transfer. The link delay and link utilization are also considered in the fuzzy logic decision function.

In [24], an Adaptive route selection policy is proposed. The algorithm is based on back-propagation neural networks, which are used to predict the optimum policy for adapting to dynamically changing network load conditions. The back propagation method is used to train the neural network to learn the relationship between different policies and the resulting effects on the network traffic.

B. Routing in Mesh Topologies

Mesh (Grid-like) topologies have been widely used in communication networks as for example in packet/circuit switching between wireless [6, 7] and wired networks [8, 9]. The functions of routing algorithms in general are the provision of the fastest path, prevention of deadlocks, low latency insurance, network utilization balancing, and fault tolerance. Routed by these classical methods, grid-like structures provide multiple paths which have the same hop count. The mesh structure is reliable and offers redundancy which in turn can be used to improve routing performance [10, 11].

In 2000, John Kleinberg [12] introduced a family of small-world network models based on the work of Watts and Strogatz [13]. His models are built of k-dimensional grids with a lateral length of n, in which each peer has undirected local links connecting it to its neighbors. Additionally, directed far distant links are generated randomly. Kleinberg showed, that optimal routing performance can be gained, when a long distance link between two nodes u and v is constructed with a probability proportional to $d(u,v)^{-n}$. Hence, for the two-dimensional case, links are added with a probability proportional to the inverse square of the lattice distance of u and v. In such structures, a path with an expected length of $O(\log^2 n)$ can be found by using a simple greedy algorithm which relies only on local knowledge.

Martel and Nguyen [14] re-analyzed Kleinberg's Small-World model and deduced an expected path length of $\Theta(\log^2 n)$ and a diameter of $\Theta(\log n)$ for the 2-dimensional case. By making use of some additional knowledge of the graph they show that the expected path length can be reduced to $O(\log^{1+1/k} n)$ for a general k-dimensional model ($k \geq 1$)

By taking the neighbors of a node's neighbor into account for decision-making, Naor and Wieder [15] improved the delivery time for greedy algorithms. Finally, Zou et al. [16] claimed that Kleinberg's model needs to use global information to form the structure. Consequently, they proposed to use cached long distance links instead of fixed ones. The structure is refined as more queries are handled by the system.

C. Thermal Field Algorithms

A routing approach in analogy to temperature fields in thermal physics was first introduced by Unger and Wulff [17] in 2004 to locate nodes managing contents of common interest in P2P networks. Each node features a temperature, which is an index for the activity of that node. The heat of each node radiates towards its direct neighbors and therefore influences their temperature as well. Whenever the content of a node is accessed or updated, its temperature is increased, whereas during periods of inactivity, the temperature falls of exponentially to align with the temperatures of the surrounding neighbors.

In 2007, Baumann et al. [18] introduced the *HEAT* routing algorithm for large multi-hop wireless mesh networks to increase routing performance. *HEAT* uses anycasts instead of unicasts to make better use of the underlying wireless network, which uses anycasts by design.

HEAT relies on a temperature field to route data packets towards the Internet gateways. Every node is assigned a temperature value, and packets are routed along increasing temperature values until they reach any of the Internet gateways, which are modeled as heat sources. It is a distributed protocol to establish such temperature fields which does not require flooding of control messages. Rather, every node in the network determines its temperature considering only the temperature of its direct neighbors, which renders our protocol particularly scalable to the network size.

III. Multi-Criteria Routing Algorithm

We present three algorithms for making routing decisions in grid-like structures, where each routing node only has local knowledge. Additionally to the Euclidean distance from the current node to a message's destination, the approach also takes the current buffer stages of a routing node's neighbors into account to find optimal paths around congested areas or nodes.

A thermal field is used to communicate the buffer utilization over the network, rendering every node to memorize its neighbors' temperatures. A lower temperature indicates that the respective neighbor currently has more communication resources available and will therefore be capable of handling new data. On the other hand, a message should still be directed towards its destination. Therefore, in the route selection process, the distance between the origin and target node, the length from the current peer to the target, and the distance from each neighbor to the target node are measured.

We evaluate the performance of our fuzzy-logic based decision function against both an adaptive probability function as well as a weighted decision function. All three approaches base the routing decision on a combination of neighborhood temperatures and target distances. Before we provide a detailed description of each of the approaches, we show how the temperature values of each node are calculated and distributed to build the thermal field.

In the presentation of the algorithms, we denote a grid-like network by a set of lattice points in $n \times m$, $\{(i,j): i \in \{0,1, \ldots\ n-1\}, j \in \{0,1, \ldots\ m-1\}\}$. A node's ID is defined by its coordinate (i,j).

A. Thermal Field

In the discussed algorithms, the temperature θ indicates the usage level of a peer's incoming- and outgoing message buffer. The temperature of a node c is referred as θ_c. The possible values of θ_c are in the range from 0 to 1, where 0 denotes an empty buffer and a value of 1 indicates that the buffer is full.

$$\theta_c = \frac{\text{Messages in Buffer}}{\text{Buffer size}}, \quad 0 \leq \theta_c \leq 1 \qquad (1)$$

To reduce complexity, each node only uses one message buffer, which is organized in a FIFO manner. Hence, the temperature of that buffer is equal to the temperature of the node.

Since the routing decision strongly depends on θ_c being up to date, the temperature is recalculated with every message that enters or leaves a buffer. Additionally, the messages themselves act as temperature-carriers, conveying a node's temperature from one peer to another until they either reach their target or expire. This underlines the analogy to convectional processes in thermal physics, where temperature is conveyed by rapidly moving particles.

Each node keeps account on the temperature of its neighbors. Let $N(c)$ be the set of neighbors of c and let k is the number of neighbors when $1 \leq k \leq 4$ in degree of distribution mesh structure is 4. Let i be the index of each neighbor N_i in $N(c)$ where $1 \leq i \leq k$. Additionally, let φ_i be the number of messages sent from N_i to c. Now, there are two cases to update a neighbor's temperature $\theta(N_i)$ in c's dataset:

(i) Whenever c receives a message from neighbor N_i, containing that node's temperature θ_i, the previously stored value, is overwritten:

$$\theta(N_i) = \theta_i, \quad \text{if } \varphi_i > 0 \qquad (2)$$

(ii) If no message is sent from N_i to c, θ (N_i) is decreased exponentially over time with a configurable time constant of λ:

$$\theta(N_i) = \theta(N_i) \cdot e^{-\lambda t}, \quad \text{if } \varphi_i = 0 \qquad (3)$$

The thermal field was used for all approaches analyzed in this article to enable decision making with only local knowledge.

B. Adaptive Probability Function

The basic concept of using an adaptive probability function is to base the decision on which path to select on a configurable parameter P_θ, which denotes the probability for selecting low-temperature routes in preference over the shortest path. Each node on the route randomly selects a low-buffer route with a probability of P_θ, or a direct route with the probability of $1 - P_\theta$. Higher values P_θ make each node prefer low buffer routes, which may lead to longer routing times. On the other hand, smaller values for P_θ let peers select a direct route more often, and hence may increase the number of message losses due to overloaded nodes along the shortest path. Thus, the challenge is to find values for P_θ which result in both optimal routes and a load balanced network.

It is clear that suitable values for P_θ depend on the current distance that a message still has to bridge to reach its target. If the message is still close to its source, making a detour is acceptable, whereas if only few hops are left to the destination, direct paths should be preferred. Therefore, we propose to use adaptive probability functions (AP_θ) that provide values for P_θ depending on the relative remaining distance, which we denote as Ω. When a message is sent from a source node σ to a destination node ϕ, the distance between the two nodes is $d(\sigma,\phi)$. The distance from any node c along the path to the target is $d(c,\phi)$.The relative remaining distance Ω is now determined as follows:

$$\Omega = \frac{\text{Distance\_current\_to\_target}}{\text{Distance\_source\_to\_target}} = \frac{d(c,\phi)}{d(\sigma,\phi)} \qquad (4)$$

In previous works [18] it came out that two adaptive probability functions of Ω showed good results for different scenarios, which we denoted as AP_θ^4 and AP_θ^5.

Adaptive Probability4 (AP_θ^4): AP_θ^4 is an exponential cumulative distribution function (cdf). The probability of using a low-temperature route has a co-domain of [0, 1). It results in strongly preferring low-temperature paths at the beginning of the routing process. The closer the message comes to the target, the more the direct route is preferred. When the target is only a few hops away, the thermal field is almost completely ignored.

$$AP_\theta^4(\Omega; \lambda) = 1 - e^{-\lambda\Omega} \qquad (5)$$

Adaptive Probability5 (AP_θ^5): Whenever the low-temperature path is preferred over the shortest path, the message could go astray, which results in higher probability of message losses. Therefore, AP_θ^5 is designed to pull back the message onto the shortest path, whenever the current distance to target becomes larger than the overall distance between source and target. In such cases, the probability of using the path with the lowest temperature decreases.

$$AP_\theta^{5}(\Omega;\lambda) = \begin{cases} e^{-\lambda\left(1+\frac{1}{\Omega(t)}\right)}, & d(c2\phi) \leq d(\sigma2\phi) \\ 1 - e^{\left(\frac{-\lambda}{\Omega(t)}\right)}, & d(c2\phi) > d(\sigma2\phi) \end{cases} \quad (6)$$

Fig. 2 depicts both AP_θ^4 and AP_θ^5 as functions of the relative remaining distance Ω.

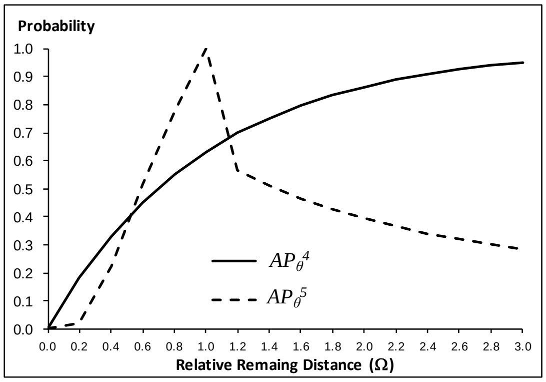


Figure 2. Adaptive Probability Functions AP_θ^4 and AP_θ^5

C. Weighted Decision Funtion

In this approach, a weight is assigned to every neighbor that is based on the neighbor's distance to target and temperature. The message is then routed to the neighbor with the lowest weight. Each weight is calculated as a linear combination, where the coefficients control the influence of each summand on the total weight. Although this approach seems similar to the adaptive probability function presented in the previous section, it is different in one important point: it always selects the path with the minimum weight, whereas the AP approach in some situations might select paths with higher temperature to explore them.

Again, let all N_i with $1 \leq i \leq 4$ be the neighbors of current node c and let ϕ be the target node of the message that is to be routed. Additionally, let $d(N_i,\phi)$ be the Euclidean distance from a neighbor to the target node and finally, let $\theta(N_i)$ be the temperature of neighbor N_i. The weight of the edge to the neighbor N_i is now calculated as follows:

$$f(N_i, \phi) = \alpha \cdot d(N_i, \phi) + (1-\alpha) \cdot \theta(N_i) \quad \text{with } 0 \leq \alpha \leq 1 \quad (8)$$

The coefficient α defines the influence of the remaining distance to target and the load of the next hop on the total weight and therefore on the routing decision. Higher values for α let the node select a more direct path while taking the risk to lose the message due to buffer overflows. On the other hand, lower values for α result in selecting a low buffer route, which on the other hand may leading to long routing times.

The pseudo-code of the weighted decision function approach is shown in Fig. 3. This code is executed on every node at each simultion timestep.

```
1   ..
2   while (receiveMsg != null)  {
3     updateNeighborTemperature();
4     if (currentIsTarget())
5       continue;
6     else if (queueBuffer !=MAX)
7       keepInQueue();
8     else
9     lostMessage();
10  }
11  while (queueBuffer != null && outBW != MAX){
12    popMessageFromQueue();
13    for (1 to sizeOfNeighbor)  {
14      d = α * (distanceToTarget);
15      t = (1-α) * (Temperature);
16      WeightNeighbor = d + t;
17    }
18    nextNode = minWeightNeighbor();
19    forwardMsg( nextNode );
20  }
21  spreadTemperature();
22  ..
```

Figure 3. Pseudo-code of weighted decision function approach

The algorithm consists of three parts. In lines 2 to 10, the node receives messages, brings its neighborhood temperature database up to date and decides, if the message needs to be routed or already received its target. If the message is to be routed further, the buffer is checked for remaining free space to handle the message. If no free space is available, the message is dropped.

Lines 11 to 20 describe the forwarding process that is started, when the buffer contains any messages. In line 12, the message is taken from the FIFO buffer, in line 13-17, the weights for each neighbor are calculated. Then, the minimum weight is selected and the message is forwarded to the according neighbor (lines 18-19). The minimum value is found on line 18. Afterwards, line 19 is used to forward the message. Finally, the node recalculates its own buffer's temperature.

D. Fuzzy Logic Approach

Fuzzy Logic was first introduced by Zadeh [21] in 1965. It allows a computer to take decisions the same way as humans do it: not always precise. People think and reason using linguistic terms such as "hot" and "fast", rather than using precise numerical terms as "90 degrees" or "200 km/hours", respectively. The fuzzy set theory models the interpretation of imprecise and incomplete sensory information as perceived by the human brain. Thus, it represents and numerically manipulates such linguistic information in a natural way via membership functions and fuzzy rules. Some advantages of fuzzy logic are that it is conceptually easy to understand, flexible, and tolerant towards imprecise data. It can model nonlinear functions of high complexity, and it also can be built on top of expert's experience.

A key feature of Fuzzy Logic is to handle uncertainties and non-linearity as they exist in physical systems, similar to reasoning conducted by human beings, which makes it very attractive for decision making systems. A fuzzy logic system comprises basically three elements: (i) Fuzzification, (ii) Knowledge base (rule and function), and (iii) Defuzzification. Fig. 4 shows the generalized block diagram of a fuzzy system.

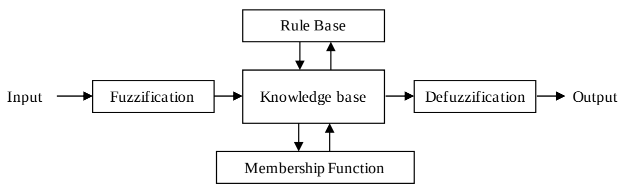


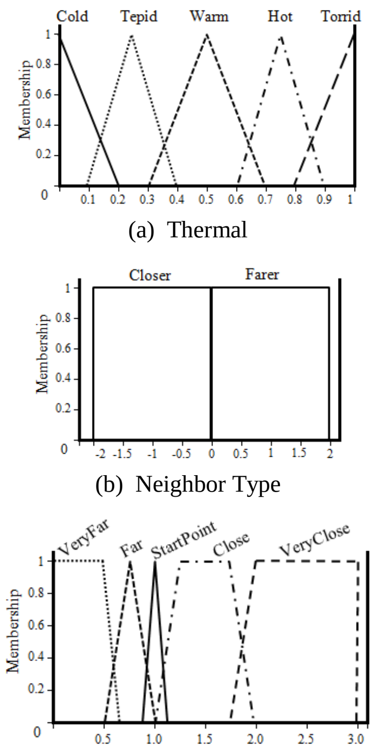
Figure 4. Block diagram of a generalized fuzzy system

The function of fuzzification is to determine an input value's degree of membership to the best corresponding value of a fuzzy set. The fuzzy rule base is used to present the fuzzy relationship between input- and output fuzzy variables. The output of the fuzzy rule base is determined based on the degree of membership specified by the fuzzifier. The defuzzification is used to convert outputs of the rules into so-called "crisp" values, like real numbers.

In terms of temperature field routing, the inputs to the fuzzy controller are: (i) buffer usage status, (ii) current distance to target, and (iii) neighbor type. These three selection parameters make the route reflect the network status, the nodes' ability to reliably deliver packets as well as the direction to the target. The buffer usage is calculated the same way as in (1), the current distance to target is the reciprocal value of Ω as calculated (4). The neighbor type is defined by the difference of the distances from current node to target and from the respective neighbor to target, $d(N_i,\phi)-d(c,\phi)$.

Those three input variables are now fuzzified. The neighbor's temperature is now described as either "*Cold*", "*Tepid*", "*Warm*", *Hot*" or "*Torrid*", the neighbor type can either be "*Closer*" or "*Farer*".

Finally, the distance can either be "*VeryFar*", "*Far*", "*StartPoint*", "*Close*" or "*VeryClose*". Fig. 5 shows the respective membership functions to classify the input variables. Five terms are defined to describe the output of the evaluation of each neighbor: Using a neighbor as the next hop can either be rated "*VeryBad*", "*Bad*", "*Fair*", "*Good*", or "*VeryGood*" as shown in Fig. 6.



(a) Thermal
(b) Neighbor Type
(c) Distance

Figure 5. Fuzzy Membership function of input variable

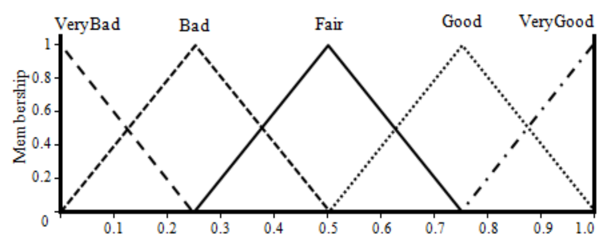


Figure 6. Fuzzy Membership function of Neighbor Rating

A rule set containing 50 atomic rules as shown in table 1 is now used to evaluate the suitability of each neighbor for routing the message next. The rules assign a single rating value to every possible combination of input values. By merging of rules, the number of rules can be reduced to a total of 37.

Table1. Fuzzy Rule Base

Neighbor Rate		**Neighbor = Closer**					**Neighbor = Farer**				
		Thermal					*Thermal*				
		Cold	*Tepid*	*Warm*	*Hot*	*Torrid*	*Cold*	*Tepid*	*Warm*	*Hot*	*Torrid*
Distance	VeryClose	VeryGood	VeryGood	Good	Fair	Bad	Good	Good	Fair	Bad	VeryBad
	Close	VeryGood	VeryGood	Good	Fair	Bad	Good	Good	Fair	Bad	VeryBad
	StartPoint	VeryGood	VeryGood	Good	Fair	Bad	Good	Good	Fair	Bad	VeryBad
	Far	VeryGood	Good	Fair	Bad	Bad	Fair	Fair	Bad	VeryBad	VeryBad
	VeryFar	VeryGood	Good	Fair	Bad	Bad	Fair	Bad	VeryBad	VeryBad	VeryBad

Some example rules are:

R1: IF thermal IS Cold AND neighbor IS Closer THEN neighbor_rate IS VeryGood;
R2: IF thermal IS Torrid AND neighbor IS Farer THEN neighbor_rate IS VeryBad;
…
R37: IF thermal IS Hot AND distance IS VeryFar AND neighbor IS Farer THEN neighbor_rate IS VeryBad;

Rules *R1* and *R2* are combined rules that only depend on two input values, because the third value (*distance* in this case) has no influence on the result.

The rating can now give guidance for which neighbor to use as the next hop. If more precision is needed, because several neighbors have the same rating, Defuzzification, i.e. the process of conversion of a fuzzy output set into a single number, can provide more clarity. In our simulations, Mamdani's "Center of Gravity" (COG) method has been used:

$$Neighbor\ Rate = \frac{\sum_{i=1}^{N} x_i \cdot \mu(x_i)}{\sum_{i=1}^{N} \mu(x_i)} \quad (9)$$

So, the center of gravity is calculated by multiplying each input value (x_i) with the output of its corresponding membership function ($\mu(x_i)$), sum up all of those products and divide it by the sum of the membership function's outputs. The COG method is the most widely used defuzzification strategy, which is reminiscent of the calculation of the expected value of probability distributions.

IV. Simulations

In this section, the simulation results for the three different decision making approaches based on thermal fields for buffer load propagation are discussed.

A. Simulation Setup

1) *Simulation Tools* – The simulation was analyzed using P2PNetSim, a simulation environment for large distributed P2P networks [22]. This flexible tool can be used to simulate, model, and analyze any kind of networks. It has been used for example to analyze distributed RFID-processing as well as the spreading of infectious diseases. Due to the distributed nature of the simulation engine, it is able to handle simulations with millions of individuals. Peers are configured collectively but still individually using an open XML configuration format. for simulation setup. The peer-behavior can be implemented in the Java programming language.

2) *Network* – In the simulations, the networks are organized into two-dimensional grid structures, each composed of 10,000 nodes (100x100). Nodes are connected to their neighbors in all four directions. The coordinate of a node is serves as its ID. The grids overlay a simulated IPv4 network. The buffer sizes and outgoing bandwidths are limited for all the peers, both distributions following a power-law distribution. There are two types of messages: data packet and acknowledgements. The system handles data packet in First-In-First-Out (FIFO) manner, while the acknowledgements are handled with priority.

3) *Traffic pattern* – Traffic is generated randomly by all network nodes. The sending probabilities and intensities are distributed exponentially for both a source node generates, as well as the number of messages that can be sent per simulation time-step. The constants λ_{send} and λ_{number} therefore indicate the load (congestion) of the simulated network. All simulations run until 500,000 messages have been processed,

4) *Performance measurement* – The metrics used to measure the performance using different decision methods are *loss and success ratios*, *average hop-count*, *average delay time (time-steps)*, and *average routing time*. The total routing time includes both the routing steps and waiting times (delay) on busy nodes. Furthermore, load balancing performance was assessed by the number of heated nodes with a buffer usage ration of more than 0.7. For the three decision mechanisms described in section 3, Adaptive Probability Function (AP), Weighted Decision Function (WF), and Fuzzy Logic (FL), the performance is measured. For the weighted decision function, three configurations for *α:1-α* have been used: 0.1:0.9, 0.5:0.5, and 0.1:0.9. So, in the 0.1:0.9-configuration, the decision function considers the distance to target with a weight of 0.1 and the temperature with 0.9, whereas in the 0.9:0.1-configuration, the distance to target is weighted with 0.9 and the temperature with 0.1.

All three approaches are compared to a pure shortest path approach (SP), which does not take the current buffer level of the next hop on the route into account.

B. Simulation Results

The first scenario compares the performance of the decision mechanisms in a low congestion networks. The time constant that defines the probability to generate messages on a specific node is $\lambda_{send} = -0.1$ and each source node generates only one message per time at maximum. The average number of messages generated per simulation time step is approximately 800 messages.

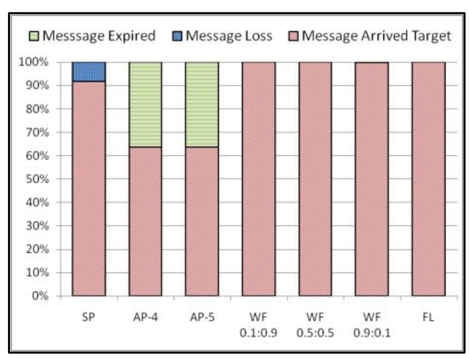


Figure 7. Success delivery ratios in low congestion networks.

Fig. 7 shows the success delivery ratio comparing all decision algorithms in such low congestion networks. The weighted functions as well as the fuzzy logic approach show the best results with 100% of successfully delivered messages, so there was no message expired or lost due to overloaded nodes. On the other hand, the adaptive probability functions show a message expiry ratio of 36%. Those losses occur, when the decision functions strongly prefer low-temperature routes over the shortest path. This way, messages can take remarkably longer routes, inevitably leading to a higher message expiry ratio.

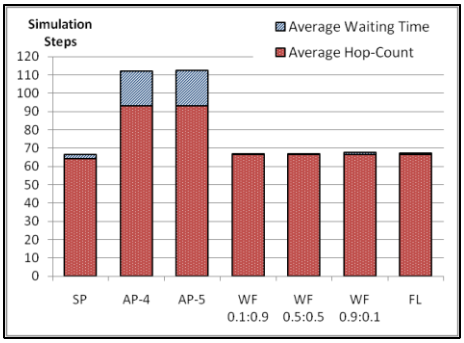


Figure 8. Average routing time is the summary of hop-count and waiting time in low congestion networks.

The average routing time for low congested networks is shown in Fig. 8. Shortest path, weighted decision function and fuzzy logic approach all have similar performance results. In terms of hop-counts and delivery time, the shortest path approach generates the best results with an average of 60.87 hops per message. During 3.83 time-steps, the average message is delayed in congested message buffers, so that the SP approach sums up to a total routing time of 64.70 time-steps. The performance of the weighted decision functions in the configurations 0.1:0.9, 0.5:0.5, and 0.9:0.1, as well as the performance of the fuzzy logic approach are in the same range with only slightly increased values. The hop-counts are 65.12, 63.28, 63.51, and 64.98, whereas the delay times are 3.58, 3.45, 3.81, and 4.08 respectively. So, the average routing times are 68.71, 66.73, 67.32, and 69.06 in order. But the adaptive probability function results show remarkably higher number of hops and delay time.

In Fig. 9, the load balance of networks is presented. The graphs represent number of nodes that have temperature or buffer utilization level higher than 0.7 or 70% of the buffer space. The shortest path method obviously shows many high temperature nodes comparing to others decision algorithms.

The results of the first simulation scenario are that in low congestion networks, the shortest path approach delivers messages in the fastest possible manner and therefore shows the best performance. On the other hand, SP generates the highest amount of heated nodes, even though total load of the network is yet low. The weighted decision function and the fuzzy-logic approach accept short detours, resulting in slightly higher routing times, but utilize the network resources much better and therefore generate remarkably fewer heated nodes. The adaptive probability function approaches show considerably longer routing times.

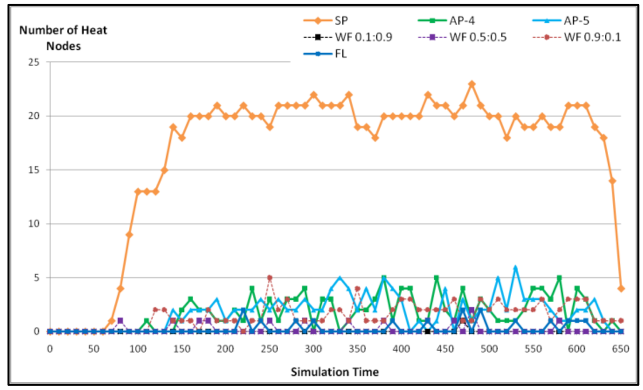


Figure 9. Number of nodes had higher buffer usage level than 70% of buffer size in low congestion networks.

In the second scenario, medium congestion networks are analyzed. The time constants for the probability distribution functions are $\lambda_{send} = 0.1$ for the number of messages per peer and $\lambda_{number} = 0.5$ for the number messages per time-step. The average number of messages that are launched per simulation time is approximately 1,000 messages.

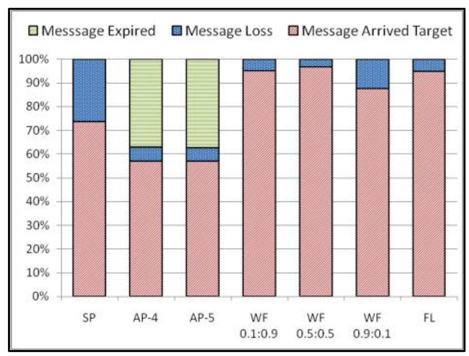


Figure 10. Success delivery ratios in medium congestion networks.

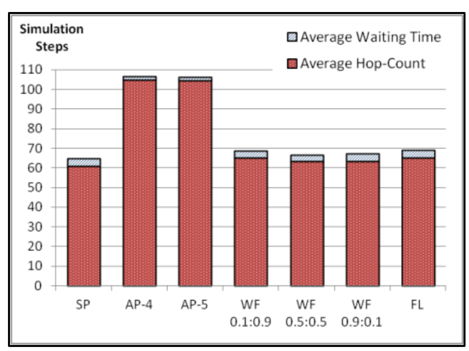


Figure 11. Average routing times in medium congestion networks.

Similar results can be seen when analyzing the routing times as shown in Fig. 11. Again, the 0.5:0.5 configuration for the weighted decision function shows best performance with 66.73 time-steps of average total routing time, which consists of an average number of 63.28 hops and 3.45 time-steps of delay. Again, the 0.1:0.9 configuration as well as the fuzzy-logic approach show similar performance. In contrast to the message loss ratio in Fig. 10, the 0.9:0.1 also shows good results, when it does not loose messages due to overfull buffers. The adaptive probability function again needs a lot more hops to route the message.

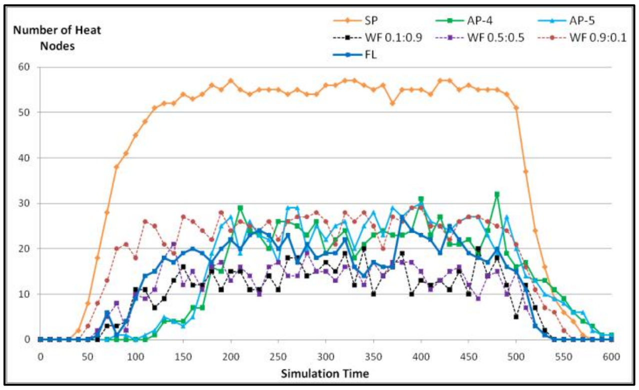


Figure 12. Number of nodes had higher buffer usage level than 70% of buffer size in medium congestion networks.

As expected, the shortest path approach tends to produce overfull buffers a lot more than all multi-constraint approaches, as can be seen in Fig. 12.

In the final scenario, the traffic in a highly loaded network is analyzed. The time constants for the distribution functions now are $\lambda_{\text{send}} = 0.1$ and $\lambda_{\text{number}} = 0.1$. The average number of messages per time-step is approximately 3,200.

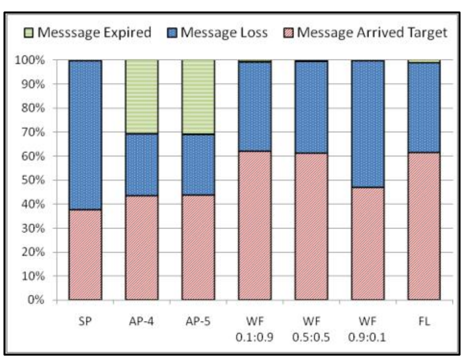


Figure 13. Success delivery ratios in overloaded traffic networks.

In this scenario, the shortest path approach can only deliver a little more than one third of the messages, because now a lot of highly loaded nodes exist and the shortest path approach puts even more load on these nodes. Both adaptive probability functions now perform a little better than SP with 43% of successfully delivered messages (Fig. 13). It is remarkable that the AP approach does loose the most of the messages due to TTL expiries and not because of overloaded buffers. Again this is a direct result of this approaches tendency to make detours into network regions that are far away from the shortest path. So, in highly loaded networks, increasing the TTL could make the AP approach very successful. In terms of successful delivery, the 0.1:0.9 and 0.5:0.5 configurations of the weighted decision function and the fuzzy-logic approach show the best performance with approximately 60%.

If the shortest path algorithm is able to route a message to target, it does so in the fastest possible manner. The weighted decision functions and the fuzzy-logic approach take slightly longer routes, which directly results from avoiding highly loaded nodes. The adaptive probability functions need a lot more hops but on the other hand show the best values for message delays in congested buffers (Fig. 14).

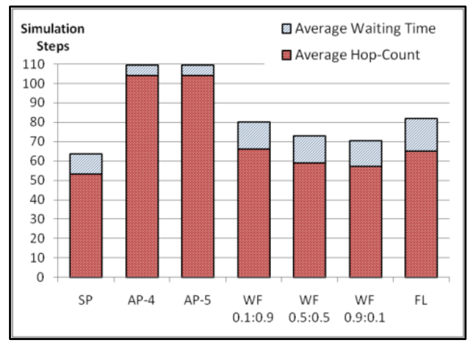


Figure 14. Average routing time is the summary of hop-count and waiting time in overloaded traffic networks.

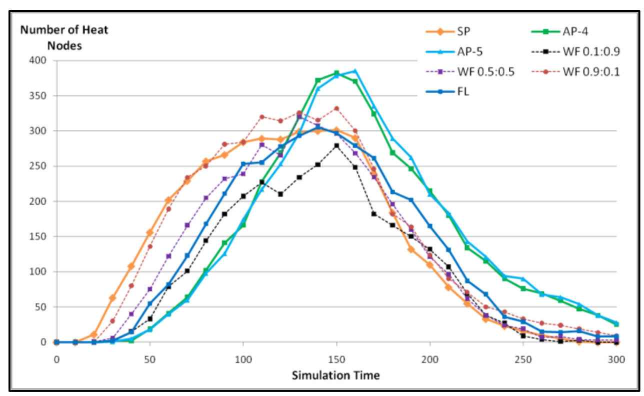


Figure 15. Number of nodes with higher buffer usage level than 70% of buffer size in overloaded traffic networks.

In highly congested networks, the fuzzy-logic approach distributes the load best over time and therefore does produce the lowest amount of heated nodes per time-step (Fig. 15). It is remarkable that the results of shortest path approach are similar to those of the weighted decision functions and even better than the adaptive probability functions. This is, because a lot of messages are dropped long before they are delivered or expire and therefore do no longer add on the network load.

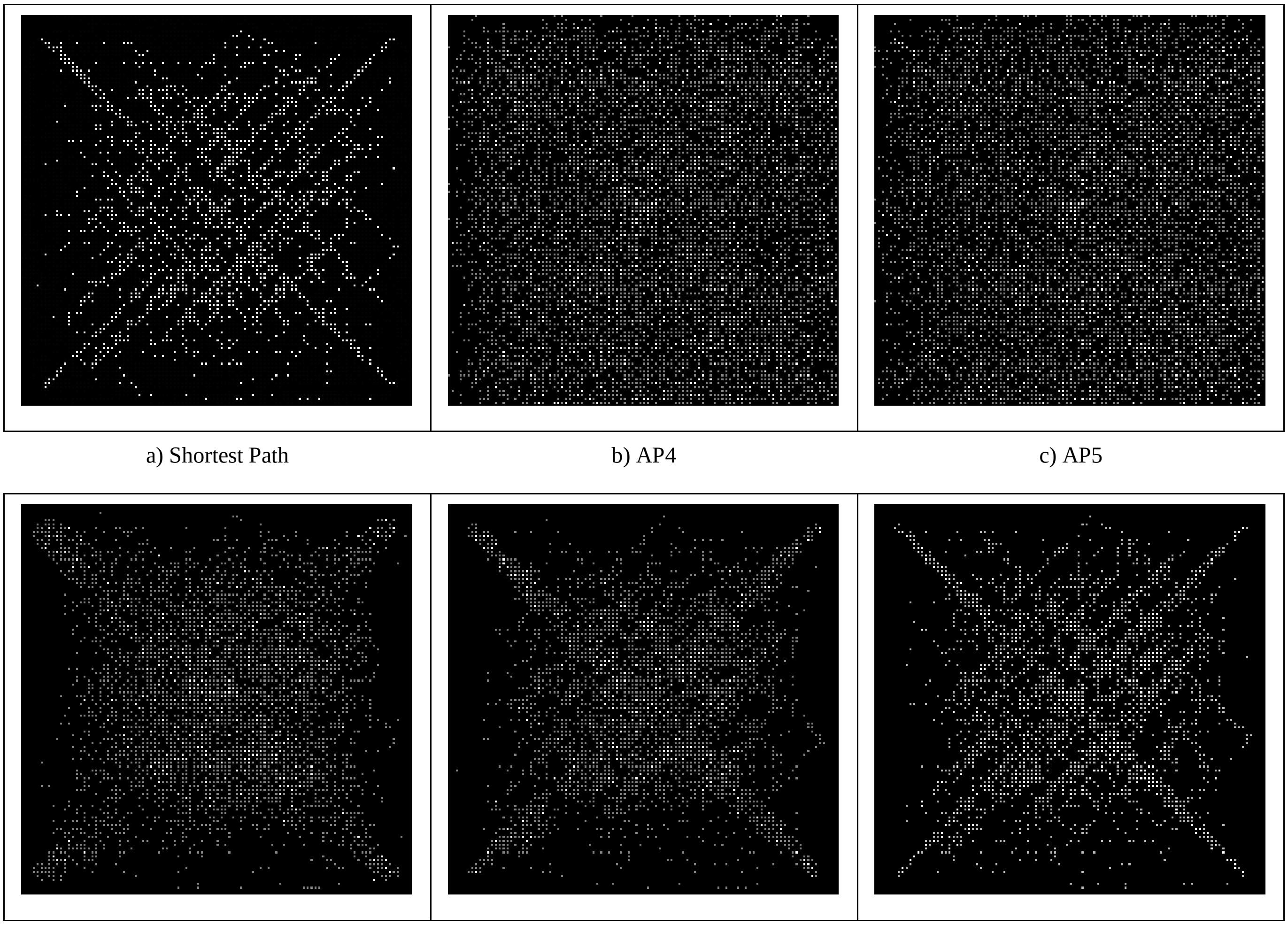


Figure16. Heat nodes distribution in highly congested network

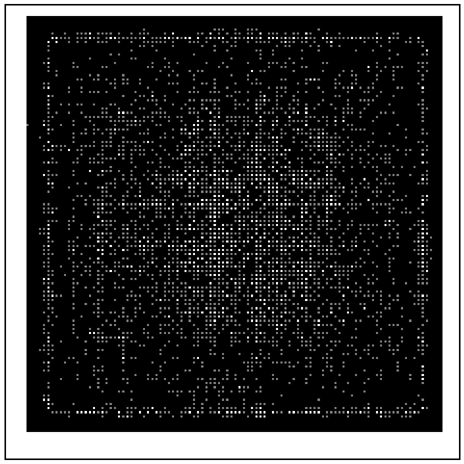
Another part of the simulation results is the distribution of highly loaded nodes over the whole network, which is shown in Fig. 16. The snapshots refer to the high-congestion scenario. Each subfigure represents the 100x100 grid at the simulation time of highest load. Each pixel represents one node. The lighter the node is, the more load it has, i. e. the higher its buffer utilization and therefore its temperature is. To make the source and target nodes clearly visible, in this simulation only four nodes placed at the corners of the grid generate messages to the nodes on the opposite side of the network.

It can be seen that the shortest path algorithm (Fig. 16a) generates a lot of overloaded nodes along the direct path from source to target, but only few load aside from that paths. On the contrary, the adaptive probability functions (Fig. 16 b,c) distribute the load over the whole network. One can see that this approach does not generate fewer loads than SP. This is, because the messages stay a lot longer in the network, until they are dropped due to TTL.

In Fig. 16 d to f, the three configurations for the weighted decision function are shown. In general, the results for this approach show that it uses a balanced path between shortest distance and lowest temperature. So, the 0.1:0.9 configuration allows for some detours from the direct path and therefore generates only few highly congested nodes along the direct path and more mildly congested nodes left and right from that path. In the 0.5:0.5 configuration, the shortest paths are taken stronger into account, so that more highly loaded nodes can be seen on that path. Finally, the 0.9:0.1 configuration looks similar to the shortest path approach although it still balances the load better than SP.

Finally, the fuzzy logic algorithm distributes the message paths over almost all possible routes and so takes most of the load from the highly stressed center of the network in Fig. 16g. In terms of network balancing, this approach therefore generates the best results.

Summarizing the simulation results, one can say that both the weighted decision function and the fuzzy-logic approach are able to handle high-traffic situations remarkably better and with an almost twice the success ratio as the shortest path approach does. While the weighted decision approaches show slightly better performance in terms of overall routing



g) Fuzzy Logic

Figiure16. Heat nodes distribution in highly congested network

time, the fuzzy-logic approach balances the traffic much better over the whole network and therefore avoids creating overloaded regions. Although the adaptive probability functions also distribute the load over the whole network, they show poor performance in low-congestion situations.

Additionally, fuzzy-logic has the advantage that it can take more constraints into account, as e.g. bandwidth, size, load prediction, etc., which makes this approach more flexible than all other approaches analyzed in this article.

V. Conclusion and Future Work

We have presented and analyzed three general approaches for multi-criteria optimum path decision making in distributed systems. All approaches base their decision on both the distance to target as well as the current load of the possible next hop nodes. The load was distributed using a thermal field approach. Through simulations, we have shown that the weighted decision function and the fuzzy-logic approach show good performance in different network traffic scenarios. The flexible fuzzy-logic approach also is able to balance the network load over the whole network in a very good manner.

Because both the weighted decision function and the fuzzy logic approach showed good performance in high-congestion networks, both concepts shall be merged as part of our future works. The result shall be a dynamically generated weighted decision function that can take more than just two parameters into account and so be truly multi-criteria. Additionally, the thermal filed approach can also be used to build a traffic dependent overlay network structures that can enable and disable links depending on the load of nodes or regions.

Finally the outlook for the project is to implement and deploy the proposed algorithms using real tested data, as well as compare to existing similar approaches.

References

[1] L. Lertsuwanakul, S. Tuammee, and H. Unger, "Routing with Temperature Field in Mesh Overlay Network", In Proc. of 9th International Conference on Networks (ICN 2010), France, 2010, pp. 285-290.

[2] B. Peng, A. H. Kemp, and S. Boussakta, "QoS Routing with Bandwidth and Hop-Count Consideration: A Performance Perspective, Journal of Communications, Vol. 1, No. 2, May 2006.

[3] M. Song and S. Sahni, "Approximation Algorithms for Multiconstrained Quality-of-Service Routing," IEEE Transactions on Computers, vol. 55, no. 5, May 2006, pp. 603-617.

[4] X. Lin and N. B. Shroff, "An Optimization Based Approach for QoS Routing in High-Bandwidth Networks," In Proc. of IEEE INFOCOM, 2004.

[5] R. Zhang and X. Zhu, "Fuzzy Routing in QoS Networks", FSKD 2005, LNAI 3614, pp. 880-890.

[6] A. Lee and P. A. S. Ward, "A Study of Routing Algorithms in Wireless Mesh Networks", ATNAC, December 2004.

[7] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," In Proc. Of 10th annual international conference on Mobile computing and networking (MobiCom'04), USA, 2004.

[8] A. Vishwanath and W. Liang, "On-Line Routing in WDM-TDM Switched Optical Mesh Networks,", 6th International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05), 2005, pp. 215-219.

[9] D. Seo, A. Ali, W. T. Lim, N. Rafique, and M. Thottethodi, "Near-Optimal Worst-Case Throughput Routing for Two-Dimensional Mesh Networks,", 32nd Annual International Symposium on Computer Architecture (ISCA'05), 2005, pp. 432-443.

[10] A. Lee and P. A. S. Ward, A Study of Routing Algorithms in Wireless Mesh Networks, ATNAC, December 2004.

[11] A. V. Mello, L. C. Ost, F. G. Moraes, and N. L. Calazans, Evaluation of Routing Algorithms on Mesh Based NoCs, Technical Report Series No.040, May 2004.

[12] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective", In Proc. of 32nd ACM Symposium on Theory of Computing, 2000.

[13] D. Watt and S. Strogatz, Collective dynamics of small world network, Nature (393), 1998, pp. 440-442.

[14] C. Martel and V. Nguyen, Analyzing Kleinberg's (and other) Small-world models. In Proc. 23rd ACM Symposium on Principles of distributed computing, Canada, 2004, pp. 179-188.

[15] M. Naor and U. Wieder, Know the Neighbor's Neighbor: Better Routing for Skip-Graphs and Small Worlds. LNCS, Vol. 3279, 2005, pp. 269-277.

[16] F. Zou, Y. Li, L. Zhang, F. Ma, and M. Li, A Novel Approach for Constructing Small-world network in Structure P2P Systems. LNCS, Vol. 3251, 2004, pp. 807-810.

[17] H. Unger and M. Wulff, Search in Communities: An Approach Derived from the Physic Analogue of Thermal Fields, Proc. the ISSADS 2004, LNCS 3061, Mexico, 2004.

[18] R. Baumann, S. Heimlicher, V. Lenders, and M. May, HEAT: Scalable Routing in Wireless Mesh Networks Using Temperature Fields, IEEE Symposium on a World of Wireless, Mobile and Multimedia Networks, 2007.

[19] L. Lertsuwanakul, and H. Unger, "An Adaptive Policy Routing with Thermal Field Approach", In Proc. of 9th Innovative Internet Community Systems, Germany, 2009, pp. 169-179.

[20] L. Lertsuwanakul, "Fuzzy Logic Based Routing in Grid Overlay P2P Network", In Proc. of 10th Internaional Conference on Innovative Internet Community Systems, Thailand, 2010, pp. 140-149.

[21] L. A. Zadeh, "Fuzzy sets: Information and Control", Vol. 8, 1965, pp. 338-353.

[22] H.Coltzau, "Specification and Implementation of Parallel P2P Network Simulation Environment", Diploma Thesis, University of Rostock, 2006.

[23] S. J. Mirabedini1, M. Teshnehlab, and A. M. Rahmani, "FLAR: An Adaptive Fuzzy Routing Algorithm for Communications Networks using Mobile Ants", Proc. Int. Conf. on Convergence Information Technology (ICCIT 2007), 2007, pp. 1308-1315.

[24] F. Jing, R. S. Bhuvaneswaran, Y. Katayama, and N. Takahashi, "Adaptive Route Selection Policy Based on Back Propagation Neural Networks", Journal of Networks, Vol.3, No.3, March 2008, pp. 34-41.