Ant Metaphor on Routing in Packet Access and Unstructured P2P Network

Roime Bin Puniran  
Faculty of Computer Science & Information System,  
Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia  
roime@tmrnd.com.my

Dr Muhammad Shafie B Hj Abd. Latiff  
Faculty of Computer Science & Information System,  
Universiti Teknologi Malaysia, 81310, Skudai, Johor, Malaysia  
shafie@fsksm.utm.my

ABSTRACT
Research on unstructured P2P network still remain a big challenge. Despite it easy to use, support inherent heterogeneity of peers and high resilient of peers failure, unstructured P2P still facing a critical issues on high traffic in a network and lack of bandwidth control. It may occur from the structure itself. Most unstructured P2P application in one reason simply implement message flooding during searching even it’s not fully perform inside the network, its still produce huge amount of traffic and could slow the network performance. Due to a big challenge in its implementation, several approaches were investigated to overcome an issue especially in traffic and bandwidth. This paper will overlook an ant algorithm used in 3 novel approach that manipulate ant algorithm in various different application; AntSearch, AntNet and AntHocNet. It’s impossible to cover all current implementation, but these approaches were subjects on major researches where particularly always being a referenced in several adaptive optimization algorithms in distributed computing studies. At the end of this paper, the comparison discussion will be made to evaluate all approaches used.

Keywords:
P2P, Ant, Overlay Network, Pheromone, Routing

1. INTRODUCTION
Ant, in a natural foraging behavior, are autonomous agent that travel through networks and spreads pheromone on their path [1]. In access technology (both in wireless and fixed network), ant algorithm metaphor was successfully applied. But, it’s still a doubt of its capabilities in a query routing implementation in distributed network.

The fundamental of routing technology describes that network routing refers to the process of selecting the next hop for an incoming data packet being forwarded based on information held in routing tables. Acronyms in between was, Ant Colony Optimization (ACO) meta-heuristic proposed by Di Caro and Dorigo [2]. Dedicated subset of ant algorithm was specifically designed for managing routing table in telecommunication network as well as in mobile ad-hoc network [3][4][6].

1.1 Query process
Query routing was used in distributed environment to enable peers finding a path towards a destination according to a given keyword id. Routing tables employed in each peer must be able to maintain information particularly about their neighbors and that information should be able to be accessed by the other peer as a path of references. Each peer has to issue the queries and in the same time, all peers in the network must be ready to collaborate to answer the query.

Ant based routing have already proven on network access routing [1] [6] but it implementation still on debut on distributed overlay network. Basically, in overlay distributed network, most query routing was derived from a routing technique that already implemented in access network technology in order to perform a searching works [5]. When peers start requesting for a resource in the network, they have to decide which neighbors probably could be chosen based on several metrics such as degrees of traffic, the condition of its pheromone trails, availability status, hit queries on
particular neighbors and Time-to-Live parameters (TTL). Those considerations must be statistical calculated to enable peers select the appropriate probability of next hop.

2. PREVIOUS RESEARCH ON ANT METAPHOR

The ant collective foraging behavior was first founded by Grass’è who name it as stigmergy [8]. Ant method was applicable into unstructured P2P network [7]. During foraging, ant use trail-laying and trail-following behavior according to an organic chemical produced called pheromone to find a shortest path between its nests into resources. It’s a quite simple behavior since it does require 3 basic principles:

1. Ant drops a pheromone each time they travelled from the source to the destination.
2. During its travel, it sense its path to chose; 
   a. If there is a pheromone already dropped, it follows the trails. The probability the ant follows the trails is proportional to the amount of pheromone its sense.
   b. If there is no pheromone, ant walks randomly.
3. Pheromone evaporated every time and if the trails are not being used, the pheromone will disappear.

The evaluation of ant algorithm in telecommunication was defined by Di Caro and Dorigo [2][9] in their long explanation on ACO. Dorigo and Stutzle [10] claim that ACO can be applied in any discrete optimization problem. Thus, ACO had applied three meta-heuristic building blocks where the sequences were not much different from Grass’è;

1. Trail following - Each ant will determine the solution of the optimization problem, where ant will determine their path through the network. 2 type of information needed; local information and global information.
2. Trail lying - It can be considered as a total cost of newly found path calculated.
3. Evaporation - Measured by how much percentage of pheromone evaporated in each iteration.

ACO then had been enhanced into many approaches such as Ant System (AS), Ant Colony System (ACS) and MAX-MIN Ant System.

a) Ant System (AS)

[11] and [13] explained that, AS exploit an algorithm which is a set of artificial ants cooperate to the solution of a problem by exchanging data through a pheromone value deposited on graph edges. It seems useful enough for discovering an optimal solution up to 30 cities. Time required finding such results making it unfeasible enough for a larger problem.

b) Ant Colony System (ACS)

ACS [11], [12] was an updated algorithm of AS. The routing mechanism derived from 3 main sequences of ACO building block; the state transition rule, pheromone update rule and evaporation. State transition rule enable ant to complete their tour towards a destination by choosing a city (or nearly neighbors) according to a probabilistic; cities which has high pheromone value and a shortest edge will be selected most. Pheromone update rule and evaporation consists 2 different rules but complement each other-Global pheromone update rule and Local pheromone update rule. Global pheromone update rule applied when ant has found its destination. Additional pheromone is dropped when ant found a new better destination path. Local pheromone update rule used to enable ant reach a less-visited destination and made it destination more attractive. It can be considered as a local tour between ant neighborhoods.

c) MAX-MIN Ant System

Its similar to ACS [14] in most of their pheromone update rule and ant transition rule.

3. ROUTING IN DISTRIBUTED ENVIRONEMNT

ACO was invented to solve a problem of Graph-Based Optimization Problem especially in centralized manner. Ant algorithm is exclusively point out on a modification of pheromone trail and it’s required an ant to have a local knowledge about the graph. Therefore, its technique can be suited into distributed environment like P2P without a requirement of centralized control.

To enable ants maintains a local knowledge about the graph, a routing table is specifically construct. There we an approach in network access technology where each node will have their own routing table and that particular routing table is responsible to keep update the pheromone value of their neighborhood nodes. Ant Colony Routing (ACR) [17] was then designed to adapt ant algorithm metaphor for managing routing table in telecommunication, ad-hoc mobile, IP-Based protocol and fixed networks [16]. There is a number of ant algorithm variant employ in data packet routing environment:
AntSearch was designed to overcome traffic issues caused by free-riding problem while searching in unstructured P2P networks [3]. During crawling, each peer maintains pheromone value to represent its hit rate in previous receive queries and records a list of pheromone value for its neighbors. Message query will be broadcasted to those who are not being considered as free riders. AntSearch algorithm basically improves search efficiency of a flooding by reducing the number of message sent by a peer. Regarding Figure 1, despite the implementation of control message flooding, requester peer only broadcast its query message into those who have higher pheromone value and considered as a non-free riders.

AntSearch employ 2 main phases; **Probe phase** and **Flooding phase**. **Probe phase** allows requester peer broadcasts a “probe query” or a message query to neighbors with small timestep flags Time to Live (TTL). Statistical calculation will be made about the total of files and peers founded. Probing table generated will summarized all of it. **Flooding phase** will take an advantage of probing table where, an appropriate neighbors with k value (k=% of peers should be chosen to flood a query) will be chosen and peers start to compute TTL value for the query. When a proper k had been chosen, requester peer starts its iteration process to find a result.

During iteration, requester peer will randomly choose a neighbor and calculate proper TTL when message transferred into a neighbor. Iteration process continued until the number of required files was found and all neighbors are visited. Pheromone value was sent by implementing PING and PONG method of Gnutella protocols.

Peers keep updating each pheromone record if they receive a query from a neighbor and remove pheromone if neighbor leave the network.

**Pheromone Table:** The main objective of pheromone table in AntSearch is to records the hit rate of previous in each neighbors to enable peer to select which neighbors has a high pheromone value to forward a query. It’s a primarily to measures which neighbors considered as free-riders. 2 values have to maintain by each peers when they joining into the network; hit queries $N_h$ and a number of total processed queries $N_q$. For a peer $q$, suppose that $d_q$ is a degree of peer $q$ ($q$ has a $d_q$ number of neighbors), the pheromone value $(pV_q)$ can be compute as follows;

$$pV_q = \frac{N_h}{N_q} \times \alpha + \sum_{i=1}^{d_q} \frac{pV_i}{d_q} (1 - \alpha)$$

In the first part, $N_h/N_q$ is the hit rate of previous queries of peer $q$. $q$ can be considered as a free-riders if the ratio of $N_h/N_q$ is very small.

**Algorithm:** AntSearch using control flooding technique to find a resource within time specified. There is 2 processes comprised; (1) Probe process and (2) Flooding process. In probe process, requester peer will floods a probe query to a few neighbors within a timestep flag TTL. Requester peer collects the statistic of a number of result and a number of peers visited when probe query finished. A probe table is responsible to summarize a result and for the next timestep, its will floods for a different k% of neighbors.

Since a probe table is already generated, in flooding process, for the first step, a proper k value will be choosing and in the same time TTL will be compute for the query. Regarding Figure 2, for each k, a number of peers the flooding has can be easily computed to enhance searching to retrieve a number of results.
Second steps was, to choose a proper \( k \). Requester peer have to compute TTL value for each \( k \), and finds a minimum value of it (TTL less than or equal to a TTL threshold, MAX_TTL). Then, requester peer starts an iterative process to find a result regarding \( k \) being chosen.

There is a tradeoff between choosing a larger \( k \) and choosing smaller \( k \). Flooding with larger \( k \) results more query message and will cause too much network traffic, meanwhile smaller \( k \) will produce longer TTL to be calculated, thus flooding takes longer time to reach a destination. The pseudo code of choosing \( k \) algorithm is showed in Figure 3.

---

**Figure 3** The pseudo code of choosing \( k \).

---

1) **Choosing \( k \)** (Probe_Table, MAX_TTL)
2) Begin
3) For \((k=0.1\text{ to }1.0)\)
4) \( H_k = \frac{h_k(N-n_k)}{n} \)
5) \( \text{TTL} = \log_{(28-\delta)} \left( \frac{H_k(Dk-2)}{dk} \right) \)
6) if \( \text{TTL} \leq \text{MAX_TTL} \) then break
7) End for
8) return \( k \)
9) End begin

---

**b)** **AntNet**

AntNet [18], [19] was design specifically for packet switched network. Its pheromone updating approaches were suited for a symmetric and asymmetric network. Ant metaphor used in the system is collaborated in order to build routing tables to adapt network traffic condition with aims to optimize the performance of an entire network.

The network consists of matrices \( N \times l \), where \( N \) indicated as size of nodes in a networks graph while \( l \) is a weighted of an outgoing links. To enable peer define a goodness of a destination route, a probabilistic value of \( P_{nd} \) will be used towards the neighborhood link \( n \) and with particular destination node \( d \). At a startup, all value for all entries in a routing table will be equal thus in such a way, sum of all entries will be 1. Each node \( n \) in a routing table \( Tn \) will manages an array \( M \) and responsible to store a statistical data towards a destination node \( d \) with appropriate time it require to travel from node \( n \) to node \( d \). There is 2 type of ant employed in AntNet; Forward ant and Backward ant (Figure 4).

---

**Figure 4** Forward ant and backward ant model of AntNet [16]

**Forward ant:** To let routing table \( T \) and its array \( M \) keep updated, so called forward ant will be created. Each node will create forward ant that build a path by randomly selected a route from itself towards a destination \( d \) by applying transition rule;

\[
P_{nd}^{f} = \frac{P_{nd} + \alpha I_{nd}}{1 + \alpha (|\text{neighbors}(k)| - 1)}
\]

When forward ant has reached its destination, timestep \( T \) will be calculated. \( T \) was used to indicates how well the route. The calculation result will be stored locally in each peer. Since it’s stored locally, destination node has to create backward ant in order to bring (copy) a result back to a requester peer within a stack following the nodes the ant travelled. Forward ant will be terminated after the creation of backward ant.

**Backward ant:** Backward ant is responsible to updates the cost value between source peer to a destination peer by altering the information of \( T \) and \( M \) in each of node visited by forward ant. All nodes used in a route towards
a destination will be given by 1. This can be achieved if a probability values for a node used in a path will be increased and a value for a destination node $d$ will be decreased according to:

$$P_{x,d} = P_{x,d} + r(1 - P_{x,d})$$
$$P_{nd} = P_{nd} - rP_{nd}, \ n \in \text{neighbors}(k), \ n \neq x$$

If backward ant arrivers at node $k$ from node $x$ for example, $T_x$ will be updated. The amount of pheromone will be derived from a comparison of time trip $T$ to the best time trip $W_{best}$. Since it’s a proportional between time trip and current load traffic, mean and variance will be taken to be estimated as well.

AntNet also handles the sub node where ant was traveled within several nodes towards a destination (Figure 5).

![Figure 5 Sub path from requester node $a$ to destination $d$](image)

To enable backward ant travelled from node $d$ to $a$, it will updates the entries;

a) $M_c(\mu_d, \sigma_d^2, W_{best_d})$

b) $M_b(\mu_d, \sigma_d^2, W_{best_d})$

c) $M_a(\mu_d, \sigma_d^2, W_{best_d})$

and increment the value of;

a) $T_c(P_{cd})$

b) $T_b(P_{bd})$

c) $T_a(P_{ad})$

Basically, the backward ant knows the trip time in each node it’s travelled.

AntNet also deliver a strategy to prevent cycles. Forwarded ant will manages each node had been visited and each time it travelled into the same node, that particular node will be excluded. When time span t spent in a cycle more than 50% of ant total life time, the iteration will be terminated.

c) AntHocNet

AntHocNet [6][20][21] founded by Ducatelle, Di Caro and Gambardella designed specifically for mobile ad-hoc network (MANET) technology [22][23] due to frequently changing of technologies and limited bandwidth on most mobile access network that result very limited route established in such network. AntHocNet actually inspired from ACO framework. It consists of 2 main components; reactive and proactive.

The idea behind AntHocNet was derived from Ant Based Routing [15] algorithm where several approaches were born regarding on that (Ant Based Control [24] and AntNet). The combination of reactive and proactive makes AntHocNet as a hybrid approach to optimize a solution of indicating the quality of path from source node towards a destination by means of artificial pheromone behavior. A possible way to define a path with a great quality is by spreads a packet into a multiple path following the pheromone value along the path.

**The algorithm:** Reactive part starts when starts node $s$ starts a process of finding a destination node $d$ during data session. $s$ will determine an update status of routing information towards $d$. If there is no up-to-date information for $d$, $s$ will sends out ant-like agent called reactive forward ant to observe the path condition towards $d$. These ants will collect all the information according to the quality of path during its travel to $d$. When they reached in $d$, they will become a backward ant where they will travelled back to $s$ by tracing the path they used and updates a routing table $T^i$.

$T^i$ in every nodes $i$ contains an entry of each destination $d$ and possible next hop $n$ of value;

$$T^i_{nd} \in \mathbb{R}$$

$T^i_{nd}$ can be considered as a path quality estimation between $n$ to $d$ called pheromone. Pheromone table in each node basically can indicates different multiple path from $s$ to $d$. Nodes will select its next hop according to a probability proportion to its pheromone value.

Proactive part starts when path was established. $s$ sends proactive forward ant to $d$ with a probability to be broadcasted or not. There is a big chance for $s$ to explore a new path if proactive forward ant being broadcasted.

**Reactive part:** It used to enable $s$ looking for path towards a destination $d$. When some nodes receives an ant from a same source, it will compare a path an ants used to travelled before. And the number of hop as well as a travel time will be calculated. If it’s within a certain
factor, nodes will forward these ants. This technique can be used to eliminate an overhead by throw an ant that used a low quality path.

Reactive path used to keep a list $\mathcal{P}$ of nodes $[1,...,n]$ it has visited. Backward ant created upon $s$ arriving at $d$ will track back a path by retracing $\mathcal{P}$. Routing table will be updated by backward ant by calculating $T_{\mathcal{P}}$, time an ants takes to travel along $\mathcal{P}$ towards the destination. $\mathcal{P}$ basically a total of local estimated time $\hat{T}_{i\rightarrow i+1}$ in every node $i \in \mathcal{P}$ of the time towards next hop $i+1$:

$$\hat{T}_{\mathcal{P}} = \sum_{i=1}^{n-1} \hat{T}_{i\rightarrow i+1}$$

Value of $\hat{T}_{i\rightarrow i+1}$ was defined as $(Q_{mac}^i + 1)\hat{T}_{mac}^i$ that is an average of one packet being send. The value of $\hat{T}_{mac}^i$ times the total packet in queue (plus one) will be sent at MAC layer, $Q_{mac}^i$.

Meanwhile, an average of time elapsed between packet accepted in MAC layer and previous successful transmission will be calculated on $\hat{T}_{mac}^i$.

If $t_{mac}^i$ used to calculate a time ant took to send a packet from node $i$, node $i$ updates will be estimated as follows:

$$\hat{T}_{mac}^i = \alpha \hat{T}_{mac}^i + (1 - \alpha) t_{mac}^i$$

with $\alpha \in [0, 1]$.

Ant that arrive in to a node $i$ from its neighbor $n$ will creates an entry in routing table $T_i$ and takes $n$ as a next hop to reach a destination $d$. This entries will contains a pheromone value $T_{nd}^i$ which is an indication of a quality of path towards $d$ over $n$. Pheromone value represent a total average of cost that consists of time used and a number of hop used to travel to $d$ through $n$.

If $\hat{T}_{i\rightarrow d}$ is a total time used to travel by an ant and $h$ is a number of hop, the total pheromone value could be calculated based on:

$$\tau_{id} = \left(\left(\hat{T}_{s\rightarrow d} + h \hat{T}_{hop}\right)/2\right)^{-1}$$

where $\hat{T}_{hop}$ is a fixed value that represents time usage to take one hop in unloaded condition.

When there is exists multiple next hop towards $d$, ant will randomly select one of them by applying:

$$P_{nd} = \frac{\tau_{nd}^2}{\sum_{i \in N_d} \tau_{id}^2}$$

, with a probability $P_{nd}$ of next hop $n$ assigned as a square of its pheromone.

**Proactive part:** Proactive path is responsible to do path maintenance and path rectification in already existing path. In this process due to the ad-hoc network itself that compromise with the moving nodes, broadcast will be used when there is lack or no pheromone value pointing to a destination node in some neighbors. To reduce this issue, AntHocNet used 2 approaches;

1. Proactive ant which is doesn’t finding a routing information path within 2 hops will be deleted. But, it will cause an ant that searching for a new path will concentrate only the current path.

2. By using Hello Message within a regular interval. For example, if node $s$ send a hello message to their known neighbor $b$, but doesn’t receive any hello message form $b$ within a period of time, pheromone information of node $b$ will be remove in routing table of node $a$. This technique used to make a local repair of broken link as well as to discover a new path. Additionally, if node $s$ receive a hello message form unknown node $c$, it will add a pheromone value of node $c$ into its routing table.

4. **DISCUSSION**

Refer to [4], even though each forward-backward ant algorithm are complex enough in order to find a quality route to adapt the table routing on single source-destination path, its still cannot solve a problem on global routing optimization. AntNet, by using a
probabilistic technique on its routing table, it consists of triple positive effect:
1. Redistributing data traffic on new routes much better.
2. Built-in evaporation features are provided within ant.
3. Arrival rate of ant exploration rate allow ant to assign a cumulative reinforcement.

The strength of stigmergy-based control is coupled to a model of building activity. Information collected by ants during travel not only used to modify the routing table information, but also to build a local model for a better direct of routing during routing table modification. The difficulties to apply AntNet algorithm into P2P query network occur because of each node must be able to have knowledge about the other node in the network to apply random selection node to forward ants. AntNet was specifically applied in datagram access network. But, certain components still can be adopted into P2P query network [16]. A distributed data structure in AntNet can be used in P2P environment as well except the entry in a routing table that corresponding to destination node must be replace into a keyword id of the query. Other than that, the concept of forward-backward ant can be used in P2P environment except in AntNet, formal ant usually have a knowledge about well-known destination node, while in P2P environment, node doesn’t know the information about the target/destination node. Preventing cycle technique in AntNet can be adopt and implement in P2P environment without much changes. [25] explained that AntNet algorithm still employed a similar general strategy of Ant Based Control [24]. Its only have to maintain an optimal path in their routing table and then automatically adapts their data to enable spreading.

AntHocNet based on [25] is a hybrid algorithm that was proposed to overcome a weakness of ABC and AntNet on multipath routing. AntHocNet used 2 technique in their implementation; reactive and proactive forward. Reactive forward used to enable peer send out and ants in order to observe for a path towards a destination. Once its reach a destination, a backward ant will be created, where it will trace back a source node by following a path it has visited during searching and in the same time, it will updating a pheromone table. Proactive forward ant will be launch once backward ant reach into a source path to keep track the goodness of the path meanwhile in the same time, a new path could be explored. [20] point out that, ant based method can be fully applied in mobile ad-hoc network because of it can be used to adapt to a network chance, its robust enough and can provide multi path routing as well. Reactive forward ant basically comprised with a forward-backward ant applied in AntNet together with the exploiting and exploring strategy known in ACS.

Meanwhile, AntSearch development purposes was to eliminate free-riding and in the same time it will reduce network traffic by only sending a query message into a peer that considered as non free-riding. By using Gnutella PING-PONG protocol for sending a pheromone value, it makes AntSearch implementation as simple as possible and make it easy to be used in a real system.

5. CONCLUSION

Ant algorithm was mostly used and well proven in a datagram packet access network such as MANET, ad-hoc network and packet access network. It still makes a debut on overlay P2P network. Some research [16] had been done in order to implement ant algorithm into query overlay P2P network but it still dragged much component especially its algorithm from a system that run in access network environment.

References


