

An on demand path marking and capacity reservation method using Split Agent

Constandinos X. Mavromoustakis and Helen D. Karatza

*Department of Informatics,
Aristotle University of Thessaloniki,
54124 Thessaloniki,
GREECE
{cmavrom, karatza}@csd.auth.gr*

Abstract

Different schemes for large scale networks hosting distributed applications have been recently adopted for network path marking based on adaptive behavior of swarm-based agents. Topologically complex networks must use efficient routing methods in order to route data traffic from a source to a destination. In this paper the Split Agent-based Routing Technique (SART) is applied to a network in which a certain amount of data traffic is injected to mark different paths. Split Agent-based Routing Technique (SART) [20] is a variant of swarm-based routing [9, 22] where agents are split after their departure to the next node on a hop-by-hop basis at the same time performing path marking. Packets that are delay sensitive are marked as prioritized. Agents recognize these packets as being a part of, and try to influence the two way routing tables. Thorough examination has been done for the performance and path recoverability using SART algorithm for marking any requested path, taking into account a number of metrics. It is shown that the split agent scheme offers an efficient on demand path marking and path-capacity reservation in a decentralized manner (particularly efficient for large scale networks). SART method undoubtedly represents a solution for significant network optimization particularly for the equal share of network resources. Also it is shown that this scheme offers a way to increase overall performance.

1. Introduction

In a communication network, intermediate nodes are purely responsible to forward calls from a source to a destination. This requires that a mechanism exists in order to find the cost effective path from a source to a

destination. Agent-based routing has nowadays become one of the most attractive routing methods in wired and wireless communication networks. Combining various hybrid schemes of different agent-based methods it enhances the overall performance of the system offering to end user reliability and integrity. The objective of agent-based routing is to achieve high resource utilization and to reduce user contention for network resources.

As load is not uniformly distributed, network resources decrease in the form of efficiency, bandwidth, processing power and memory, leading network to an unpredictable behavior. It becomes evident that the mechanism used for load balancing has to avoid overloaded nodes such that transmission delays (latencies) are minimized, whereas, whenever load conditions are drastically changing, alternative routes should be activated.

When network dimensions increase, traditional routing algorithms described in [1, 2] and [4] do not scale well particularly for large scale networks. In topologically complicated networks -in terms of large scale data distribution- the lack of adaptability of routing algorithm could become disastrous for the offered Quality of Service (QoS). Routing algorithms must have the ability to adapt in any network changes and cope in anytime network state changes (capacity of nodes and links, traffic within paths, load changes etc).

Previous work [4] on efficient network routing (distance vector and link state) have shown that the centralized control of information in a network, "sacrifices" delay for informing the nodes for any changes due to the central supervision of a part of a network. On the contrary, decentralized routing schemes [5] involve a number of controllers each of which is supervising a part of a network, allowing the central control to be applied locally to a part of a

network and then all "local" supervisors are cooperating deploying a decentralized exchange of information.

Introducing agent's idea in large-scale systems is nowadays growing, meeting the advances in telecommunications. The spread of the Internet and electronic commerce mean that information infrastructure operates as a global dynamic system. As time passes, the density and diversity of interconnections in such system will increase rapidly. Moreover, such systems are required to service the needs of a diverse set of users (whatever their distinctive needs), not just a virtual 'representative' user. Thus, such systems must adapt to personal requirements, by providing highly customized packages of services. Simultaneously providing highly diverse services to a huge user population in an enormous, interconnected system is a complex task beyond centralized management techniques.

The proposed scenario benefits the adaptivity of QoS in heterogeneous networks where different levels of QoS occur. It enables distributed resource and service management across heterogeneous platforms and a common base for autonomic management solutions. Additionally it offers a Coordinated service management (CSM) in which the desired user-level qualities of service is monitored and can be modified to service level agreements (SLAs) negotiated with various services located within a single enterprise or operated by service providers.

This work is an extension of our previous work [20] where a hybrid technique is designed using an agent-based routing control. In that paper a distributed hybrid method Split Agent-based Routing Technique (SART) using swarm-based message passing [5, 16] is presented which provides a model for distributed network data flow organization. This hybrid scheme is based on swarm intelligence and is applied to a packet network that exhibits different class of service behavior for all packets. SART method contributes to "mark" a proper path and reserve the required capacity for transferring the data packets with priority, based on the delay sensitive coefficient of each packet. We have considered number of metrics that are associated with network performance and the characteristics of agents' behavior. The metrics measured can improve the QoS offered. It is shown that by marking each time the path which corresponds to link capacity and buffer level of each node in a distributed way, lots of QoS requirements can be satisfied. First of all the amount of bandwidth that is reserved by an application is a main issue. Network operators must be in a position to predict the required bandwidth by the real time application. Another big issue for the QoS is that

different types of applications have different service requirements. For example if an on-line user is playing a simulation game it might not be needed to have the same bit rate than a user playing an RPG game on-line with other users on the network. Of course a crucial metric is response time-latency- where some milliseconds might have disastrous effects for users demanding multimedia services, since they cannot any longer actualize their request. These issues are indeed solved by using the SART method, which ensures the QoS to end-users. More specifically SART can inform on a hop-by-hop basis the exact neighbors the same time spreading and affecting their routing tables.

The organization of the paper is as follows: In section 2 a description of the basic principles of the Split Agent-based Routing Technique (SART) method is presented as well as SART's contribution to path marking and capacity reservation. In section 3 the simulation results are shown and finally in section 4 the concluding remarks and further research is presented.

2. Agents and their contribution to path marking and capacity reservation

2.1. Agent-based technique

Agent-based approach was first introduced and standardized by Appleby and Steward's mobile agents algorithm [5]. Further studies [6, 8, 21, 22] have shown that an ant-like mobile agent algorithm could be applied to a network with significant optimization of the QoS metrics of the network [8, 22]. In [16], Dorigo and Gambardella used the metaphor of trail laying by ants to certain combinatorial optimization problems [3, 6].

Several agent-oriented approaches [5, 6, 8, 9, 14, 15] have recently been proposed that appeal to principles extracted from Swarm Intelligence (SI) and aspire to solve routing problems to wired and wireless communication networks. On the other hand, within these methods there are some trade-offs that have to be taken into account. These trade-offs deal with network overall performance, such as generated overhead in messaging passing for agents communication, network utilization, simplicity for the implementation etc.

Agent based network routing could be biologically inspired and based on insect colonies which exhibit a simple behavior for their communication and living. Real ants have similar behavior with agents and are represented in the network, as artificial agents that bias the network collecting useful information for the whole environment through their hormones called

pheromones. In this research a hybrid (proactive and reactive) agent behavior is used in which ants are adapting their behavior to network circumstances simultaneously splitting themselves for passing information to neighbor nodes.

Real ants in nature can find the shortest path between two nodes (food-source, nest-destination) by exploiting pheromone information onto ground. Ants simply deposit pheromone on the route while walking, which influences the rest of them to follow, in probability, the same route. In Figure 1 there is an illustration of this principle called the binary bridge experiment. Two ants are leaving from their source node the same time by following different paths, which are laid, with pheromone trails. Both paths are being chosen with a certain probability. The ant that chooses the shortest path returns to its nest-source first. The shortest route will influence more ants afterwards because it contains a double quantity of pheromone (the quantity of pheromone while spread on the shortest route is higher in density) [8]. The deposited pheromone has intensity inversely proportional to the length of the path.

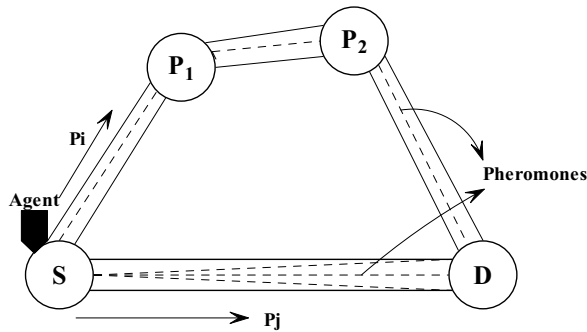


Figure 1: Migration strategy using edge pheromones-trail laying principle.

The explored pheromone quantity could be 'compiled' into a concept for network bandwidth. Pheromones correspond to bandwidth in terms of capacity reservation as shown in Figure 2. Bandwidth is inversely proportional to the capacity reservation of the link as pheromone is inversely proportional to the length of the path. This pheromone corresponds to bandwidth in terms of capacity reservation, which means that the higher the probability is the more bandwidth reservation. Pheromone values are normalized to capacity corresponding values and registered to routing table.

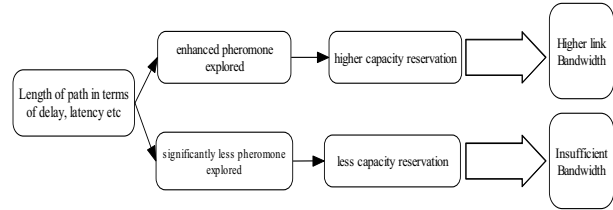


Figure 2: Pheromone - bandwidth relation and intermediate processes.

The generic ant-based routing scheme consists of three agent types: explorers, allocators and deallocators [5, 22]. Explorers exhibit the foraging behavior and follow trails of pheromones laid down by previous explorers (positive feedback). Allocator agents traverse the path determined by explorer agents and allocate the bandwidth on the links used in the path. When the path is no longer required, a deallocator traverses the path and deallocates the bandwidth (probabilities) used on the links. Figure 3 shows the positive feedback mechanism that forms a continuous circle, so the shortest path is strongly marked. Ant-based algorithm generates mobile agents embodied in packets at regular intervals within the nodes of the network. These packets which agents are a part of, select a purely random destination and try in the next time step in the network to reach the destination by selecting the best path from their source. Roughly speaking, ants bias the network at a steady state [8, 19-22] achieving load balancing. This method appears to be quite attractive particularly for large scale systems since it allows the parallel operation of ants' packets activated in the network¹.

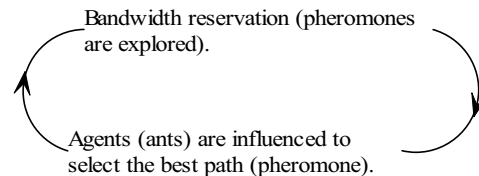


Figure 3: Positive feedback mechanism for reinforcement of the agents.

2.2. Agent-based network operations

In a network using a typical decentralized routing model, each node maintains a routing table indicating where the packet has to go in order to reach the final destination. Agents adjust the table entries continuously affecting the current network state. Thus, routing tables are represented with pheromone tables

¹ In previous researches packets are waiting until an algorithm or a heuristic finds a path.

having the likelihood of each path to be followed by the artificial ant-packet. Pheromone tables contain the address of the destination based on the probabilities for each destination from a source in a bi-directional format. In our implementation each packet-agent launches in the network and influences the pheromone table [9, 22] by increasing or reducing the entry for the proper destination using antipheromone [21]. The pheromone table at each node with neighbors can be measured as:

$$R_i = \left[r_{1,m}^i \right]_{n-1,k(i)} \text{ with } n-1 \text{ destinations and } k \text{ next nodes for } N_i. \quad (1)$$

Artificial ants are biasing the network by generating ant-packets at every simulation time step destined to every node randomly. In the network ants are walking according to probabilities assigned in pheromone tables and they are visiting one node at every time step, randomly selected. In this way ants increase the entry corresponding to the node from which it came by:

$$P = \frac{P_{old} + \Delta P}{1 + \Delta P} \quad (2)$$

where ΔP is the quantity of pheromone increased and P_{old} is the previous entry. The other entries in the table of this node are decreased accordingly following the formula:

$$P = \frac{P_{old}}{1 + \Delta P} \quad (3)$$

Routing tables contain a two-way pheromone table parameters (bi-directional links with different capacity) which are maintained in each node, and are expressed as:

$$P_{k(i)}^{i \rightarrow n_i} \text{ and } P_{k(i)}^{n_i \rightarrow i} \quad (4)$$

where $k(i)$ are the next nodes² for N_i , n is one of the $n-1$ possible destinations and n_i is the possible next node at a certain time step. All probabilities are thresholded [18] between $\frac{1}{(\text{number_of_neighbors})^2}$ and 0.75 in order to prevent the pheromone saturation state.

Artificial ants are dying when they reach their destination. In order to force ants to choose short lengths, pheromone increase reduces progressively with the age of the ant using the equation:

$$\Delta P = \frac{a}{t_{age}} + b \quad (5)$$

² Nodes can be both endpoints (can be source and destinations) and switches (can perform routing functions).

where t_{age} is the number of time steps passed since the birth of the ant and a and b are proper constants chosen by the designer of the network. Thus if t_{age} is small enough, each ant can influence the path to destination by increasing the probability P in the pheromone table entry. Congestion problems are faced, by delaying split-agents (as it will be shown in section 2.2) at congested nodes according to the following formula:

$$\text{delay} = [C \cdot e^{-d \cdot s}] \quad (6.1)$$

$$\text{or } T = \sum_i \text{delay} = \sum_i [C \cdot e^{-d \cdot s}] \quad (6.2)$$

where in (6.1) and (6.2) C and d are constants and s is the spare capacity of the node. Equation (6.2) evaluates the sum of delays-latencies of the nodes, which correspond to the life of the split-agent.

An independent operation but in parallel with ants is validated for routing the packets. When a packet is placed in the network, a set up path is initiated from source to destination. Every node has a certain probability to be the destination node. At every time step packets are generated at each node by following Poisson distribution (7.1) as depicted in [24], destined for a random destination uniformly selected.

$$P(X = x) = \frac{[\mu^x \cdot e^{-\mu}]}{x!}, \text{ for } x = 0, 1, 2, \dots \text{ with } E(x) = \mu \quad (7.1)$$

The mean duration of each packet is below a certain time limit for delay sensitive packets and randomly selected for all the others.

The route, where intermediate nodes have large pheromone quantities, is selected as the best-chosen path (an on demand behavior). If the destination can be reached on a hop-by-hop look-up table method, the route is valid otherwise the packet is blocked [22] and lost. Routing table entries updates are measured [21-22] according to the following:

$$r_{i-1,s}^i(t+1) = \frac{r_{i-1,s}^i(t) + \delta r}{1 + \delta r} \quad (8.1)$$

where δr is the step size parameter and s is the source node. Similarly for all neighbors to i , $r_n^i(t)$ is found that:

$$r_{n,s}^i(t+1) = \frac{r_{n,s}^i(t)}{1 + \delta r}, n \neq i-1 \quad (8.2)$$

Previous values and updated entries of the pheromone table of node n should satisfy the following:

$$P_b = \sum_i r_{n,s}^i = 1 \quad (9)$$

In this way smart data packets and ants have an interaction in the means that ants affect the routing tables and data packets influence the service rate of the

traffic on nodes, which affect the ants with the delay mechanism. This agent-oriented approach encompasses the generic agent based concept which enables agents to move around the network, gathering information about the topology of the system and the traffic at a discrete time.

2.3. Split Agents for path marking and capacity reservation for large scale systems

Agent-based concept is a set of dynamical processes whereby local supervisions enable distributed global view of a system from the interactions between the agent-based entities and upper and lower level processes. In large-scale networks, congestion control algorithms must be able to offer scalability, robustness and stability in order to handle all kinds of situations including the unforeseen network changes. The SART method undoubtedly represents a solution for significant network optimization particularly for the equal share of network resources.

Based on the class of service at any time in the network, nodes must be able to service packets (route to destination) at a limit below their capacity limitations. Particularly for delay sensitive services (multimedia packet transmissions) the routing algorithm must guarantee that the maximum delay of each packet transmission is less than the preset bounded end-to-end delay. Split Agent-based Routing Technique (SART) is a variant of ant-based routing technique [9, 22] that allows agents to be split after their departure from source node to the destination node on a hop-by-hop basis. Based on the idea of Fisheye³ routing [12] where each node can see its absolute neighbors, SART is applied only to the path that packets follow to reduce resource overhead.

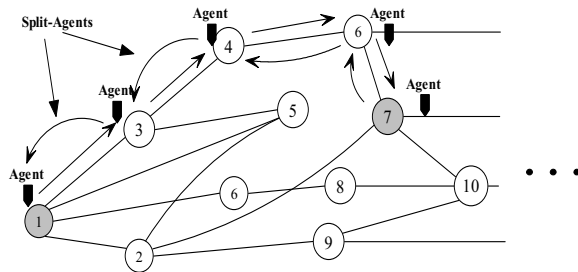


Figure 4(a): Split Agent-based message passing and on demand path marking on a hop-by-hop basis.

³ Eye captures with high detail the pixel near the focal point and detail decreases as the distance increases. Fisheye technique uses a node as focal point

Split Agent-based Routing Technique (SART) Pseudocode

```

Initialize pheromone table on all elements;
While (stopping criterion not met){
  Packets_generation();
  Add_packet_header(); //generate smart packets
  packets_launche_generation_at_each_node_randomly;
  measure_network_capacity();
  if (link(i)_capacity() <= defined_capacity){
    Nu_pt = Generate random number(0, 1); //for influence the dest_pher_table
    BWV_reservation_Agent_process();
  }
  else decrement_ph_table_entries();

  While (active packets exists from any source node to dest_node){
    For (each active packet){
      Determine probability of choosing each node;
      Nu_pt = Generate random number(0, 1);
    }
  }
  Estimation_of_delay {
    Return (delay_age);
    Estimate_pack_delay();
  }
  Do Packet_time_step();

  Smart_packet_movent {
    For (each node in the path evaluate the p.t.)
    {
      search each row and column;

      for (each entry in the p.t.){
        if (present_p.t. > previous_p.t.)
          increment p.t. (pheromone); // pheromone mechanism
        else
          decrement(pheromone); // antipheromone mechanism
      } //for loop
    } //for loop

  }
  Normalize_p.ts (entries);
  // Smart_packet_movent

  Split_agents {
    Set path();
    N-hop within path_nodes_estarrnition(); // it is just like reasseble_path();
    While (N!=source_node){
      Split_agent_packet_visit ((N-hop)-1); //agent is splitted after departure
      //from source node to the destination node
      //on a hop-by-hop basis.

    } //while
  } // Split_agents

  Destination_split_agent {
    Estimate_entries();
    While (packets_reached_destination) {
      Split_agents();
      While (previous node is NOT the source node){
        If (return_update_NOT_occured){
          Influence_routing_tables();
        } //if
        else
          update_entry_in_path_for_source_node();
      } //until it reaches the source node
    } //main while
  }
}

```

Figure 4(b): Pseudocode for split agent-based method.

In Figure 4(a) the split agent-based technique is illustrated. The agent forwarding mechanism is activated for sending from a source node packet to destination node. A packet needs to be forwarded from node 1 to node 7. Agents are moving according to pheromone tables (agents have already biased the network [20-23]) from node 1 selecting node 3 as the

next node. Once agent reaches node 3 it is split to an ant-based agent which keeps its journey to destination (node 7), following the same rules as the ant-based routing scheme [8, 9, 22] and also it is split to notify agent. The split agent is mainly the allocator agent which informs the previous node where it came from as well as the link capacity (uplink and downlink) and submits an acknowledgement about the state of the packet (received or not). Yet split agent informs previous node about possible link overflow. After the split agent returns back to previous node it dies. This procedure continuously occurs until agent reaches node 7 (destination). Additionally the returned-split agent if necessary reduces the priority of node (pheromone table) for delay sensitive services, which is totally based on the class of service of the packet. In Figure 4(b) a pseudocode is provided for split agent-based technique including the corresponding functions of SART path marking and swarm agent-based process.

Agent has to reach destination in a time interval depending on the packet. Packets marked as sensitive activate agent's time limitations not reach t_i , where t_i is the time interval for packet i . Roughly speaking packet $p_i(t-t_i)$ must not exceed the limit of $t \geq t_i$. This predetermine time intervals for delay sensitive packets are called Epochs [24]. The overall number of sensitive packets at any time in the network must not exceed the 7% of the total number of packets, during initialization period and transient state.

In split agent-based implementation we also used the pathrater agent that run on each node's structure. Pathrater agent is informing the visiting ant-based agent about the spare buffer capacity of the node and maintains rating for each other node it knows that arises from routing table entries.

At any time in the network there must be some level of assurance that traffic and service reliability are satisfied. QoS is very crucial for end user satisfaction. The QoS must involve the minimization of delivery latency, minimization of delay variation (delay Jitter) and provide the required data throughput capacity. To handle the proper flow of traffic in the Network, our proposed SART method ensures the QoS to end-users. SART method can significantly improve the QoS using the smart packets for capacity allocation for allocating different levels of service to different users depending on the traffic of the paths. SART method bounds the delay and overall delay by using a hybrid reactive behavior.

The above split agent-based technique enables agents to mark each time the path which corresponds

to link capacity and buffer level of each node in a distributed way [22]. In the described scenario link capacity is 2 Mbps (bi-directional) and buffer capacity is 680kb. The network traffic is modeled by generating constant bit rate (CBR) flows. Each source node transmits one 512-bytes (~4Kbits) packet. In the prescribed manner packets are routed to each destination by selecting the resource sufficient path and additionally packets are prioritized in a form based on the class of service notifying agents to detect for a new data path.

SART can thoroughly offer workflow services. These services can support the coordinated execution of multiple application tasks on multiple distributed Grid resources. Also these services can be expanded into a problem determination for distributed computing which include trace and log mechanisms with event tagging (through discrete pheromone execution) and event correlation capabilities.

2.4. Network generated overhead

As the number of information factors increases, the communication overhead becomes prohibitive. As known routing overhead is the number of control packets that are sent relative to the data packets. In the described Split agent-based scenario packets are delivered from one node to another causing traffic to increase significantly by forwarding packets from source node to all intermediate nodes and leading to destination. Though routing related transmissions can affect the overall performance of the system causing significant end-to-end delays and delay variations. The split agent-based scenario does not use specific control packets like other routing schemes since agents pursue the control by being a part of the packets (smart packets). Therefore agents map and control the traffic at any time during transmission in the network and overhead is potentially reduced. Another issue that has to be taken into account for resource overhead reduction is that split agent scheme enables each node to view its absolute neighbor(s), thereupon-additional queries to neighbors are avoided. Finally the inexistence of generated overhead is as a result of the non-transmitted routing tables values or other information blocks to neighbors or to all nodes of the network.

3. Simulation and results obtained

To demonstrate the design methodologies discussed in this paper, we performed exhaustive simulations to a partially meshed large-scale network consisting of 180

nodes (nodes are partially interconnected to each other). Each node's buffer has a relatively large capacity (680kb). Network's performance is examined through a number of various metrics that characterize routing-path marking algorithm's efficiency.

We have modeled and simulated (based in C/Objective C programming language) the previously discussed scenario soundly based on our source code and our C libraries built. There is no specific underlying platform for this implementation since agents represent individually the packets entering the system and being a part of them so-called smart packets. The program follows the general guidelines proposed in [8, 21] and [22].

Average utilization of nodes versus the total buffer capacity that is used to carry data traffic is shown in Figure 5. A significant reduction is apparent in the number of ants used in simulation as shown in Figure 6. Split agent method reduces the number of ants since it is applied only to bandwidth-saturated paths. These paths occur only in case of routing-path lock [20].

Fluctuations in Figure 7 are only presented in transient state [22] where network nodes are not biased by the SART algorithm. In the steady state where nodes are biased SART method significantly reduces buffer overflow in saturated paths.

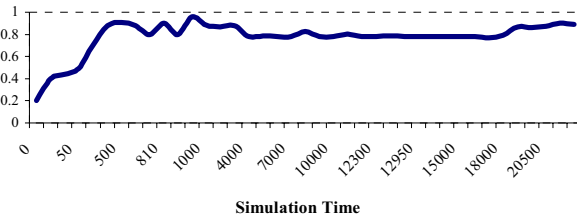


Figure 5: Average utilization of nodes during simulation time.

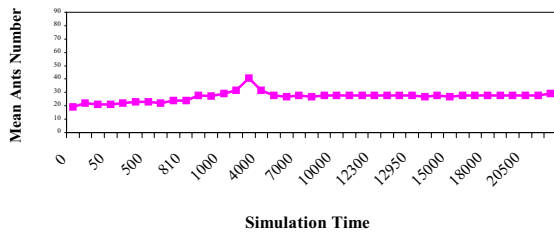


Figure 6: Mean number of ants used in simulation.

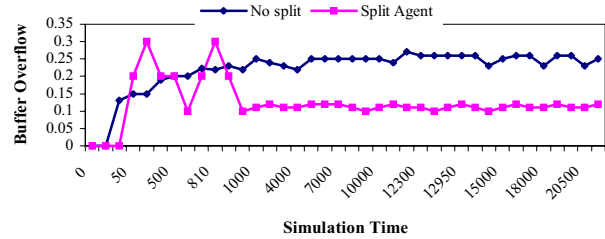


Figure 7: Buffer overflow in saturated paths using SART and the generic Ant-based scheme.

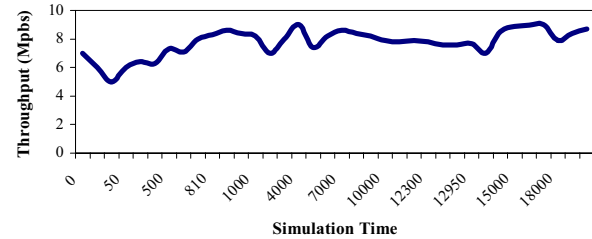


Figure 8: Throughput of the system and average throughput in Mbps.

The throughput of the system is shown in Figure 8. Total throughput of the system is estimated as the *total link utilization* $\cdot (1 - P_{loss})$. The throughput of the system is relatively high for both states (transient and steady state).

Figures 9 and 10 show packet loss and packet delivery rate. When the system overcome the transient state [19, 22] then packet loss is stabilized below 0.2 and succeeded packet delivery rate is solidificated at a relatively high rate of 0.979. Figure 11 shows the different spikes in bandwidth reservation grade at 2 Mbps. Path marking of SART contributes for increasing bandwidth capacity reservation which in turn increase overall link bandwidth. The shaded area represents the possessed bandwidth at 2 Mbps link-capacity rate.

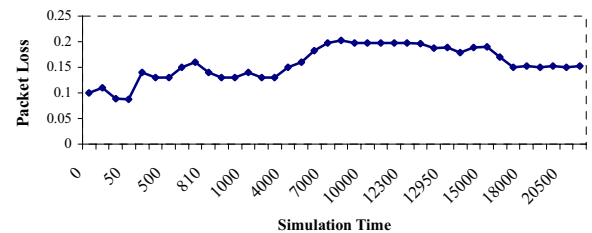


Figure 9: Packet loss using marked paths.

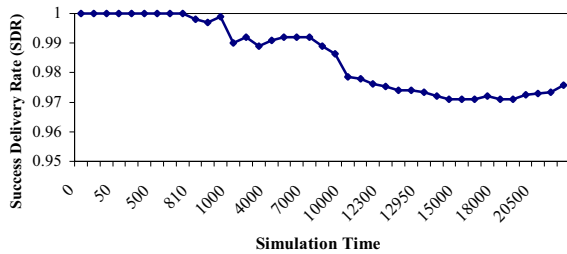


Figure 10: Packet delivery rate (SDR) performance.

Extremely lower buffer overflow percentages in split agent-based scheme compared with generic ant-based routing scheme [22] are presented in Figure 7. Buffer overflow shows that in the steady state the split agent scheme reduces the overflow of any node buffer almost 150%.

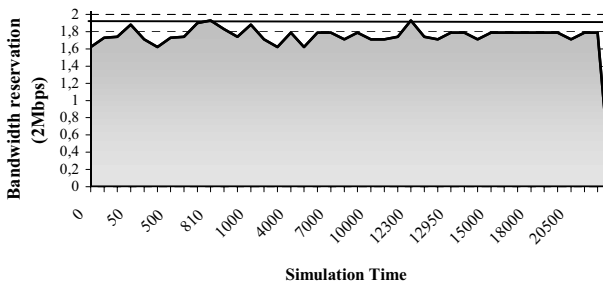


Figure 11: Bandwidth reservation grade at 2 Mbps link-capacity rate.

As shown in Figure 12 the path reuse frequency—which is a remarkable metric for measuring the efficiency of SART path marking capability, behaves at a relatively high percentage. Paths which are available based on their remaining capacity have overcome the routing lock and are again chosen as best effort path for transferring a proper packet.

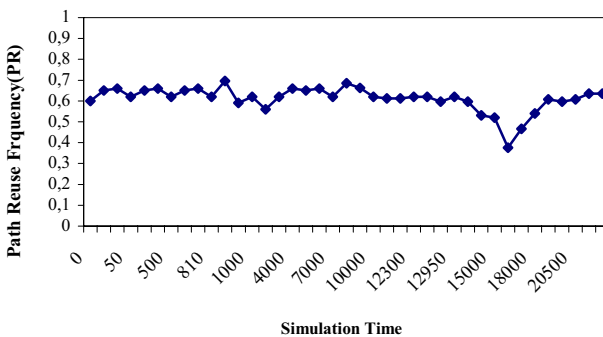


Figure 12: Path reuse frequency.

In Figure 13 as the number of hops increases throughput drops drastically. The explanation for this issue is that as load balancing occurs the paths become

overloaded more and more. At this point we must state out that the number of hops does not affect the efficiency of SART scheme for path marking.

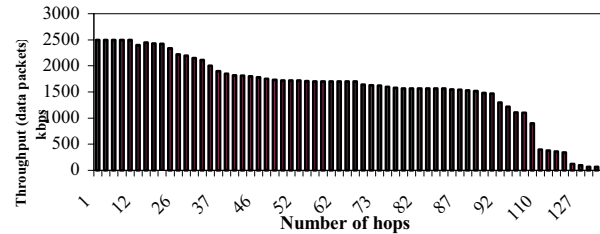


Figure 13: Throughput versus the number of hops.

Generally split-agent algorithm for path marking and capacity reservation scaled well. The significant superiority of split-agent control system against other decentralized routing schemes shown in [20] proves the adaptive behavior of the method under heavy traffic and limited buffer capacity. Split-agent algorithm is shown to encourage new paths to be marked and this has as consequence the reserved extra capacity on their links which in turn results in better utilization of network resources. Throughput of the system seems to be high enough to handle QoS based services while generated overhead is negligible due to smart packets. On the contrary with other routing algorithms, where a major disadvantage is the unbounded resource overhead [4, 10, 15], split-agent method handles routing information-path marking and capacity reservation successfully with embodying swarms in the data packets (smart packets) resulting in an improvement on the overall throughput response of the system. Furthermore contrarily with the major drawback of prioritization of IP packets (IP networks cannot facilitate prioritized traffic), SART algorithm partially faces this problem by adding intelligent packets (smart packets) into the network.

4. Conclusions and further research

The idea of developing a path marking scheme based on agents does not imply that network parameters would be improved and consequently the performance of the system will. In this paper, we have implemented an agent-based scheme called split-agent based technique used in large scale network for path marking and capacity reservation on demand. We have taken into account several measures for large scale packet switched communication networks. Split-agent method substantially contributes in congestion avoidance in the network's paths. Messages between nodes are replaced by agents embodied in data packets where simultaneously bias the network parameters by

laying pheromone on route from a source to a destination. This pheromone corresponds to bandwidth in terms of capacity reservation. As higher the probability is the more bandwidth reservation. In turn, nodes are normalizing the corresponding values to capacity.

The implementation of split-based routing scenario proved that pheromone table entries do not only represent the cost effective path (best route) or a likelihood of a node being congested, but furthermore relative merits of stability of nodes, and the reflection of delay sensitive packets. Split-based scenario has shown that it can successfully perform path marking (avoiding routing-path lock) and based on the service provided, it successfully reserves bandwidth.

A great feature of SART method is the significant optimization on sharing network resources to video streams, other data and/or e-mail packets. SART can mark the proper path and reserve the required capacity for transferring the data packets with priority, based on the delay sensitive coefficient of each packet. It also faces the fact that real time distributed services (voice and/or video traffic) must compete with e-mail traffic, database applications and file transfers, which affects the QoS. Much investigation of the principles (including new metrics) of split agent-based routing algorithm remains to be done. A hot spot for the network is the scalability and robustness that a telecommunication network must have in order to adapt in frequent network changes (traffic, grade of service, topology etc). Such networks must be tunable, enabling the adaptation to different QoS levels depending on the data traffic at a time, as well as characterized by controllable resource overhead.

Here specifically we propose a solution for adaptivity in heterogeneous systems where network resources have to be equally shared to users. This hybrid technique enable the coordinated service management which affects the desired user-level quality of service particularly for delay sensitive services such as video on demand, web based multimedia services (interactive music) and interactive network game playing. Network games must satisfy some requirements that involve the network connectivity and reliability of the service provided to end-users. A strict scenario that in our future work can be reached and specified, is in real time interactive networking games, where a cluster of different geographically located users can play against another cluster. This scenario can be extracted as an example of the geographic distribution of network resources. Roughly speaking by ensuring the QoS to end users different network applications would be reliable and

fully functional offering adequate performance which results to a satisfied customer-base.

This research could be extended for network scalability examination using hybrid-agent based schemes and combined with a Mobile Ad-hoc Network (MANET) configuration where no infrastructure exists. Therefore the next series in our research is the implementation of an agent-based algorithm applied to a *broadband* MANET with agents having special characteristics. These characteristics will position the agent's behavior. In turn agent's actions could be specified by the "indexing" of data packets. Thus agents could be entirely responsible for adapting -on demand- the proper resource allocation to the type of service (i.e. voice, multimedia and other delay sensitive services). Furthermore a scope of interest would be the exploration of different ways in the manipulation of the pheromone concentration on the edges as well as ability of agents to extract as more information as needed for MANETs characteristics i.e. mobility of node, energy consumption, motion frequency-velocity, etc. These characteristics influence the performance behavior of the agent-based algorithm and the overall throughput of the system.

5. Acknowledgements

Constandinos X. Mavromoustakis is partially funded by the A.G. Leventis Foundation.

6. References

- [1] A.S. Tanenbaum, Computer Networks, 3rd edition, Prentice-Hall, (1996).
- [2] J. Walrand, Communication Networks: A First Course, 2nd edition, McGraw Hill, (1998).
- [3] D. Estrin et al, Connecting the Physical World with Pervasive Networks, IEEE Pervasive Computing: Mobile and Ubiquitous Systems, Volume 1, Number 1, pp.59-69, 2002.
- [4] D. Bertsekas R. Gallager Data Networks, Prentice Hall, Inc, Upper Saddle River, New Jersey, 1992.
- [5] S. Appleby, S. Stewart, Mobile software agents for control in telecommunications networks, BT Technology Journal, vol. 12, 1994, pp. 104-113.
- [6] M. Dorigo, V. Maniezzo, A. Coloni, The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, vol. 26, No. 1, 1996, pp. 29-41.
- [7] Kaixin Xu, M. Gerla, A Heterogeneous Routing Protocol Based on a New Stable Clustering Scheme, IEEE MILCOM 2002, Anaheim, CA, Oct. 2002.
- [8] R. Schoonderwoerd, O. Holland, J. Bruinen, Ant-like Agents for Load Balancing in Telecommunications Networks, Proceedings of Agents'97, Marina del Rey, CA, 1997, pp. 209-216.

- [9] R. Schoonderwoerd, O. Holland, J. Bruten, L. Rothkrantz, Ant-based Load Balancing in Telecommunications Networks, *Adaptive Behavior*, vol. 5, 1997, pp. 169-207.
- [10] J. Kulik et al, Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks, *Wireless Networks*, vol. 8, no. 2-3, pp.169-185, March-May 2002.
- [11] L. Franck, G. Maral, Candidate algorithms for routing in a network of intersatellite links, AIAA-American Institute of Aeronautics and Astronautics.
- [12] G. Pei, M. Gerla, and T.-W. Chen, Fisheye State Routing in Mobile Ad Hoc Networks, *Proceedings of Workshop on Wireless Networks and Mobile Computing*, Taipei, Taiwan, Apr. 2000.
- [13] S. McCanne and S. Floyd, NS network simulator version 2.1b8, June 2001, <http://www-mash.cs.berkeley.edu/ns/>.
- [14] Michael Lawlor, Tony White, A Self Organizing Social Insect Model for Dynamic Frequency Allocation in Cellular Telephone Networks, AAMAS Conference '02, Month 1-2, 2002, Melbourne, Australia.
- [15] N. Minar, K. H. Kramer, P. Maes, Cooperative mobile agents for dynamic network routing, MIT Media Lab, E15-305 20 Ames St, Report, Cambridge MA 02139 USA.
- [16] M. Dorigo, L. M. Gambardella, Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, (1997) 1:53-66.
- [17] Kermarrec, L. Massoulié, A. J. Ganesh, Probabilistic Reliable Dissemination in Large-Scale Systems, *IEEE Transactions on Parallel and Distributed Systems*, Vol. 14, No. 3, March 2003.
- [18] P. Arabshahi, A. Gray, I. Kassabalidis, A. Das, S. Narayanan, M. El-Sharkawi, R. J. Marks II, Adaptive Routing in Wireless Communication Networks using Swarm Intelligence.
- [19] C. X. Mavromoustakis, H. D. Karatza, On the extensibility properties and performance measures of circuit switched telecommunication networks, using agent-based distributed routing algorithm, *Proceedings of 2003 International Symposium on Performance Evaluation of Computer and Telecommunication Systems-SPECTS*, Montreal, Canada, July 20 - 24, 2003, pp. 240-247.
- [20] C. X. Mavromoustakis, H. D. Karatza, "Split agent-based routing in interconnected networks". To appear in *International Journal of Communication Systems (IJCS)*, Wiley (accepted for publication).
- [21] C. X. Mavromoustakis, H. D. Karatza, "Agent-based throughput response in presence of node and/or link failure (on demand) for circuit switched telecommunication networks". *Computer Communications*, Elsevier, 2004, Volume 27, Issue 3, pp 230-238.
- [22] H. G. Sandalidis, C. X. Mavromoustakis, P. Stavroulakis, Ant based probabilistic routing with pheromone and antipheromone mechanisms. *International Journal of Communication Systems (IJCS)*, Wiley, January 2004, 17, p.55-62.
- [23] H.G. Sandalidis, K. Mavromoustakis, P. Stavroulakis, Performance Measures of an Ant based Decentralized Routing Scheme for Circuit Switching Communication Networks, Special Issue On Computational Intelligence In Telecommunications Networks, *Softcomputing Journal* 5, Springer Verlag, (2001) 4, 313-317.
- [24] H. D. Karatza, R.H. Hilzer, A Simulation-Based Performance Analysis of Epoch Load Sharing in Distributed Systems, *Simulation: Transactions of the Society for Modeling and Simulation International*, 2002, Vol. 78(7), pp. 461-471.
- [25] H.D. Karatza, Task Scheduling Performance in Distributed Systems with Time Varying Workload, *Neural, Parallel & Scientific Computations*, Dynamic Publishers, Atlanta, 10, 2002, pp. 325-338.