Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Beaver Algorithm for Network Security and Optimization: Preliminary Report

Aladdin Ayesh

Abstract— **This paper presents the theoretical account for designing and developing an algorithm for network security inspired by the behavior of Beavers. The algorithm uses the beaver behavioral patterns in constructing dams and water tunnels to create analogous secure tunnels and information lakes. The approach is a user-centric and the paper demonstrates the use of the algorithm in security and route optimization with the assumption that the beaver agent is deployed on a mobile device (e.g. smart mobile phone). An algorithmic approach to design the beaver agent and swarm is followed here. The set of algorithms presented is complemented by critical review outlining the further work needed.**

I. INTRODUCTION

Beavers spend most of their times building dams and water tunnels to secure themselves from predators and protect their winter food supplies. This behavior is the inspiration of an algorithm for information security in ad hoc networks presented in this paper. The approach aims to be a user-centric where it is to be deployed on mobile devices such as smart mobile phones. Once the system is deployed the beaver agent will start gathering information about its environment and connections building an analogous dam. In the process, it can gather information about routes and connections to establish a trust table including where to maintain it larder of information. This larder of information may be shared in parts with other beavers in similar fashion to animals through foot stamping for example for tagging routes. Other information may be shared through warehousing.

In this paper, an algorithmic design of the proposed algorithm is presented at both levels of a single beaver agent and a swarm of beavers. The algorithms developed with a mobile device deployment environment in mind, however, they can easily be ported for other networking applications. The paper starts with a brief background followed by algorithmic design for a single beaver and swarm of beavers. It ends with a critical review outlining the further developments to follow.

II. MOTIVATION AND BACKGROUND

In recent years, the advances in swarm intelligence and ad hoc networks overlapped at number of points. Network optimization is particularly worth mentioning as the most fruitful of this marriage of swarm intelligence algorithms and ad hoc networks. This section covers some elementary information on ad hoc networks and swarm intelligence [1], [2], [3].

Networks security is a subject of continuous development. Ad hoc networks just add extra factors in the equation of security. The lack of fixed infrastructure and predetermined nodes means identifying intruding nodes and securing data exchanged while maintaining the dynamic infrastructure becomes a difficult task. In ad hoc networks, two main issues are necessary for a node to identify: network selection [4] and route selection [5].

Swarm intelligence is a nature inspired computational paradigm. Number of physical and biological systems inspired number of swarm intelligence algorithms. In addition, social animals and insects, e.g. ants and bees, inspired some of the most successful algorithms in this area [6], [7], [8], [9], [10], [11] to mention but few examples.

Swarm intelligence is based on the interaction of the system entities. This resembles networks where nodes communicate with each other. Furthermore, new networking protocols are designed with collaborative nodes in mind with a reduced or relaxed infrastructure. This led to a number of attempts to use swarm intelligence algorithms over networks, especially sensor networks, and in managing information and event, which is applicable to number of networking tasks, to mention but few intersection points between swarm intelligence and networks [12], [2], [3], [13], [14], [15].

This paper presents the theoretical development of swarm intelligence algorithmic approach based on behavior of beavers for network security and optimization. The approach utilizes some features of agents technologies such as threads and behavior based architectures.

Beavers are water animals that are known for their dams building abilities and water routes construction. They are often found in mountains that are water rich areas but food is often scarce over winter periods. As a result, each beaver aims to build a larder to store food for winter. The aim of the beaver is to establish its own dam to protect itself and its larder. Also, it establishes water routes to escape and to forge for food. There is clear analogy one can build on the beaver behavior with networks especially in the context of mobile devices (e.g. smart phones). If one aims to develop a user-centric approach to security and communication, a mobile device may be seen as a dam. The route establishment is network routes and connections possible from the given device. The searching for data and communication can be seen as forging for food, which then can be stored in private, e.g. address book, or semi-shared warehouses, e.g. restricted sharing of links depository or a calendar, similar to the beavers larder. One has to assume the storage of such information will be on external devices, and

Dr. Ayesh heads the Intelligent Mobile Robots and Creative Computing Research Group, Department of Informatics, Faculty of Technology, De Montfort University, Leicester LE1 9BH. Email: aayesh@dmu.ac.uk

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

not on the mobile phone, within a grid or cloud computing infrastructure. Social networking sites, which are available on mobile phones already with private and shared information, can be cited as a close example to be managed by the beaver agent.

Swarm intelligence paradigm often deploys agent technologies in realizing the swarms. Agent technology use over networks have been growing [16], [17] which makes it a good means of developing the beaver algorithm proposed here.

III. BEAVER BEHAVIORS BASED APPROACH

A. Beaver Algorithm

A single beaver can build a dump and routes so it can protect itself and moves between lakes. Part of this is also collecting and maintaining winter food supplies which one can think of it as information. A beaver can be seen as a mobile agent that collects and aggregate information that are necessary for two objectives: environment preservation and winter food preservation. Let us define a beaver agent (BA) is a mobile agent on a mobile device then one can define the mobile device to be the lake to be maintained and winter food can be an analogy for information collected and maintained. The mobility of the agent is a requirement for the fact that while the mobile device forms the core part of the dam, the beaver's larder will be a storage space or a series of storage spaces within a grid or cloud computing infrastructure. The practicality of the likely implementation and deployment of the algorithms presented in this paper requires the agent to be mobile.

Information collection and maintenance require the ability to interact with similar or opposing entities. This will be discussed in the following section. In this section the focus is given to a single beaver agent building its wall and routes. There are three sets of behavioral rules:

- Building rules: these rules enable the beaver to build and maintain its lake to secure itself and its winter food.
- Routing rules
- environment-preservation rules

In addition there are the standard agent rules such as self-preservation, physical and other rules.

An overall algorithm encompassing the Beaver behavioral rules can be presented in a form of a subsumption architecture [18], [19], [20], [21]. The original architecture is slightly altered to accommodate coupled behaviors. Thus we define separate threads for running and levels of priority. When a behavior is subsuming another behavior they the two will share the same thread. An overall summary algorithm can be presented as follows:

Figure 1, please see appendix, shows the system overview presented in algorithm 1 in a diagrammatical form. The different threads and levels are shown. The architecture does not conform to subsumption nor to motor scheme architectures instead it borrows elements from both.

The following sections will look at the different components of the beaver algorithm in relation to a single beaver agent system (section III-B) and multi-agents or a beaver swarm system (section III-C).

Algorithm 1 Beaver Algorithm - Main Thread

```
Initialize region and environment
while $(true)$ do
    Create and maintain dam over thread 1 - low priority
    Regular clean dam over thread 2 - Level 2 priority
    Route finding over thread 2 - Level 3 priority subsumes Level 2 activities
    Route Optimization over thread 3 - Level 3 priority coupled with route finding
    No residence over thread 1 - alert priority
    Sharing warehousing (inc. footstamping) thread 3 - Level 3 priority subsumes route optimization
    Intruders Alert and Community Watch: collaborative information exchange and data fusion - [This step have not been developed within this work, instead it presents extensions to follow.]
end while
```

B. Single Beaver Agent - Routing and Optimization

A mobile device may have a single agent that maintain its workings.

Algorithm 2 Dam construction Behavior - Over Thread 1 (Low Priority)

```
while $(true)$ do
    Search device
    if device has active communication protocol then
        check protocol profile;
        if protocol profile exists then
            observe and update;
        else
            create a new profile;
        end if
    end if
end while
```

Algorithm 2 runs at the lowest level of the architecture of a beaver as it transcends into the default behavior. If the beaver is threatened or information is lost, a beaver has to re-construct its dam or repairs it.

When a protocol or a communication port on a device found to have more than one profile and yet they are different, this can be defined as a potential security breach. However, the idea behind using intelligent agents is to enable an interactive response in dealing with changes happening to, and in the behavior of, communication protocols and ports especially on devices with multiple communication ports. This part of algorithm 3 will need further development. Algorithm 4 shows the process of profiling routes and nodes within network routes. The aim is always to use known route that is evaluated to be safe. Changes in nodes or communication can be encountered and thus appropriate actions in examining these routes are necessary. The approach here is

Algorithm 3 Dam Cleaning - Over Thread 2 (Level 2 Priority)

```
while $(true)$ do
    if Number of profiles exceeds the number of communication protocols-devices available then
        compare profiles for duplication;
        search log for newly created profiles over same protocol-device
    end if
    if Duplicate found then
        isolate communication protocol-device for observation-repair
        archive profiles and start a new profile
    end if
    if protocol-device has multiple variation profiles then
        disable the device
        report misbehavior to user or other beavers
        further actions requires
    end if
end while
```

to imitate beavers in curious but cautious approaching to new connections and routes.

C. Beaver Swarm - Exchange of Information

Once a beaver identified the tunnels through which it operates and tagged them with its security assessment these tunnels and other beavers can use the established tagging for their own evaluation of routes. This is the minimum information a beaver can share with other beavers and it does not require specific authentication beyond the initial authentication of beavers to operate on the services provided by the network. This tagging we shall call footstamping and it be maintained in a general service depository by the network management processes.

Beavers are emigrating animals, however, once a beaver settles in an area it starts to define its region and the related connected regions. This fact has been already used in route finding and establishing a table of trusted connections and nodes. Similarly, region definition can contribute to more secure devices with high quality services. This localized views is exploited in the way information is shared and co-operation with other beavers and active services is initiated. The information sharing has two rules:

- no residence rule to prevent any other beaver from residing in the same dam (i.e. device) and
- warehousing share within the region.

Rule 2 creates an organizational structure similar to acquaintance networks in multi-agents system organization. Algorithm 6 implements the first rule.

Algorithm 7 implements the rule of warehouse sharing but also implements the region definition concept. The region definition is updated depending on the changes of the mobile device location and registration.

The collaborative and community sharing part of the beaver algorithm still needs further development. There are

Algorithm 4 Route Finding - Over Thread 2 (Level 3 Priority)

```
while $(true)$ do
    observe connections C over all routes R
    $\forall c \in C \& \exists r \in R \& cispossibleoverr$ check:
    CASE 1: c was visited before
    observe, evaluate and leave footstamp
    if $\exists n \in N$, where n is a new node in the routing then
        evaluate n and leave footstamp
        find alternative known node then route or initiate limited services to examine n.
        if new n proven secure then
            add new r for c
        end if
    end if
    END CASE 1
    CASE 2: c is not visited before
    observe, evaluate and leave footstamp
    if $\forall n \in N$, in the routing known or secure then
        evaluate n and leave footstamp
        find alternative known connection or initiate limited services to examine c.
        if proven to be secure then
            add new c and associated routes
        end if
    end if
    END CASE 2
    CASE 3: c and r are not visited before
    quarantine
    examine extensively over time and evaluate
    leave footstamps; examine footstamps
    END CASE 3
end while
```

Algorithm 5 Footstamping Behavior - Over Thread 3 (Level 3 Priority)

```
while $(true)$ do
    Search device
    if device has active communication protocol then
        check protocol profile;
        if protocol profile exists then
            observe and update;
        else
            create a new profile;
        end if
    end if
    forall (c,r) visited
    search protocol profile
    if (c,r) exists then
        update entry;
    else
        assert a new entry.
    end if
    end forall
end while
```

Algorithm 6 No Residence Behavior - Over Thread 1 (Alert Level Priority)

```
while $(true)$ do
    Search device for active processes
    if $\exists p$ where p is a process & p is not owned by self then
        quarantine p;
        record $< a, o, c, r >_p$ where a is set of activities, o is owner, c and r are connection and route used by p respectively;
        terminate p;
    end if
    if p persist or clone then
        close $< c, r >_p$;
        report; recreate dam;
    end if
end while
```

Algorithm 7 Warehouse Sharing Behavior - Over Thread 3 (Level 3 Priority - subsumes route optimization)

```
while $(true)$ do
    define region - cell of home operation
    if request-received; region-beaver then
        friendly beaver access level granted to sharing warehouse;
    else
        if out-region-beaver or not-defined-agent then
            limited access granted
            monitoring access for abuse
            report any unusual access and terminate if necessary
        end if
    end if
    if device in roaming mode then
        define a temporary region
        rebuild dam($t_i t_j$)
    end if
    if device reinitiated (e.g. new sim card) or re-established then
        quarantine old dam;
        build new dam;
    end if
end while
```

number of issues related to security, identification, level of collaboration and others that should be considered in developing a beaver community over a network.

IV. Critical Review

A beaver inspired approach was presented in this paper. Algorithmic means were used in developing this approach resulting in a set of algorithms. The approach is a user-centric. It assumes a beaver agent will be deployed on a single networkable device that is likely to be a smart phone. The approach demonstrates the possibility of using agent technologies and swarm intelligence paradigm to provide interactive and intelligent means to network security and optimization. However, there are a number of critical points to be highlighted:

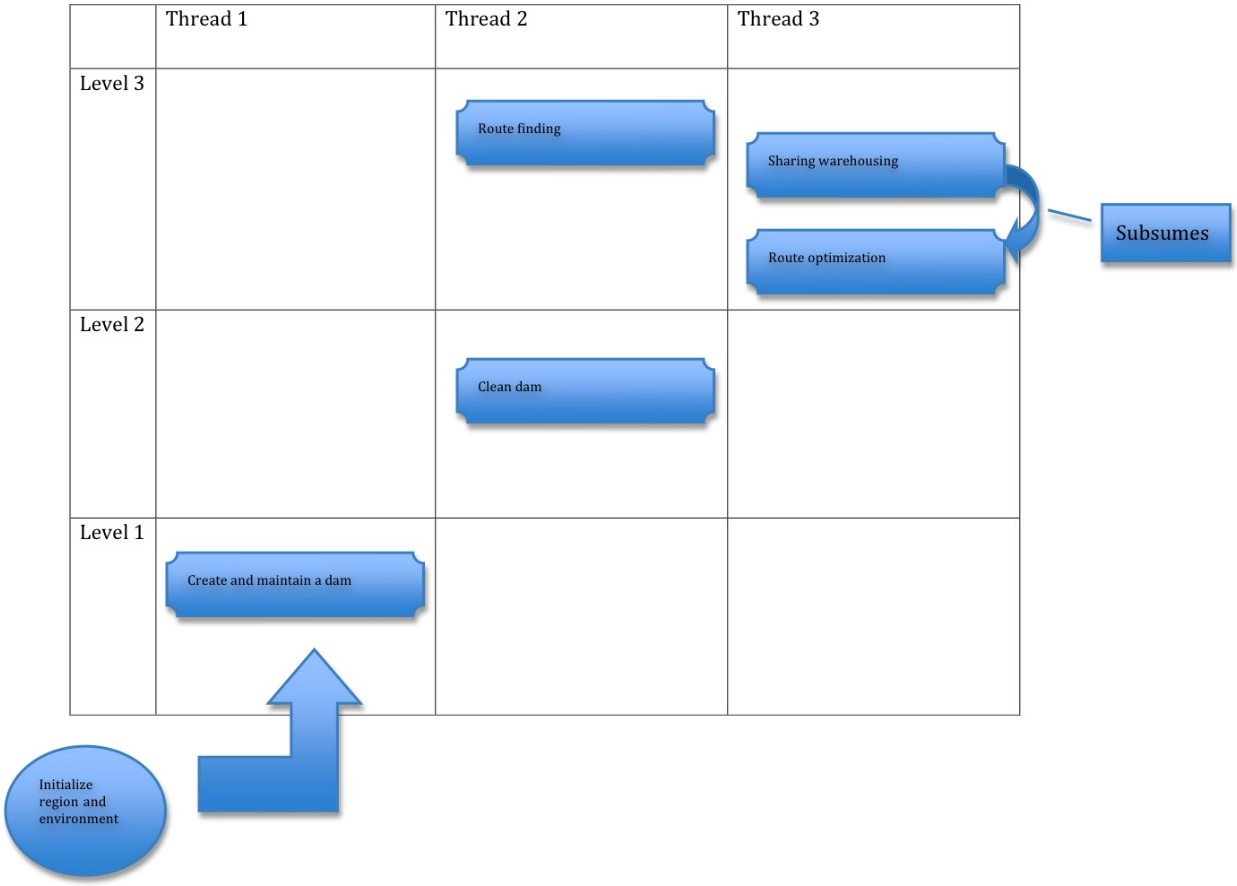
- First and upmost is the processing and energy consumption required. If this system is running on a mobile device this is a crucial point to be considered. The development in active processes management will help to reduce the consumption but further research in intelligent energy saving through patterns of sleep is required.
- The amount of data to be saved and retrieved can easily go out of control. The approach assumes large over writing with short interest-concentration span. However, active memory management and processes should be developed explicitly(e.g. [22]).
- Implementation details are not complete. A real life implementation over network protocols has its own problems that may not appear in simulations.

Overall the approach proposed is a plausible approach and contributes to the increasing interest in the use of agents and swarms over networks. An implementation of the system using Jade-LEAP is in progress, which may highlight interesting results. A simulation using Jade can then be ported to real mobile devices where real life experimentations can be achieved.

V. Conclusion

This paper presents a new algorithm for mobile network management inspired by beavers. Beavers are water animals that create dams and water tunnels to secure themselves and their winter food storage. An analogy between the beavers creation and networks can be found in the form of routing and information management. This analogy is exploited in a new algorithm to manage mobile devices over networks. The algorithm presented is a end-node centric and thus enables for a user-centric mobile device management agent development. The paper demonstrates the proposed approach with detailed algorithmic description. The paper finishes with a critical review of the approach proposed and future developments to follow.

Appendix



	Thread 1	Thread 2	Thread 3
Level 3		Route finding	Sharing warehousing Route optimization
Level 2		Clean dam	
Level 1	Create and maintain a dam		

Fig. 1. Diagrammatical representation of the Beaver Algorithm - System Overview

References

[1] M. Al-Obaidy and A. Ayesh, "Optimizing autonomous mobile sensors network using pso algorithms," in *Proc. International Conference on Computer Engineering & Systems ICCES 2008*, 25–27 Nov. 2008, pp. 199–203.

[2] J. A. Krill, D. A. Day, M. J. O'Driscoll, and K. W. O'Haver, "Swarming network for intruder detection," in *Proc. 3rd International Conference on Intelligent Sensors, Sensor Networks and Information ISSNIP 2007*, 2007, pp. 341–346.

[3] Y. Lu, G. Zhao, and F. Su, "Adaptive ant-based dynamic routing algorithm," in *Proc. Fifth World Congress on Intelligent Control and Automation WCICA 2004*, vol. 3, 2004, pp. 2694–2697 Vol.3.

[4] M. M. Alkhwlani and A. Ayesh, "Access network selection using combined fuzzy control and mcdm in heterogeneous networks," in *Proc. International Conference on Computer Engineering & Systems ICCES '07*, 2007, pp. 108–113.

[5] A. Zahary and A. Ayesh, "Analytical study to detect threshold number of efficient routes in multipath aodv extensions," in *Proc. International Conference on Computer Engineering & Systems ICCES '07*, 2007, pp. 95–100.

[6] H. A. A. Bahamish, R. Abdullah, and R. A. Salam, "Protein conformational search using bees algorithm," in *Proc. Second Asia International Conference on Modeling & Simulation AICMS 08*, 2008, pp. 911–916.

[7] X. Bi and G. Luo, "The improvement of ant colony algorithm based on the inver-over operator," in *Proc. International Conference on Mechatronics and Automation ICMA 2007*, 2007, pp. 2383–2387.

[8] D. T. Pham, M. Castellani, and A. A. Fahmy, "Learning the inverse kinematics of a robot manipulator using the bees algorithm," in *Proc. 6th IEEE International Conference on Industrial Informatics INDIN 2008*, 2008, pp. 493–498.

[9] D. T. Pham, S. Otri, A. Ghanbarzadeh, and E. Koc, "Application of the bees algorithm to the training of learning vector quantisation networks for control chart pattern recognition," in *Proc. 2nd Information and Communication Technologies ICTTA '06*, vol. 1, 2006, pp. 1624–1629.

[10] G. Tan, Y. Liu, H. Yao, J. Li, and N. Han, "A hierarchical routing model for large scale networks based on ant algorithm," in *Proc. International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies ICN/ICONS/MCL 2006*, 2006, pp. 88–88.

[11] N. Xiong, J. He, J. H. Park, T.-H. Kim, and Y. He, "Decentralized flocking algorithms for a swarm of mobile robots: Problem, current research and future directions," in *Proc. 6th IEEE Consumer Communications and Networking Conference CCNC 2009*, 2009, pp. 1–6.

[12] A. Arami, B. R. Rofoee, and C. Lucas, "Multiple heterogeneous ant colonies with information exchange," in *Proc. (IEEE World Congress on Computational Intelligence). IEEE Congress on Evolutionary Computation CEC 2008*, 2008, pp. 3298–3304.

[13] E. M. Saad, M. H. Awadalla, and R. R. Darwish, "A data gathering algorithm for a mobile sink in large-scale sensor networks," in *Proc. Fourth International Conference on Wireless and Mobile Communications ICWMC '08*, 2008, pp. 207–213.

[14] M. Yuan, S. Wang, and P. Li, "A model of ant colony and immune network and its application in path planning," in *Proc. 3rd IEEE Conference on Industrial Electronics and Applications ICIEA 2008*, 2008, pp. 102–107.

[15] W. Zhengjia, Z. Liping, W. Ying, and W. Kui, "Optimization for multiresource allocation and leveling based on a self-adaptive ant colony algorithm," in *Proc. International Conference on Computational Intelligence and Security CIS '08*, vol. 1, 2008, pp. 47–51.

[16] S. AlZahrani, A. Ayesh, and H. Zedan, "Regionally distributed architecture for dynamic e-learning environment (rdadele)," in *Proc. Conference on Human System Interactions*, 2008, pp. 579–584.

[17] ——, "Multi-agent system based regional data grid," in *Proc. International Conference on Computer Engineering & Systems ICCES 2008*, 25–27 Nov. 2008, pp. 337–342.

[18] L. Bourgois, A. Delteil, and F. Levy, "Web services subsumption with a specific pdl," in *Proc. International Conference on Internet and Web Applications and Services/Advanced International Conference on Telecommunications AICT-ICIW '06*, 19–25 Feb. 2006, pp. 158–158.

[19] S. Nagano, T. Hasegawa, A. Ohsuga, and S. Honiden, "Dynamic invocation model of web services using subsumption relations," in *Proc. IEEE International Conference on Web Services*, 6–9 July 2004, pp. 150–156.

[20] H. Nakashima and I. Noda, "Dynamic subsumption architecture for programming intelligent agents," in *Proc. International Conference on Multi Agent Systems*, 3–7 July 1998, pp. 190–197.

[21] R. Brooks, "A robust layered control system for a mobile robot," vol. 2, no. 1, pp. 14–23, 1986.

[22] A. Ayesh, "Towards memorizing by adjectives," in *AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems.*, 2001.