Situated Service Oriented Messaging for Opportunistic Networks

Juhani Latvakoski, Tomi Hautakoski, and Antti Iivari

VTT Technical Research Centre of Finland,
Kaitoväylä 1, P.O.Box 1100, FIN-90571 Oulu, Finland
{juhani.latvakoski,tomi.hautakoski,antti.iivari}@vtt.fi

Abstract. A novel concept for situated service oriented messaging applicable in the context of biologically inspired opportunistic networks has been provided in this paper. The solution utilizes different contextual information sources to create and update a view of the communicational situation. Smart diffusion of relevant control data between neighbouring nodes using novel swarm intelligence based method enables spreading of information only to the interested nodes without unnecessarily disturbing the non interested nodes. The evaluations done against epidemic routing protocol indicate that the proposed solution lowers the amount of transmissions in the network, thus reducing precious resource usage in the nodes. This is achieved without introducing further delays or deteriorations in the message delivery ratio.

Keywords: opportunistic communication, context awareness, service awareness.

1 Introduction

The number of wireless communicating embedded devices has continuously been increasing in recent years. Because of the inherent nature of such devices is to be both mobile and dynamic, it is obvious that the destination of communication is not necessarily reachable at the time of communication need. This type of challenge has also been described previously in the context of InterPlaNetary networks (IPNs), Delay-Tolerant Networks (DTN) [1, 2, 3] and opportunistic networking [4]. A common essential feature for them is that the source and destination may never be connected to the same network at the same moment of time, but communication may be enabled on a hop-by-hop basis. In such a case, finding a route by means of Mobile Ad hoc Networks (MANETs) such as e.g. Ad hoc On-Demand Distance Vector (AODV) is not possible. To solve the problems, several proposals have been provided such as combination of DTN and MANET routing [5]. The problem of the referred *disconnected communication* is still rather open research item and it has been one starting point problem for this research.

The opportunistic routing can be categorized to e.g. dissemination based or context based routing [4, 6]. Dissemination based routing techniques aim to deliver messages to the destination by simply diffusing them all over the network. Usually the

E. Altman et al. (Eds.): Bionetics 2009, LNICST 39, pp. 50–64, 2010.

dissemination methods work by offering the messages to neighbor nodes, when they are in the radio coverage. The offering can consist of sending the full message data to the neighbors, who then apply various filtering techniques to lower the network load. Another approach is to send advertisements of the data available at the sender, and receiver can request for the data based on the advertisements. Finally the sender will respond with the actual data message. Examples of dissemination based routing techniques are Epidemic routing [7], Meeting and Visits protocol [8] and Network coding based routing protocol [9]. Dissemination based approach work quite well when contact opportunities are very common. However, the problem with the dissemination based routing may be the heavy load generated into the network, which may cause network congestion resource over-usage situations. The network traffic can be lowered by limiting allowed hops or number of copies of the messages. The context based routing applies information about the contextual situation to achieve more efficient routing. Examples of context based routing techniques are Context-Aware routing (CAR) [10] and MobySpace routing [11]. The context based approach can reduce the network load compared with dissemination based approaches. However, the reasoning of the next hop increases the needed amount of CPU and memory resources from the nodes.

As a contribution, the situated service oriented messaging concept, and the algorithms applicable in the context of biologically inspired networks has been provided. Simulations are applied to evaluate the usefulness of the concept. The contribution essentially differs from the dissemination and context aware routing, because here both situation and service awareness are applied to optimize the message forwarding. The contribution extends the situated message forwarding concept [12] in the sense that here a new set of algorithms, and a novel swarm-based service-oriented approach for efficient communication inside a network island have been provided.

The rest of this paper is organized as follows. Chapter 2 describes the situated service oriented delivery concept and related algorithms. Chapter 3 describes the simulation based evaluations of the provided methods. Finally, conclusions are provided in chapter 4.

2 Situated Service Oriented Messaging

2.1 Concept

It is assumed in this research that the service and situation awareness in the message forwarding will decrease the amount of the load in the network by decreasing the amount of useless traffic while still keeping the reliability of messaging high. In addition, it is assumed that the increase of self-organization capabilities in the system will enable better scalability. In our approach, the neighbourhood information is applied to increase situation awareness, and also service information is applied when deciding the delivery scope. The decision is made locally in each individual node according to principles of self-organization, and in such a way to enable better system scalability.

From the service and content point of view, the general requirement set for the spreading of information is that all the nodes, who are interested about the information, eventually receive the information content. For example, in the targeted advertising scenario, only a group of nodes should receive the information content, and the others are not interested about it at all. To save network resources the spreading of information should not be blind, but it should be *service aware* instead e.g. by enabling smart diffusion of relevant control data between neighbouring nodes. There should be no need to disturb the non interested nodes by delivering them information which provides little or no value to them. On the other hand, even non-interested nodes may act as carrier of information for a specific group.

The conceptual mode for the service and situated opportunistic communication is visualized in figure 1. The model consists of User nodes (U), tiny nodes (T) and access point (AP) nodes, according to the characteristics of the nodes. The T nodes are assumed to be small, limited capability nodes which cannot act as message forwarding nodes. The U nodes act as forwarding nodes, and the role of AP nodes is to route traffic from the opportunistic networks towards more static networks such as e.g. Internet. The referred nodes may belong to network islands, e.g. Bionets A, according to their communication ranges. There are three network islands visualized in figure 1.

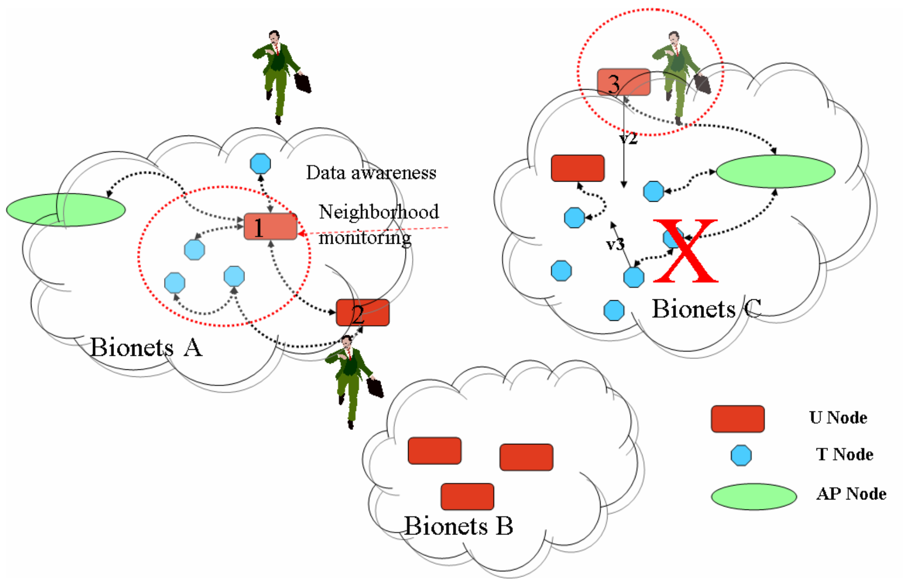


Fig. 1. Situated Service oriented opportunistic communication

The key features of the provided situated service oriented opportunistic communication concept are the following:

- Each U node monitors its neighborhood in order to collect real-time information about the situation in its' environment (neighborhood monitoring)
- The communication level in each U node receives information on the service content to enable smart data based message forwarding (data awareness)

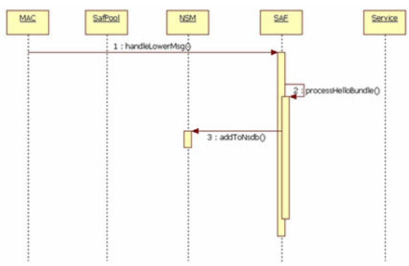
- When deciding what to do for the incoming message, each U node operates according to the locally executed algorithm, which is used to decide whether the message should be forwarded or stored into the local memory.
- The solution consists of two components: Situated Adaptive Forwarding (SAF), and Service Oriented Forwarding (SOF).
- The algorithms are used as local reasoning engines for message forwarding to enable scalable opportunistic communication.

2.2 Situated Adaptive Forwarding (SAF)

SAF is based on monitoring statuses of nodes and their neighborhood, connectivity of nodes, their resource situation (CPU, message storage space) and classification of messages into classes. The solution consist of neighborhood discovery and message sending & message forwarding algorithms, each of which are described in the following.

2.2.1 Neighborhood Discovery Procedure

We begin by introducing the how neighboring nodes are found, and what information is exchanged among them. Every node has a Network Situation Database (NSDB) which contains context information relevant to forwarding of messages. An example entry of this database is shown in figure 2.



Field	Description
Node	Node's unique ID
Delete Time	Scheduled removal time for this entry
Connectivity	Connectivity of a node
Stability	Connection stability to a node
Stability duration	Duration of a stable connection to a node
Connectivity change rate	Rate of change in a node's connectivity
Current status	Current neighborhood status of a node
History status	Past neighborhood status of a node
Battery status	Battery status of a node
Storage status	Storage status of a node
CPU status	CPU status of a node
Role	Role of a node
Location	E.g. geographical location of a node

Fig. 2. A Network Situation Database (NSDB) entry and the NSDB refresh procedure

The discovery mechanism works by sending periodic HELLO messages. If a received HELLO came from an already known source node, its old entry in NSDB is refreshed with new values contained in the message (see figure 2).

When a new node is discovered (figure 3), a transfer is made if there are stored messages that should be sent also to the newly discovered node. Next, also those stored messages that are destined for any node in the newly met node's current network neighborhood, are forwarded. Finally, copies of such stored messages, which have Remaining Chance (RC) value bigger than 0.0 are sent to the discovered node. RC values are assigned for messages in the following way: If a message cannot be delivered immediately, because of the sender's current neighborhood or its' neighbors' total sum of chance for delivery doesn't reach the value assigned for the category of the message, it is also saved in the senders' message pool. Then, a RC value is given to a message based on its category, and delivery chances of those neighboring nodes that have gotten a copy of the message.

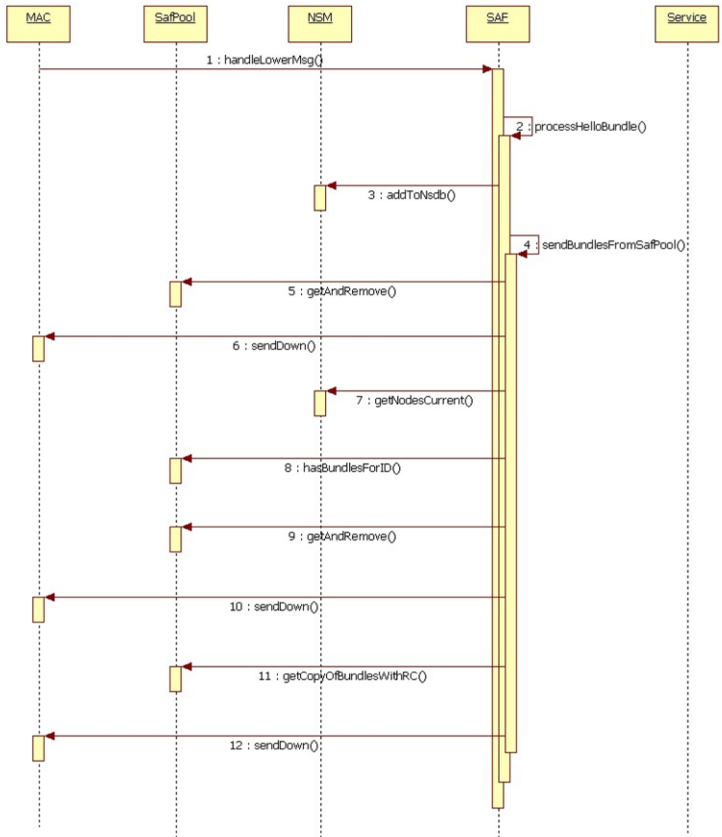


Fig. 3. A procedure when a new node is discovered

2.2.2 Message Sending and Forwarding Algorithms

To describe how the designed SAF model handles sending and reception of messages, sequence diagrams of different execution paths depending on the situation have been provided. Figures 4-6 provide explanations on how outgoing messages are handled. In figures 6-10, we show how incoming data messages are processed. The first three sequence diagrams all present the same principle case where the service layer needs to send a message to a certain node ID. At the service layer, a *sendBundle()* function is called which sends data using the following message structure: < | Type | Source | Destination | Sequence Number | Hop Count | Category | Role | Payload | >. The SAF module (at the network layer) receives the message and checks if the destination node ID is located in the vicinity of this node. In the case 1 (figure 4), this is expected to be true, and thus SAF creates a message representing a network layer message. This message is filled with headers and the service message as a payload. The headers include the following fields: < | Type | Source | Destination | Sequence Number | Purge Time | Category | Payload | >. The message is then sent to MAC module for transmission.

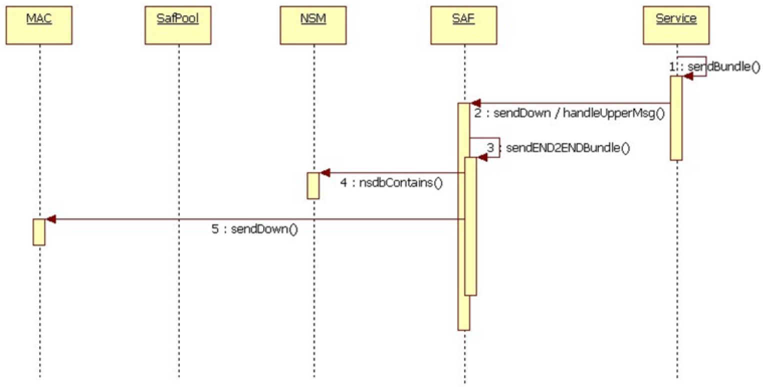


Fig. 4. Case 1, processing of an outgoing message

In case 2 (see figure 5), the destination ID is not a direct neighbor of the source node, and therefore it checks if the needed ID is within 2-hop range, which is true now and the message can be sent away.

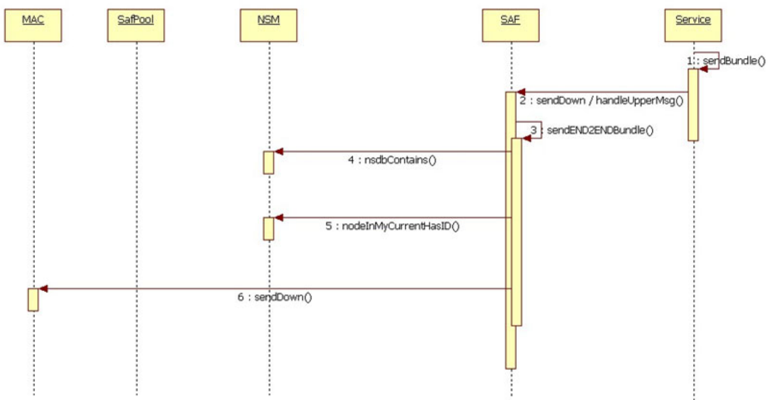


Fig. 5. Case 2, processing of an outgoing message

Case 3 (in figure 6) describes the most complex situation where the destination cannot be found within the current network neighborhood of the source node. Now dissemination of the message in the network in a best way possible given the situation at hand is started. A multi-keyed map consisting of key - value pairs is queried from the NSDB where each value is relative delivery probability and key indicates what neighboring node ID has that probability. That operation is followed by a iteration over the map to find out if the total sum of probability of our neighboring nodes exceeds required threshold for the processed message's category. Now, this is expected to be false, and more nodes are added to the map describing neighbors to which the

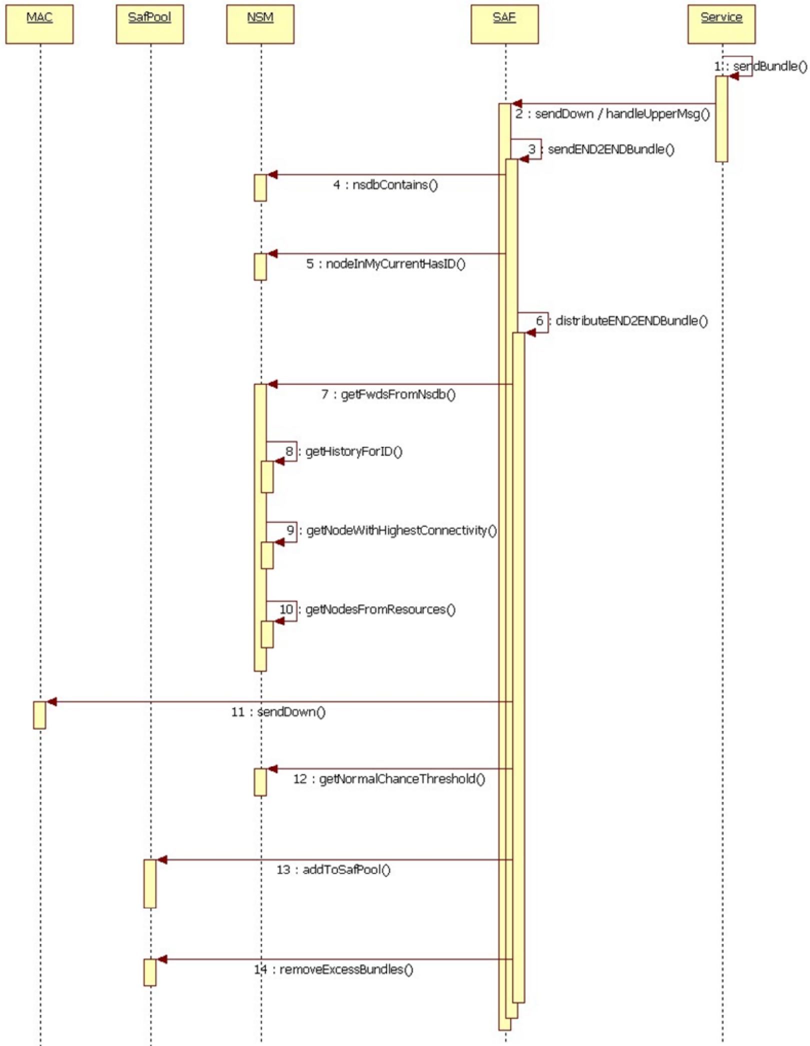


Fig. 6. Case 3, processing of an outgoing message

message is going to be sent. The delivery chance optionally increased by querying NSDB for the most connected node of the neighboring nodes as well as for nodes that have the most resources available, and adding those nodes to the map. Then the message is sent to all neighboring nodes listed in the map, and its RC value is updated as described in the earlier section.

In case 4 (figure 7) is where we begin to investigate how the SAF model handles incoming messages. First, messages are received from the MAC layer with *handleLowerMsg()* function, and in this case the incoming message was destined for this node; thus it is sent to service layer.

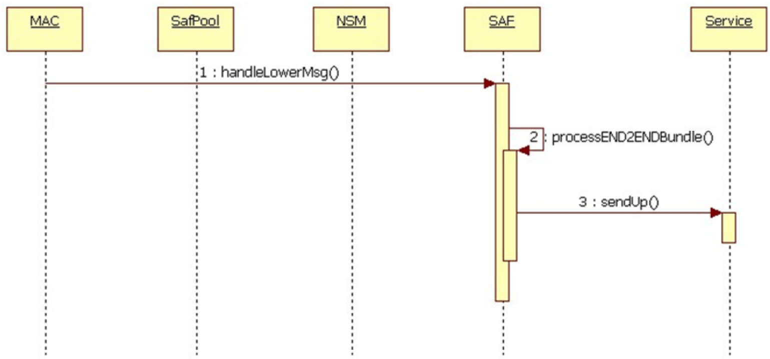


Fig. 7. Case 4, processing of an incoming message

Case 5 (see figure 8) describes a situation where the message was destined for some of the nodes that the receiver has in its neighborhood, and the message is forwarded without other procedures to the corresponding neighbor.

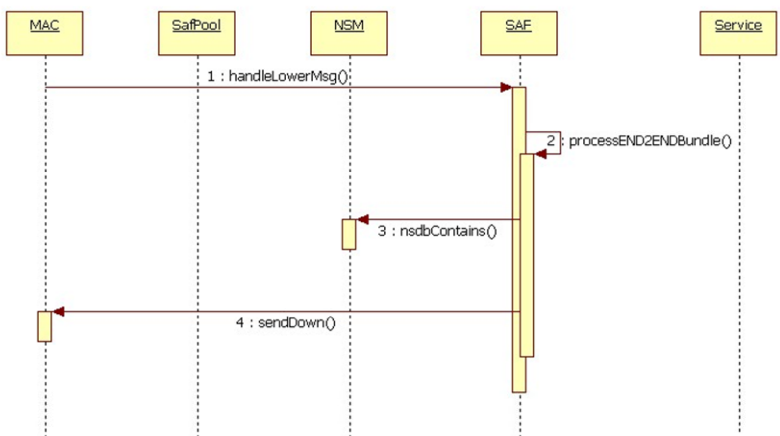


Fig. 8. Case 5, processing of an incoming message

We continue with case 6 (figure 9) where the first three steps are the same as before, but the destination ID of the message is searched within a 2-hop radius of the node's current neighborhood. A match is found and message is forwarded to a neighboring node which then forwards it again.

If none of the cases from 4 to 6 were applicable to an incoming message, case 7 (figure 10) includes a description of a sequence diagram that is used as a fail-safe option. The message is stored into the message pool of the sender. It is important to notice that in this case, the message is associated with a RC value of zero to limit too

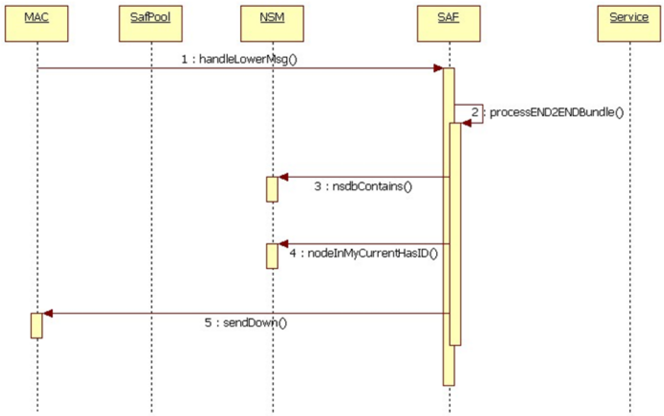


Fig. 9. Case 6, processing of an incoming message

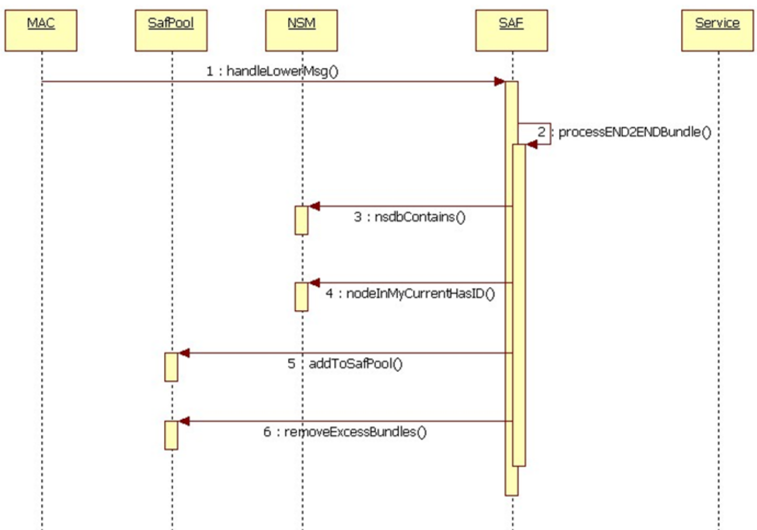


Fig. 10. Case 7, processing of an incoming message

resource exhausting dissemination. Finally, also the message pool is purged to have only valid messages and to keep it within given size limits.

2.3 Service Oriented Forwarding (SOF)

The service oriented message forwarding mechanism applies a swarm intelligence based approach, and its objective is to distribute messages inside a network island with minimal disturbance to non pertinent nodes in the network. In order to save network resources it is here proposed that the service oriented forwarding algorithm should not simply diffuse messages blindly to all the reachable nodes in the network. However, it ensures that the messages eventually reach a satisfactory amount of pertinent nodes even behind a zone of non-pertinent ones.

Since the situation will very likely change over time a smart and adaptive mechanism is required. The message diffusion algorithm needs to operate in a highly "distributed", self-organized manner and a single node will only need to follow a very simple set of rules. Swarm Intelligence (SI) based systems consist of (usually) unsophisticated agents interacting locally with each other and their environment that will eventually lead to the emergence of intelligent and coherent global behaviour. Taking into account the nature of the problem at hand and the characteristics of SI based systems discussed above, it is clear that SI will provide us an excellent basis on which it is possible to build algorithms that perform heavily distributed problem solving without centralized control or the utilization of a complicated global model. There are other communication network protocols based on SI, such as ANTNET. [13]. However, these algorithms mostly deal with problems associated with routing in standard IP networks and, as such, are not further discussed herein.

The Bio-inspired and SI based algorithm examined herein is inspired by the food foraging behaviour of a honey bee colony. This foraging process in a bee colony functions by deploying scout bees. These scout bees move randomly from one place to another in search of promising food sources. When the scout bees return to the hive, they communicate to the colony their findings. The information that they communicate contains, for example, the direction in which the food source is located and its fitness or "quality". Others have also used honey bees as inspiration for optimisation algorithms. For example, in [14] Pham *et al* describe a population-based search algorithm that mimics the food foraging behaviour of swarms of honey bees. By the application and slight adaptation of the principles discussed above, a highly adaptive decentralized networking technique is constructed that aims to efficiently disseminate messages to pertinent nodes. The source nodes will send scout messages at frequent intervals. Scout messages behave very much like scout bees in nature. The scout messages will randomly hop from node to node until they have reached a specific number of hops. They return to the originating node by going through the same path backwards. In addition to the node that originated the scout message, each node on the path of the scout message will extract and store information from the scout message. The nodes will store next hop and fitness values in a table relating to the service at hand. Over time, more and more of these scout messages will be sent and processed by the nodes in the network ensuring the emergence of applicable tables built using the information carried by scout messages at various nodes. Due to the high degree of randomness and the decentralized nature of the mechanism, the contents of these tables will keep adapting to reflect the changes in the network environment. In the beginning stage, a node initiates the process by generating and sending a scout message randomly to one neighbour. Then the neighbouring node stores information about the originating node, updates the information in the scout message and sends it to the next random neighbour (though, not a node that has already forwarded this particular message). This process continues, with each intermediate node gaining new information about the nodes on the scouts path, until a "dead-end" or the attainment of maximum hops specified for this scout message. The final node on the scout messages path will store and update information as before, but instead of sending it to a new random neighbour node, it sends it back to the previous node on the scout messages path. All intermediate nodes will now gain information also from the "forward" direction of the scout messages path while one by one returning the scout message back to the originating node through the same path, as shown in figure 11.

Algorithm 1 The process for handling a received scout message.

```
UPDATESCOUTMEMORY()                      ▷ extract and store information from the message
if scout.sourceAddress == myAddress then                   ▷ if this is the originating node
    delete scout;
else if scout.isReturning == true then          ▷ if the scout message is on its way back home
    RETURNSCOUT()                                 ▷ send the scout message back to the source
else          ▷ otherwise the scout message must be forwarded to the next random neighbor
    scout.path ←myAddress                         ▷ insert this nodes address into the scouts path
    if scout.serviceID == myServiceID then          ▷ increment the fitness count if necessary
        scout.fitness + 1
    end if
    if scout.hopCount == scout.maxHopCount then               ▷ if maxhops has been reached
        RETURNSCOUT()                             ▷ send the scout message back to the source
        return;
    end if
    GETRANDOMNEXTHOP()
    FORWARDSCOUT()                               ▷ forward the scout to a new random node
end if
```

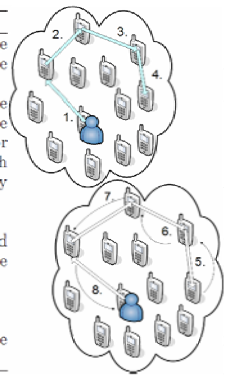


Fig. 11. Handling of a received scout message and the random hopping nature of the scout messages

Due to this ongoing process, the knowledge of the service situation in the neighbourhood is distributed among the nodes in the network island i.e. to which direction messages pertaining to a specific service should be forwarded. The longer this process goes on, the more such knowledge will be gained by the nodes receiving and forwarding scout messages.

3 Evaluation Results

3.1 Simulations

For evaluating feasibility of the SAF model, it was implemented in the OMNeT++ simulator against the popular Epidemic Routing model. The key modules, that are present in the sequence diagrams (figures 2-10), are: 1) “SAF” which implements the actual forwarding at network layer, 2) “NSM” for keeping the NSDB, 3) “SafPool” for storing messages and 4) “CSM” for having a more abstract view of different contexts (e.g. resources) that affect the communicational situation. Figure 12 lists used simulation parameters in the evaluations. Mobility pattern which the nodes followed was A Modified Reference Point Group Mobility Model with Dynamic Clustering (MRPGDC) described in [15]. Its models dynamic movement of human groups with a possibility that a person can leave his/hers group and start to follow a new group of people. For example, in real life these kinds of situations can be observed when people move in cities on mass traffic vehicles and in traffic jams.

Parameter	Value
Area size	600 m x 600 m
Nodes' speed	3 m/s
Number of CenterNodes	5
CenterNodes' speed	4 m/s
Mobility pattern	(MRPGDC)[1] [113]
Chance of dynamic clustering	0.1 %
Bitrate	2 Mb/s
Carrier frequency	2.4 GHz
Max transmission power	4 mW
Signal attenuation threshold	−110 dBm
Path loss coefficient alpha	4
MAC protocol	IEEE 802.11 WLAN
Bytes per control message	200

Parameter	Value
Bytes per data message	1400
Number of runs per scenario	100
Battery energy consumed in TX	25.0 mA
Battery energy consumed in RX	8.0 mA
NSM reduce factor	0.4
NSM reduce period	3.0 s
NSM minimum chance	0.02
Storage purge delay	0.5 s
Urgent category purge time	60.0 s
Normal category purge time	30.0 s
Bulk category purge time	15.0 s
Application sending frequency	0.5 1/s
Simulation time	250 s

Fig. 12. Used simulation parameters

Figure 13 depicts what is the total traffic generated by the nodes. The amount of bytes is calculated as a mean of control + data traffic for a single node measured from the network layer of the nodes. A clear trend can be seen as the SAF model generates less traffic with a quite linear ratio to the number of nodes present in the network. This is mainly because SAF sends much less control messages. The epidemic model instead peaks at 40 nodes, after which the restrictions of resources starts to have an effect. Next, we inspect what are the successful delivery rates of the models. Starting with only ten nodes, both models perform equally. With 20 nodes, the percentage difference starts to grow, and from 30 nodes and beyond, epidemic routing's rate gets worse so that with 90 nodes, it is only about 2.5 per cent. The SAF model is able to keep up the delivery rate around 40 % with node counts of 20 or more.

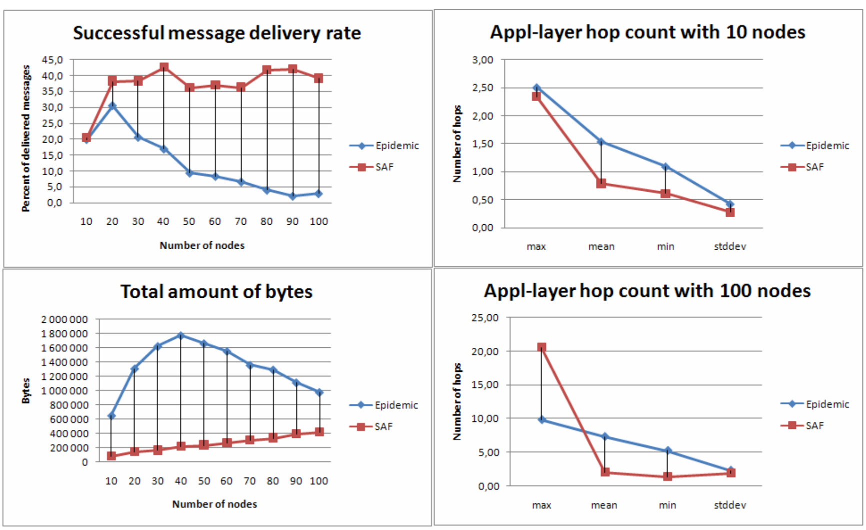


Fig. 13. Simuxlation results

In figure 13, also application (or service) layer hop counts for messages that were successfully transmitted to their destinations are depicted. The figures have been plotted with the number of nodes being 10 and 100 respectively. With both figures, SAF has smaller mean delays than with epidemic routing model. This is interesting because if the epidemic routing would work in an ideal world with endless resources, it should disseminate messages quickly as it would pass the messages blindly everywhere. However, the restrictions of storage space, messages' validity and categories have a big impact on how well that kind of approach works in a more realistic environment which has been used in the simulations. One should still notice also that in more dense scenarios, SAF can suffer from higher maximum hop counts. That is an indication that the algorithm seems to favor forwarding paths where a message has to travel through more nodes. However, from a bigger perspective does not seem to be large problems as the total network traffic numbers are smaller with SAF.

3.2 Discussion

The evaluation of the situated service oriented messaging has been carried out using OMNeT++ discrete event simulation environment. First the SAF module has been developed, and then the SOF module is investigated. The first simulation results indicate that SAF can outperform Epidemic Routing in many aspects as discussed in section 3.1. The SOF module does not require any complex, time consuming computations nor extensive memory storage capabilities, but only simple decisions and small memory buffer. Therefore, the realization of algorithm is scalable also to small mobile devices. It is estimated that the SOF further optimizes the messaging performance, and after realizing it, the final performance evaluation of the situated service oriented messaging can be performed. Especially, the case where a large number of nodes spread unevenly with pertinent nodes strewn among them randomly is interesting, because it is assumed that the messages will be more efficiently distributed to the pertinent nodes while causing less disturbance the non-interested nodes.

The evaluation of the situated service oriented approach still left open how the solution operate in more complicated situations such as different topologies, mobility, multiple radio technologies, strong security requirements etc., and especially the scalability properties of the solution.

4 Conclusions

The key contribution of the paper is the concept for situated service oriented messaging and novel algorithms for biologically inspired opportunistic networks. Simulations are applied to evaluate the usefulness of the concept. The solution utilizes different contextual information sources to create and update a view of the communicational situation. Smart diffusion of relevant control data between neighbouring nodes using novel swarm intelligence based method enables spreading of information only to the interested nodes without unnecessarily disturbing the non interested nodes. The provided methods are then applied when deciding what to do for the incoming messages in each individual user node. The local self-organization capabilities, processing and decision making enables better scalability of the messaging. The evaluations done against epidemic routing protocol indicate that the proposed solution lowers the amount of transmissions in the network, thus reducing precious resource usage in the nodes. This is achieved without introducing further delays or deteriorations in the message delivery ratio.

The evaluation of the situated service oriented approach still left open how the solution operate in more complicated situations such as different topologies, mobility, multiple radio technologies, strong security requirements etc., and especially the scalability properties of the solution.. In the next step of this research, the aim is to simulate more complicated situations of the situated service oriented messaging and especially evaluation its' scalability properties.

Acknowledgments. This work has been partially supported by the EC within the framework of the BIONETS project (see www.bionets.eu). The authors would like to thank the project partners for the good co-operation in the project. In addition, our thanks are provided to OMNeT++ community for the working simulation platform.

References

1. Fall, K.: A delay-tolerant network architecture for challenged internets. In: SIGCOMM 2003: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 27–34. ACM Press, New York (2003)
2. Farrell, S., Cahill, V., Geraghty, D., Humphreys, I., McDonald, P., et al.: When TCP Breaks: Delay-and Disruption-Tolerant Networking. IEEE Internet Computing 10(4), 72–78 (2006)
3. Cerf, V., et al.: Delay-Tolerant Networking Architecture. RFC 4838, IETF (2007)
4. Pelusi, L., Passarella, A., Conti, M.: Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad hoc Networks. IEEE Communications Magazine, 134–141 (November 2006)
5. Ott, J., Kutscher, D., Dwertmann, C.: Integrating DTN and MANET routing. In: CHANTS 2006: Proceedings of the 2006 SIGCOMM workshop on Challenged networks, pp. 221–228. ACM Press, New York (2006)
6. Zhang, Z.: Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. IEEE Communications Surveys & Tutorials 8(1), 24–37 (2006)
7. Vahdat, A., Becker, D.: Epidemic routing for partially connected ad hoc networks. Research report, Duke University (2000)
8. Burns, B., Brock, O., Levine, B.N.: MV routing and capacity building in disruption tolerant networks. In: Proc. Infocom. IEEE, Los Alamitos (2005)
9. Widmer, J., Le Boudec, J.Y.: Network coding for efficient communication in extreme networks. In: Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 284–291 (2005)
10. Musolesi, M., Hailes, S., Mascolo, C.: Adaptive routing for intermittently connected mobile ad hoc networks. In: Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), pp. 183–189 (2005)
11. Leguay, J., Friedman, T., Conan, V.: DTN routing in a mobility pattern space. In: Applications, Technologies, Architectures, and Protocols for Computer Communication, pp. 276–283 (2005)

12. Latvakoski, J., Hautakoski, T.: Situated Message Delivery for Opportunistic Networks. In: ICT Mobile and Wireless Communications Summit 2008, Stockholm/Sweden, June 10-12 (2008)
13. Di Caro, G., Dorigo, M.: AntNet: Distributed Stigmergetic Control for Communications Networks. Journal of Artificial Intelligence Research (JAIR) 9, 317–365 (1998)
14. Pham, D., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm–a novel tool for complex optimisation problems. In: Intelligent Production Machines and Systems: 2nd I* PROMS Virtual Conference, July 3-14, p. 454. Elsevier Science Ltd., Amsterdam (2006)
15. Szabó, S. (ed.): The initial mathematical models of new BIONETS network elements and algorithms. BIONETS (IST-2004-2.3.4-FP6-027748) Deliverable D1.3.1 (2006)