Dissecting the routing architecture of self-organizing networks

Marcelo Amorim[1]*, Mihail L. Sichitiu[2], Farid Benbadis[1], Yannis Viniotis[2], Serge Fdida[1]

[1] LIP6/CNRS – UPMC
8, rue du Capitaine Scott
75015 – Paris – France
{amorim,benbadis,sf}@rp.lip6.fr

[2] Dept. of ECE
North Carolina State University
Raleigh, NC 27695-7911
{mlsichit,candice}@ncsu.edu

Abstract

The proper operation of self-organizing networks (SON) relies on the autonomous behavior of their individual nodes. Routing in such networks has been a challenging task since their conception, due to their non-traditional characteristics and design requirements. Although a huge amount of routing architectures and protocols for SONs has been proposed, very little work has been done on the fundamental characteristics that make a routing strategy efficient for a particular network and/or design requirement. Contrary to traditional techniques where the routing architecture is structured as a single unit, we suggest in this paper that routing be thought of as a combination of four main architectural components, namely addressing, dissemination, discovery, and forwarding. This logical decomposition offers significant advantages both from the analysis and the design points of view.

1 Introduction

The term "self-organization" has been used in diverse domains, ranging from engineering to physics to economy to sociology. In networking, it has been widely applied to describe functionalities in various layers of the protocol stack. Loosely speaking, a self-organizing network (SON) lacks *a priori* infrastructure, that would be present in a traditional network (*i.e.*, the Internet) at power-up time; instead, the infrastructure is built-up progressively, as, for example, nodes join the SON, and has to be rebuilt as nodes leave the SON. This lack of *a priori* infrastructure and the need for (sometimes frequent) infrastructure updates poses design requirements that are not present in traditional networks. In this paper, we focus on self-organization at the network layer.

Various classes of network architectures fall under the umbrella of self-organizing networks. Some examples of such classes are the so-called peer-to-peer overlays [1], ad hoc networks [2], sensor networks [3], and wireless mesh networks [4]. Note that these classes do not need to address the same design requirements; neither do they satisfy the same modeling assumptions.

For the purposes of this paper, the main differences between SONs and existing networks can be summarized as follows: (i) there is no central authority in charge of addressing (*e.g.*, no central DNS service, as in fixed IP networks), and (ii) network nodes can join/leave the network or move around in an arbitrary fashion. The implication of these two characteristics is that execution of routing-related functions (*e.g.*, resolution of names, mapping of addresses to present node/user location, etc.) (i) cannot be done *a priori*/must be built gradually as nodes join a SON, and (ii) has to be distributed among nodes and reallocated as frequently as a node joins, leaves, or moves away.

Traditional design approaches of routing architectures are not directly (or efficiently) adaptable to SONs. A large number of routing protocols for self-organizing networks has been proposed so far. The ones developed first were intended to be general and were basically an adaptation of existing approaches (*e.g.*, drawn from

---

[*] Contact author. Email: `Marcelo.Amorim@lip6.fr`.

wired networks) to support dynamic nodes. The essential differences between wired and wireless networks (in terms of both design requirements and operating characteristics) later led to proposals for completely different routing strategies. Nevertheless, these novel proposals have appeared in a quite disordered way.

In this paper, we decompose the routing functionality into four fundamental blocks: addressing, dissemination, discovery, and forwarding. In this context, we argue that any (proposed) routing architecture is composed of a combination of one, some, or all of the fundamental blocks (cf. Fig. 1). Furthermore, each block is associated with a weight, which corresponds to its impact on the overall performance of the system. In this sense, a particular composition determines the behavior of the routing architecture and its associated protocols.

The value-add that this logical decomposition brings about is three-fold. First, it enables the designer/practicing engineer *clarify* confusing concepts regarding the routing architecture of SONs. Not all classes of SONs need to address the same design requirements (see Section 2 for a description of such requirements). Even within the same class (*e.g.*, sensor networks), requirements may differ depending on the application. Second, it enables uniform classification of existing approaches based on the four building blocks, and thus provides a consistent mechanism to *analyze* them. Third, it provides a thorough mechanism to *quantify design* of new routing approaches, that are able to meet requirements.

The remainder of this paper is organized as follows. The inherent and some desired features of SONs are presented in Section 2. In Section 3, we describe the four basic blocks of a complete routing strategy. In Section 4, we analyze a number of existing techniques. We also provide examples of how our logical decomposition can be used to design a new routing architecture/protocol that meets given requirements.

2 Requirements for self-organization

A SON must be both *autonomous* and *adaptive*,[1] and these characteristics are guided by the requirements of a specific scenario. Indeed, the exact behavior of the network is unknown at the time it is deployed. In the following, we describe six design requirements that directly affect the routing architecture.[2]

- *Infra-structure free*: the basic requirement of self-organization is that it does not require any predefined infrastructure to which nodes must fit (*e.g.,* IP-based networks). This infrastructure is established on-the-fly and may have a varying degree, ranging from none at all to some. Each node can leave the topology at any moment without any consequence. That means that no element is essential to keep the network in good working order.

- *Robustness to mobility*: this is a fundamental requirement that many SONs may face. Although mobility has been addressed by a large number of works, it has been frequently considered as a default parameter. Different mobility degrees exist, and by adjusting them to meet specific requirements, other parameters (*e.g.*, complexity or scalability) can be more easily addressed.

- *Scalability*: wireless networks are inherently not scalable [5]. Nevertheless, physical capacity bounds are a matter of lower-level layers; we believe that the routing architecture should abstract such limitations and be designed to be scalable no matter what the characteristics of the underlying architecture (for example, think of a hybrid network with a large fixed backbone and a large number of wireless clouds attached as end-points in a self-organizing fashion). For a network to scale well, nodes should keep invariant overhead within a limited scope and any operation or changes should affect only a few nodes, independently of the total number of nodes in the network.

- *Distribution of tasks*: unless the application explicitly requires some centralization (*e.g.*, sinks in wireless sensor networks), a SON should be intrinsically distributed. A distributed SON must be fully operational through the autonomous operation of the individual nodes.

---

[1] We refer here to the ability of the network to adapt to different behaviors at the routing level.

[2] This list is of course not exhaustive. Other parameters such as robustness to failures and security are of great importance. Nevertheless, they do not fall into the same category of the requirements presented in this paper and would require special attention. Because of space limitation, we do not address these issues in this paper.

- *Low complexity*: since there is no *a priori* knowledge concerning the processing capacities of the nodes, it is important to keep management operations as simple as possible. Any overhead that is not strictly necessary may have a consequence on the efficiency of the routing protocol. For instance, consider a delay-sensitive SON: high overhead and a large number of interactions among nodes may lead to unexpected latency and consequently to delivery of messages that would be useless to the application.
- *Fairness*: it is fundamental that management tasks be fairly distributed among nodes. The term "fairness" does not necessarily mean that the overall load should be equally shared among the nodes, but that a node should be assigned an amount of responsibility proportional to the node's capacity. The capacity of a node is given by simple predefined rules and becomes effective through negotiation, which depends in turn on the network status.

Depending on the objectives of the designer, some of the requirements may be conflicting. This explains why it is important for a SON to clearly identify the desired features before deployment.

3 The four building blocks of a complete routing architecture in SONs

In this section, we describe the four basic blocks of a routing strategy.[3] We assume that the network is composed of a number of autonomous entities, possibly mobile, provided with compatible access techniques.

The four building blocks defined below, when combined, determine the characteristics of a routing scheme. We will see in Section 4 how existing solutions behave under particular situations and how the fundamental building blocks should be combined to satisfy some requirements.

3.1 Addressing/positioning

The topological identification of a node is given by its address. The way addresses are assigned to nodes is a fundamental issue and has a direct impact on the performance of the routing protocol. Addressing schemes can be classified in two main groups: topology- and position-based ones.

In topology-based addressing schemes, either a flat or a hierarchical strategy can be applied. In the flat addressing scheme, nodes choose an address (almost) independently of the other nodes. In this case, some collision-avoidance mechanism must be implemented to guarantee that two nodes do not use the same address. Such a mechanism is inherently available in hierarchical addressing schemes, where the main objective is scalability. Nevertheless, hierarchy implies structure, and a number of drawbacks may arise. For instance, in the case of mobile networks, the use of a hierarchical addressing structures may be prohibitive.

A possible alternative to the topology-based approach is to define an absolute and collision-free addressing mechanism through a position-based, geographic, scheme. In this case, the node's address is equivalent to the node's geographic position. Such an approach has received increasing attention in the literature. The reasons are twofold. First, positioning systems (*e.g.*, GPS) are becoming cheaper and smaller, and their use in practical systems can be easily envisaged. Second, position-based forwarding schemes are easier to deploy and manage. However, these systems are not sufficiently precise to give distinct addresses to two elements very close to each other (which may not be a drawback). We will see in this paper that the addressing/positioning system plays a major role in the behavior of a SON.

3.2 Dissemination

Once a node knows its own address, it must make this information available to potentially corresponding nodes and eventually to the entire topology. The dissemination degree often determines the complexity of

---

[3] Another important but often neglected component of a communication system is naming. A node's name is closely related to the idea of user friendliness and constitutes the interface between the upper- and lower-level identifications of a node. A node's identifier may have two different, not necessarily orthogonal, meanings. On the one side, it may identify a user and be completely independent of the characteristics of the network. On the other side, it may refer to a particular equipment or network zone. Naming systems are specially important in routing schemes where the address of a node's location server is obtained from the node's name. DHT-based solutions are an example of such routing schemes (cf. Section 4.1.4). In this paper, we do not address naming as an independent building block.

the discovery algorithm (cf. Section 3.3).

Dissemination can range from none to full. Zero dissemination means that each node stores location information for itself, which implies that any communication must be established in an on-demand basis. In a full dissemination scheme, each node has a copy of the location of all the other nodes. An intermediate case is, of course, limited dissemination, where the management of location information is shared among nodes. This type of dissemination can be classified as *centralized* or *distributed*. In the centralized case, some well-defined nodes know the addresses of the other nodes, whereas in the distributed case, all nodes store some other node's address in a peer-to-peer fashion.

It is also important to investigate the confidence level of the disseminating algorithm. In some cases, high confidence is required by a specific scenario/application and location information must be maintained up-to-date in each replica. Although this clearly generates a high overhead, it has the advantage of prompt address availability. Furthermore, it allows for more efficient security and trust algorithms to be deployed.

It is also possible to implement dissemination techniques that rely on increasing confidence methods. In such a scheme, the location information is associated with a confidence value, and messages are forwarded between nodes with increasing confidence values until the destination is reached.

3.3 Discovery

Closely related to the dissemination strategy is the discovery system. Discovery (also known as localization or query) is the act of obtaining a node's address by means of a discovery service.[4] The discovery and dissemination systems form together a key part of the routing structure. Their complementarity gives the scenario in which the routing performance is maximized.

Some routing protocols use flood-based route discovery, where a route request packet is flooded through the network until a node responds with a valid path to the destination. The relative cost of route discovery in such protocols tends to be very expensive. The discovery step is then a key element in the global performance of such SONs.

3.4 Forwarding

The ultimate goal of a routing scheme is of course effective data transmission, implemented by the forwarding scheme. It dictates the quality of the communication and is what users observe. The performance of the forwarding mechanism (*e.g.*, path length, cost, delay, energy consumption) closely depends on the previous three components. For instance, messages can be forwarded following a hierarchy given by the addressing scheme or a geographic path computed through Euclidean distances between nodes. It can be predefined (source routing) or based on a hop-by-hop basis. The choice of what forwarding scheme to use also depends on desirable qualitative properties such as robustness, path length, power consumption, message delivery success, and scalability.

3.5 Summary

Any routing architecture for SONs can be represented by a combination of the four basic building blocks. Different possibilities for the combinations are summarized in Fig. 1 (note that we do not consider naming as a fundamental building block). Dashed lines mean either that two basic blocks can be combined into one or that they can be performed in parallel. As we will see in Section 4, a particular combination and the importance (weight) given to each one of the building blocks determine together the behavior of the architecture. Possible dissections of a routing protocol are presented in Fig. 2. Consider for example Fig. 2(a) that shows the Terminodes strategy (cf. Section 4) which uses all four building blocks.

---

[4] We make, for the sake of clarity, the following distinction between positioning and locating systems. Positioning is the act of giving positions (coordinates) to nodes. Locating is the procedure through which a node discovers the position (coordinates) of some other node.

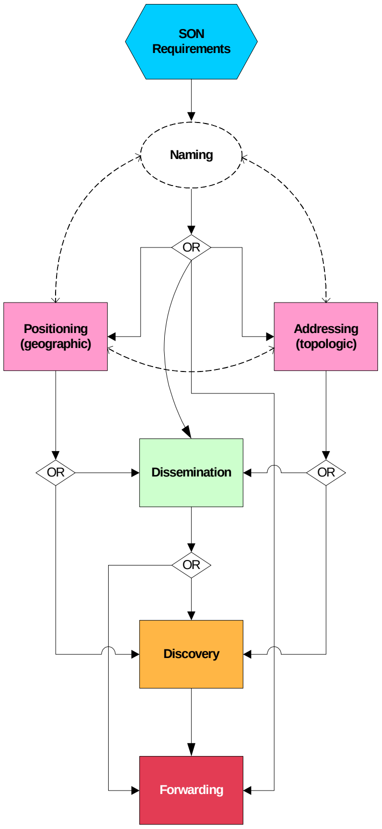

Figure 1: Flowchart of possible combinations of the four components of a SON.

4 Dissecting some existing approaches

In the following, we first provide a brief description of some existing approaches, in terms of the four building blocks described above (cf. Fig. 1). We then present a more complete analysis considering the impact of each building block on the complexity of a number of particular solutions found in the literature. We do not intend here to fully survey existing routing techniques, but to present some of them from a designer's viewpoint. For comprehensive reviews of the following approaches, refer to [2].

4.1 Routing architectures under consideration

In this section, we analyze several fully distributed routing protocols designed for SONs.

4.1.1 Reactive/On-demand

Reactive protocols (*e.g.*, AODV [6]) establish paths on an on-demand basis, and maintain in memory only the routes in use (generally controlled by a timer). Because of this, packets may be lost with topology changes, which requires a considerable traffic overhead to maintain valid routes. Such a strategy makes it possible for nodes to avoid topology-wide knowledge and to reduce the impact of the dissemination component. Reactive protocols are then basically composed by the discovery (the strongest component) and forwarding building blocks, as depicted in Fig. 2(b), which shows an example of the architecture of a reactive geographic protocol.

4.1.2 Total replication/Proactive

In proactive routing protocols (*e.g.*, OLSR [7]), every node maintains a forwarding table with entries to all other nodes in the network. In contrast to the reactive case, proactive protocols do not need the discovery

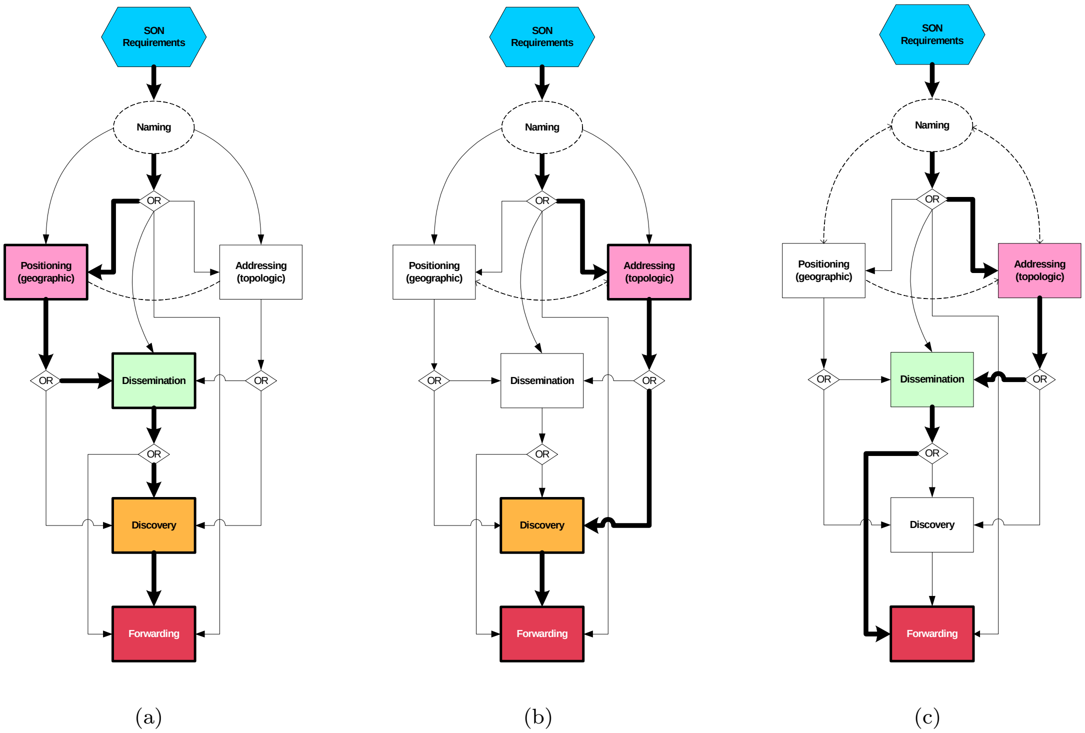

Figure 2: Some examples of combinations of the basic building blocks.

building block. The main components of a proactive protocol are the dissemination and forwarding ones, with the major role played by the dissemination block (cf. Fig. 2(c)).

4.1.3 Position-based

A large class of position-based routing protocols [8, 9, 10] assume that every node is aware of its physical position (*e.g.*, by using a GPS receiver). The sources use a dissemination/discovery protocol to find the current location of destination. All nodes on the path then forward the packet in the physical direction of the destination.

4.1.4 DHT-based

Routing protocols based on distributed hash tables (DHT) distribute location information throughout the network in nodes called rendezvous points (*e.g.*, Terminodes, HIGH-GRADE, GLS, SLURP, SLALoM, DLM, [8, 9, 11]). The address of a node's rendezvous point is obtained by applying a hash function to the node's name. This is performed in a proactive fashion but, contrarily to the previous case, only a set of rendezvous nodes knows the current address of a destination.[5] The three main building blocks that form the core of a DHT-based routing protocol are the addressing (or positioning), dissemination, and discovery ones. DHT-based approaches are a good example where two building blocks (dissemination and discovery) are tightly connected and cannot be addressed separately.

4.1.5 Others

Chord [1] provides an efficient dissemination and discovery protocol for peer-to-peer networks. In Chord the nodes are organized in a ring with every node responsible for an equal share of the information in the network.

---

[5] This set can contain only one node.

In this respect Chord is similar to DHT-based systems. Gnutella is one of the most popular peer-to-peer protocols on the Internet. Gnutella uses flooding to discover the information in the network, thus the entire complexity of the protocol is in the discovery phase. The IETF MANET autoconfiguration protocol assigns unique IP addresses for ad hoc network in a decentralized manner by randomly choosing an IP address and flooding the network to make sure that no other node in the network has that IP address. GPS-free positioning systems can be used to provide relative positioning for position based routing algorithms when GPS receivers are not available. The approach in [12] (proposed for the Terminodes project) constructs a local system of coordinates for each node (based on the positions of their neighbors) and then uses the coordinate system of the node with the smallest `ID` to synchronize the local systems of coordinates of all nodes.

4.2 Analysis

In this section we discuss several design aspects that have to be considered when designing the routing architecture of a SON. Considering the large number of existing approaches it is clear that no single solution outperforms all others in all situations. Therefore, some solutions may be preferred depending on the scenario and the goal of the SON. We make the following notations:

- N is the total number of nodes in the network.
- M is the average number of neighbors for each node.
- D is the average number of destinations for a node.
- Q is the network diameter. For two dimensional ad hoc networks $Q = O(\sqrt{N})$.

In a complete implementation of a SON, all four building blocks will be present. The addressing is, usually, fairly independent from the last three blocks, and a designer can, usually, mix and match the addressing method with any of the existing routing approaches.

For some networks, addressing may not be necessary. For example, if a GPS receiver is available for each node, the geographic address can be easily obtained. However, in general, a designer cannot interchange the remaining three blocks arbitrarily. For example, the dissemination phase of an on-demand routing algorithm cannot be coupled with the discovery phase of a proactive one. However, in some cases, it is possible to combine the forwarding method of one approach with the dissemination and discovery phases of another (especially if they handle the same type of addresses). For example, the forwarding approach used for the Terminodes project [8] may use the dissemination and discovery schemes presented in GLS [9].

The decomposition approach proposed in this paper can be used to analyze the performance of a given SON. For the analysis we will provide the communication and storage costs of a SON for two primitive operations:

- The cost of a node joining the network. This cost has to take into account the cost of finding routes to D destinations (and becoming the destination of D sources).
- The cost of maintaining network state due to mobility. For table-driven protocols, this is the costs of repairing a broken route. For geographical based protocols this is the cost of updating the location information. In general, this cost is only due to node mobility (and directly proportional to the node's average speed).

The cost of setting up a stationary network of N nodes is N times higher than the cost of a single node joining the network. Depending on the particularities of the SON under consideration, some of the costs are more important than others. For example, in a sensor network that experiences practically no node joins and no mobility, only the cost of setting up the network is relevant. In highly mobile networks with frequent link breakage, it is likely that the cost of route recovery will dominate all other costs.

We assume a uniform node density and a constant density with the increase in the number of network nodes N (*i.e.*, a constant number of neighbors M). We also assume that the destination nodes are chosen randomly independent of the physical distance between the source and the destination. We neglect the effect of retransmission errors.

Table 1 summarizes the communication costs for some protocols designed for SONs. Some of the entries were obtained from the original papers and some of them from our own analysis. An N/A entry signifies that the protocol does not explicitly specify one fundamental block. The communication cost is expressed as the total number of transmissions assuming that only neighbor nodes can communicate directly.

Table 1: Scalability of some existing SON protocols with respect to the communication cost.

Scheme	**Influence of the basic blocks**			
	Addressing	*Dissemination*	*Discovery*	*Forwarding/packet/hop*
Flooding & gossiping	N/A	$O(1)/O(1)$	$O(1)/O(1)$	$O(\frac{N}{Q})/O(1)$
On-demand	N/A	$O(1)/O(1)$	$O(ND)/O(N)$	$O(1)/O(1)$
Proactive	N/A	$O(N)/O(N)$	$O(1)/O(1)$	$O(1)/O(1)$
Geo-routing, DREAM	N/A	$O(N)/O(N)$	$O(1)/O(1)$	$O(1)/O(M)$
Terminodes	$O(M)/O(M)$	$O(Q)/O(Q)$	$O(DQ)/O(DQ)$	$O(1)/O(M)$
SLURP	N/A	$O(Q)/O(Q)$	$O(DQ)/O(DQ)$	$O(1)/O(M)$
HIGH-GRADE	N/A	$O(Q)/O(\log N)$	$O(DQ)/O(DQ)$	N/A
GLS	N/A	$O(Q)/O(Q)$	$O(QD)/O(DQ)$	$O(1)/O(M)$
DLM, SLALoM	N/A	$O(Q)/O(\sqrt[3]{N})$	$O(D\sqrt[3]{N})/O(D\sqrt[3]{N})$	N/A
Chord	N/A	$O(Q\log N)/O(Q\log N)$	$O(DQ\log N)/O(Q\log N)$	N/A
Gnutella	N/A	$O(1)/N/A$	$O(DN)/N/A$	N/A
Autoconfiguration	$O(N)/O(1)$	N/A	N/A	N/A

For a given system, the communication cost can be evaluated from its components in Table 1. For example, for the Terminodes approach the cost due to mobility is $O(M + Q + DQ + M) = O(DQ + M)$.

Similarly to the communication cost, we can evaluate the storage requirement of a SON. Table 2 summarizes the storage requirements for each surveyed architecture.

Table 2: Scalability of existing protocols with respect to the storage cost.

Scheme	**Storage**
Flooding	$O(N)$
On-demand routing	$O(\min(N, QD))$
Proactive	$O(N)$
Geo-routing	$O(N)$
DREAM	$O(M + D)$
Terminodes	$O(D + M)$
SLURP	$O(D + M)$
HIGH-GRADE	$O(D + \log N)$
GLS	$O(D + \log N)$
DLM	$O(D + \sqrt[3]{N^2})$
SLALoM	$O(D + \sqrt[3]{N})$
Chord	$O(D)$
Gnutella	$O(D)$
Autoconfiguration	$O(1)$

4.3 Choosing an architecture to meet design objectives

Tables 1 and 2 can also help to design a SON (*e.g.*, to choose an architecture among the surveyed ones) that performs well with respect to one or more of the requirements described in Section 2. The problem of optimizing *multiple* simultaneous requirements is, in general, considerably more difficult and will not be considered here.

Assume, for example, that the primary concern of a designer is to overcome the high mobility cost associated with a large network. According to Table 1, the flooding protocol has practically no mobility cost and thus should be preferred. All other protocols will have high cost for route repairs. This comes from the fact that the chosen protocols should have a low cost for route repair, as it will dominate the total communication cost. However, the flooding protocol is very expensive in terms of packet forwarding, and if the designer expects heavy traffic, a different approach should be preferred.

The main advantage of a proactive protocol is time economy during the lookup phase, because it maintains routes to all possible destinations. But this can also be very expensive. If the designer knows that each node in the network only communicates to a few possible destinations she/he should use an on-demand routing algorithm, as these protocols only maintain routes to the current destinations.

If the nodes in the network have limited storage resources (*e.g.*, a sensor network or an RF-ID tag), the designer should choose a protocol with modest storage requirements (*e.g.*, a geographic forwarding protocol).

For a large stationary network, the designer should choose a combination of protocols that has the lowest setup complexity. In this case, an important parameter would be the network density. Indeed, if much control overhead is introduced by the dissemination and/or discovery phases, the resulting impact on the physical link (*i.e.*, interferences) would invalidate the architecture. It would be preferable then to stress on the forwarding building block.

Some of the presented approaches work only in ideal conditions. For example, for geographic forwarding algorithms, it may happen that no routes from source to destination are found although they may exist. A designer may use a backup algorithm, or, if the network may tolerate some data loss, just accept the limitation of these algorithms.

Finally, if the network is known to be relatively small, and the nodes have plentiful resources, considerations other than scalability (*e.g.*, fairness or ease of implementation) should dictate the choice of the protocol stack.

5 Conclusion

In this paper we have proposed a logical decomposition of a SON routing architecture into four fundamental functionalities, namely addressing, dissemination, discovery, and forwarding. This logical decomposition can be used to establish: (a) a consistent mechanism to analyze existing approaches (and hence compare them on an equal basis), and (b) a thorough mechanism to quantify design of new routing approaches, that are able to meet predefined system requirements (such as scalability, management overhead, etc.). Thus, the approach offers significant advantages to the network engineer, both from the analysis and the design points of view.

References

[1] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: a scalable peer-to-peer lookup protocol for internet applications,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, Feb. 2003.

[2] M. Abolhasan, T. Wysocki, and E. Dutkiewicz, “A review of routing protocols for mobile ad hoc networks,” *Ad Hoc Networks*, vol. 2, no. 1, pp. 1–22, Jan. 2004.

[3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[4] B. Schrick and M. Riezenman, “Wireless broadband in a box,” *IEEE Spectrum Magazine*, no. 6, pp. 38–43, June 2002.

[5] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[6] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (AODV) routing.” RFC 3561, July 2003.

[7] T. Clausen and P. Jacquet, “Optimized link state routing protocol (OLSR).” RFC 3626, Oct. 2003.

[8] L. Blazevic, L. Buttyan, S. G. S. Capkun, J. P. Hubaux, and J. Y. L. Boudec, “Self-organization in mobile ad-hoc networks: the approach of terminodes,” *IEEE Computer Communications Magazine*, pp. 166–174, June 2001.

[9] J. Li, J. Jannotti, D. D. Couto, D. Karger, and R. Morris, “A scalable location service for geographic ad-hoc routing,” in *Proceedings of ACM Mobicom*, (Boston, MA), Aug. 2000.

[10] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, “A distance routing effect algorithm for mobility (DREAM),” in *Proceedings of ACM Mobicom*, (Dallas, TX), Oct. 1998.

[11] Y. Yu, G.-H. Lu, and Z.-L. Zhang, “Enhancing location service scalability with HIGH-GRADE,” in *Proceedings of IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004)*, (Fort Lauderdale, Florida), Oct. 2004.

[12] S. Capkun, M. Hamdi, and J. P. Hubaux, “GPS-free positioning in mobile ad-hoc networks,” *Journal of Cluster Computing*, vol. 5, no. 2, no. 2, pp. 157–167, 2002.