



Toward Collective Self-Awareness and Self-Expression in Distributed Systems

Michele Amoretti and Stefano Cagnoni, University of Parma

Simultaneously applying hierarchy and recursion enables self-awareness and self-expression in distributed systems, which can provide greater efficiency and scalability in tasks such as network exploration and message routing.

Self-aware computing systems and applications proactively maintain information about their own environments and internal states. The term self-expression includes goal revision and the self-adaptive behavior that derives from cooperative reasoning about the knowledge associated with the system's self-awareness.

An individual node's self-expression occurs "as a result of the node's review of its state, context, goals and constraints and subsequent behavior adaptation."¹ Many techniques exist to enable self-awareness and self-expression in a single node, but the self-awareness challenge is much greater in a distributed system in which multiple autonomous nodes must communicate

efficiently to achieve a global goal. Abstractly, a node is a computational entity that conceptualizes locality within a global system; concretely, it is any element in a distributed system, from a router characterized by its protocols or a subnetwork with a specific IP domain to a client or server application or a set of peers in a large network.

In a network with the global goal of high efficiency, simple search algorithms such as random walk (RW), which randomly selects the next hop from among nodes generating probe messages, will not suffice because they fail to exploit the network's recursive or hierarchical features. Thus, exploring the entire network requires many probe message propagations, even for networks with favorable topological features, such as scale invariance.²

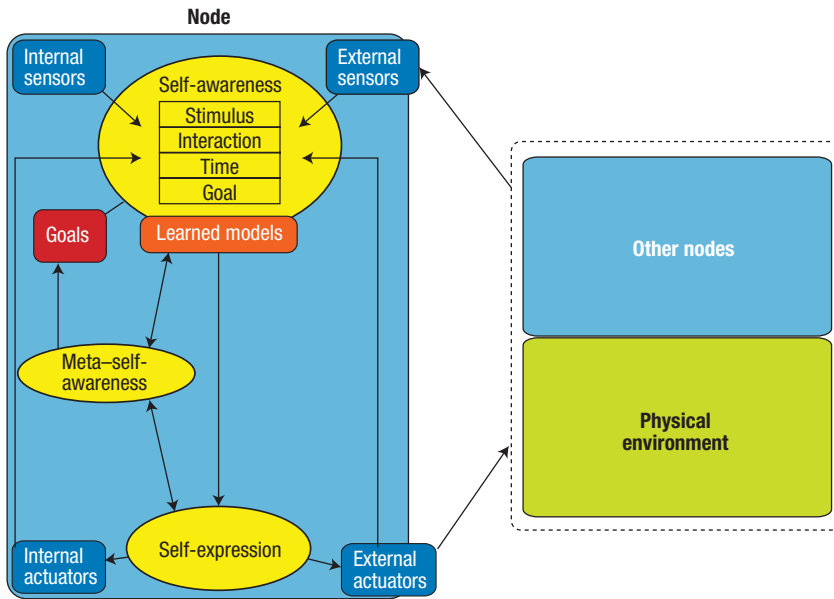


FIGURE 1. Representing a self-aware and self-expressive node. The node uses information from its internal and external sensors to construct private and public self-awareness.

To address this and other challenges, we have developed a strategy based on the simultaneous application of hierarchy and recursion (HR) to enable self-expression and self-awareness in ensembles of cooperating computational entities. Ensembles with these properties can lead to the entire system's global awareness.

Hierarchy, the categorization of nodes in a group according to their capability or status, is an important concept in dealing with scalability issues in the Internet of Things (IoT) and is a key notion in strategies such as content-centric networking.³

Recursion is the repeated use of a single, flexible functional unit to enable different capabilities over different areas of a distributed system. An example is recursive networking, which was developed to describe multilayer virtual networks that embed networks as nodes inside other networks. In the past decade, recursive networking has evolved to become a possible IoT architectural design approach⁴ and is a well-known quantum network design method.⁵

We applied our HR-based strategy to the exploration of a sample network

and determined that it can enable collective (systemwide) self-awareness and self-expression, which in turn can improve distributed system efficiency and scalability relative to traditional approaches, such as the RW algorithm, with only a minor increase in design complexity.

PROPERTIES OF A SELF-AWARE NODE

Figure 1 shows one representation of a self-aware and self-expressive node as it interacts with other self-aware nodes and its physical environment. The representation is based on a model proposed by Funmilade Faniyi and colleagues, who define a self-aware node as possessing information about its internal state and having sufficient environmental knowledge to determine how other parts of the system perceive it.¹ In their interpretation, self-awareness produces the node's behavioral model, while self-expression is concerned with goal revision and self-adaptive behavior.

Their representation is based on cognitive psychologist Ulric Neisser's self-awareness model, which consists of five increasingly complex

levels: stimulus, interaction, time, goal, and meta-self-awareness, the latter being the node's perception of its self-aware capabilities.⁶

COLLECTIVE SELF-AWARENESS

Complex systems are dynamic entities of interconnected parts that in their entirety exhibit properties that could not be inferred from examining parts individually.⁷ For example, parts of an ant colony have different roles that when observed individually might lead to a false conclusion about how the entire colony performs. The same is true for human society.¹

In computing and networking systems, an individual node's adaptiveness is reflected in its self-expression, which in turn is based on the node's self-awareness capabilities. However, collective or global, self-awareness and self-expression in a distributed system must consider outcomes for the entire network. One obvious strategy is to provide the system with a centralized omniscient monitor. However, the advantage of simplified central control rapidly diminishes in the face of the overhead required for nodes to constantly communicate their status.

COLLECTIVE SELF-EXPRESSION

A computing node exhibits self-expression if it can assert its behavior on itself or other nodes.¹ Node behavior is affected by state, goals, and constraints. Collective self-expression (also referred to as ensemble self-expression) can be defined as the ability to change the coordination pattern at run time.⁸ That is, the distributed system performs its intended operations and meets its understood goals independent of unexpected situations

by modifying its original internal organization. For example, suppose that each component of a distributed system has three collaborative approaches to complete a task: master-slave, peer-to-peer, and swarm. The system exhibits self-expression if its components can collaboratively select the most suitable strategy.

Thus, ensemble self-expression implies the assertion of collective self-adaptive behavior based on collective self-awareness. Like global self-awareness, global self-expression in a distributed system that lacks centralized control can be difficult to achieve.

HIERARCHY AND RECURSION

To illustrate the value of simultaneously using HR to enable collective self-expression and self-awareness, consider the network in Figure 2, in which packets are forwarded according to HR-based routing tables. The routing table at node 4.2 accommodates scalability by providing information about nearby destinations, in this case nodes 4.4 and node 4.7, as well as routes to remote destinations, such as subnetwork 9 (NET9).

Thus, through HR, every node has global self-awareness and self-expression. Forwarding too many packets to the same neighbor can cause congestion on that node. Feedback from congested nodes will lead to some modification in routing table use—an alternative packet destination or routing table update. The routing table update might also stem from the exchange of routing information from other known nodes. These behavior modifications, which account for changing hierarchies, gradually achieve global awareness. Indeed, the simultaneous and collaborative update

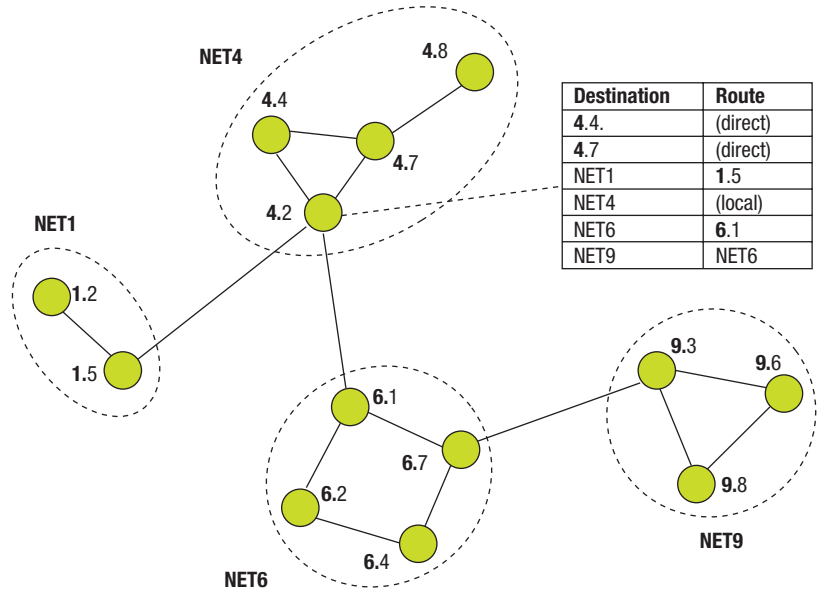


FIGURE 2. Hierarchy and recursion (HR) in a network that forwards packets according to HR-based routing tables. Each routing table (shown at node 4.2) contains information on how to reach any other known node.

of HR-based routing tables is actually a global self-expression process.

In this way, our HR-based approach satisfies all five levels of Neisser's self-awareness model:

- › *stimulus awareness*—nodes know how to manage messages;
- › *interaction awareness*—when exchanging messages and control information, nodes and subnets can distinguish other nodes and subnets;
- › *time awareness*—the network knows of past events or likely future ones, as in learning-based routing;
- › *goal awareness*—the system enforces the goal of achieving high efficiency by simplifying the search for the shortest path to a specific destination; and
- › *meta-self-awareness*—the system concurrently applies different routing strategies, choosing the best one at run time.

These examples are specific to message routing, but they illustrate the degree of system adaptiveness with

the HR-based strategy. Goal awareness, for example, is implicit in how the system populates and updates routing tables. If a global goal changes, the system could change its behavior accordingly.

NETWORK EXPLORATION

Exploring a sample network illustrates the advantages of using our HR-based method. For a network of N nodes and S subnetworks, the goal is to evaluate the approximate number of forwards that a probe message needs to propagate through the whole graph. A random node generates the probe message and sends it to one of its neighbors, which forwards it to another neighbor, and so on. A node that receives the probe message is marked as visited.

Because our HR-based strategy accounts for the presence of subnetworks and exploits collective self-awareness, fewer propagations are necessary. Every node is a member of a NET_s , where s is an element of the set $\{1, \dots, S\}$. Every NET_s has an identifier node n , where n is an element of the set $\{1, \dots, N\}$ and is unique within

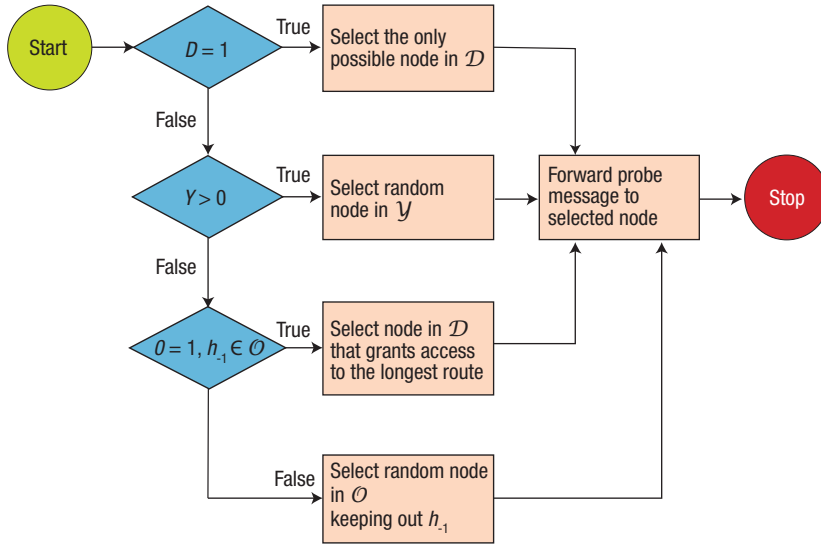


FIGURE 3. An HR-based method for exploring the sample network. D is the cardinality of the set \mathcal{D} of the considered peer’s neighbors, Y is the cardinality of the set \mathcal{Y} of neighbors belonging to the considered peer’s subnetwork that have not yet been visited, O is the cardinality of the set \mathcal{O} of neighbors that belong to other subnetworks, and h_{-1} denotes the previous hop. Routing table size is of the same order of magnitude as S .

that subnetwork. Each node has a NETs.noden designation.

A node can have neighbors that are members of other subnetworks. For example, the routing table in Figure 2 allows NET4.node2 to forward messages to

- › other nodes of NET4 that are directly reachable,
- › NET1 and NET6 through directly reachable nodes that belong to those subnetworks, and
- › NET9 through NET6.

Figure 3 is a flowchart of our HR method’s application to the sample network. Every node forwards the probe message to one unvisited neighbor in the same subnetwork. If all neighbors of the same subnetwork are visited nodes, the probe is forwarded to one random neighbor from a subnetwork that differs from the one in the previous hop. If only one neighbor belongs to other subnetworks and it is the previous hop, our method selects the neighbor that grants access to the longest route.

To simplify the routing table configuration in simulation, we assume

that every node knows which subnetworks it can reach through its direct neighbors; the neighbors provide no additional knowledge. That is, S is the same order of magnitude as the mean node degree $\langle k \rangle$, the mean number of links from the start node. For large networks, where $S \gg \langle k \rangle$, additional knowledge is needed to build meaningful collective self-awareness.

We conducted simulations with two network topologies, characterized by different node degree statistics, which we described in terms of probability mass function (PMF): $P(k) = P\{\text{node degree} = k\}$.

Scale-free network topology

In the scale-free topology, the network’s PMF decays according to a power law—a polynomial relationship that exhibits scale invariance $[P(bk) = b^a P(k), \forall a, b \in \mathbb{R}]$, such as

$$P(k) = ck^{-\tau} \quad \forall k = 0, \dots, N - 1,$$

where $\tau \in \mathbb{R}$, and $\tau > 1$ to be normalizable and c is a normalization factor. In simulation, we used the well-known

Barabási-Albert (BA) generative model,⁹ which constructs scale-free networks with $\tau \cong 3$ on the basis of growth and preferential attachment. In the BA model, every added node connects to m existing nodes, selected with probability proportional to their node degree. The resulting PMF is

$$P(k) \cong 2m^2 k^{-3} \quad \forall k > m,$$

and the mean node degree is $\langle k \rangle = 2m$.

Erdős-Rényi network topology

Networks based on the Erdős-Rényi (ER) model have N nodes, each connected to an average of $\langle k \rangle = \alpha$ nodes. The presence or absence of a link between two vertices is independent of the presence or absence of any other link, so each link can be considered to be present with independent probability p . It is trivial to show that $p = \alpha / (N-1)$

If nodes are independent, the degree distribution of the network is binomial:

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k},$$

which for large values of N converges to the Poisson distribution

$$P(k) = \frac{\alpha^k e^{-\alpha}}{k!} \quad \text{where } \alpha = \langle k \rangle = \sigma^2.$$

Simulation results

Scale-free and ER are the extremes of meaningful network topologies, since they are based on the presence of strong hubs (a hub being a highly connected node) on the one hand and a total lack of hubs on the other. Typical networks lie somewhere between the two.

Hubs are determined by the extent of Internet distribution, which researchers have shown is the combined contribution of the

provider–customer and peer-to-peer connection classes.¹⁰ Provider–customer connections are consistent with a scale-free distribution, while peer-to-peer connections follow a Weibull distribution. The deviation from scale-free distribution is largely due to Internet exchange points (IXPs)—physical infrastructures that allow autonomous systems to exchange Internet traffic, usually by means of mutual peering agreements, leading to lower costs (and, sometimes, lower latency) than in upstream provider–customer connections. Because IXPs introduce a high number of peering relationships, the higher the number of connections identified as crossing IXPs, the larger the deviation from the scale-free distribution.

Figure 4 shows simulation results in networks of 1,000 nodes, and either 20 or 100 subnetworks. With the scale-free (BA model) topology, when m is 5 and 20, the mean node degree is 10 and 40, respectively.

To have the same mean node degree for the ER topology, we set α to equal 10 and 40. Initially, the network is grown and configured—that is, the routing tables are filled after 1,000 nodes have been created and connected. At 3,000 simulation steps, the probe message is generated and forwarded at every simulation step. Because the number of subnetworks does not affect the RW algorithm, we plotted only one curve for each number of nodes. Conversely, subnetwork awareness plays a fundamental role in HR.

Both graphs show clearly that our HR-based algorithm outperforms the RW algorithm. For the two topologies, the HR-based algorithm requires about 4,000 message propagations to visit the whole network, whereas RW visits only 90 percent of the network

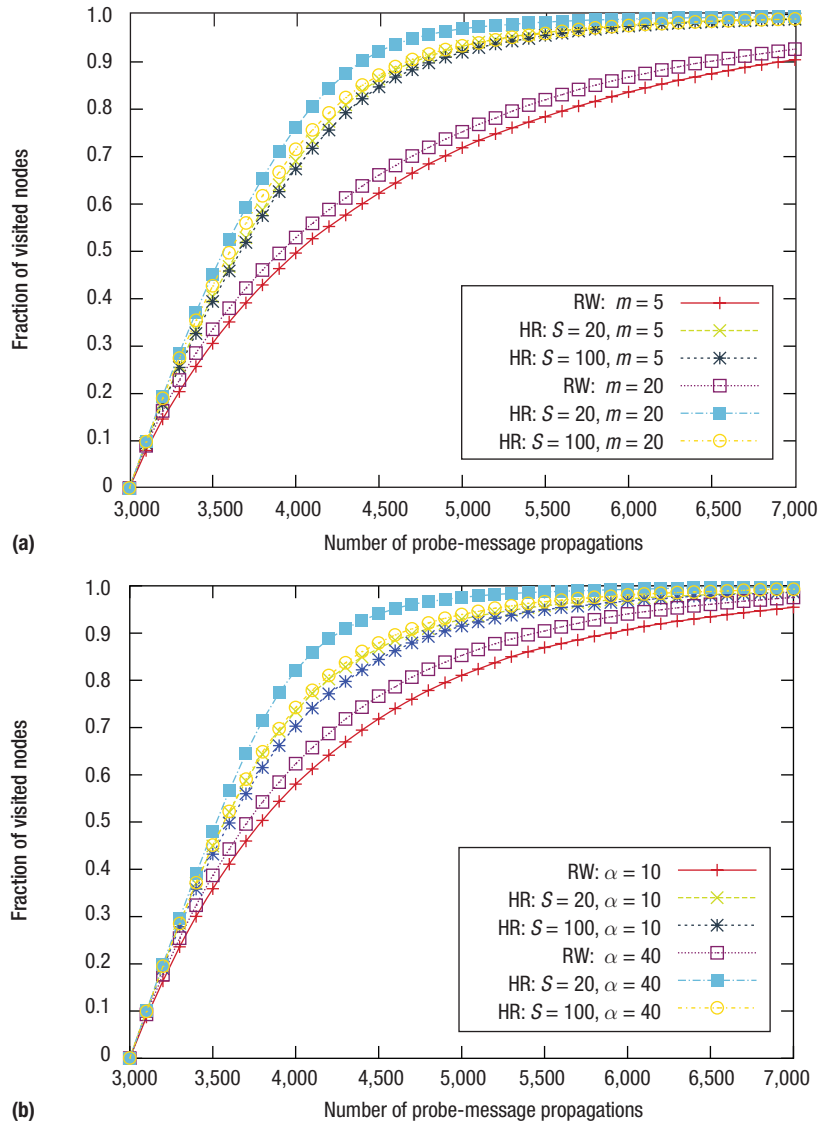


FIGURE 4. Fraction of visited nodes when the network topology is (a) scale free (Barabási-Albert [BA] model) or (b) the Erdős-Rényi (ER) model. S is the number of subnetworks, m is the initial number of connections of each node for the BA model, and α is the mean node degree for the ER model.

with the same number of propagations. The difference is more evident with the BA model, because with the HR method, the probe message visits every hub just once, whereas with the RW method, the probe message must visit hubs frequently. On the other hand, neither RW nor our HR method exploits hubs conveniently, so the performance difference is less dramatic with the ER topology, which assumes no hubs and thus more fairly distributes node degree values.

MESSAGE ROUTING

Message routing also benefits from collective self-awareness. To illustrate, consider again the network in Figure 2 and assume that node 4.2 must send a message to node 9.6. If routing tables were filled with only local information (only node 4.2's direct neighbors), routing would be quite inefficient. HR-based routing exploits collective self-awareness, enabling the network to find the route more quickly. Node 4.2 knows that

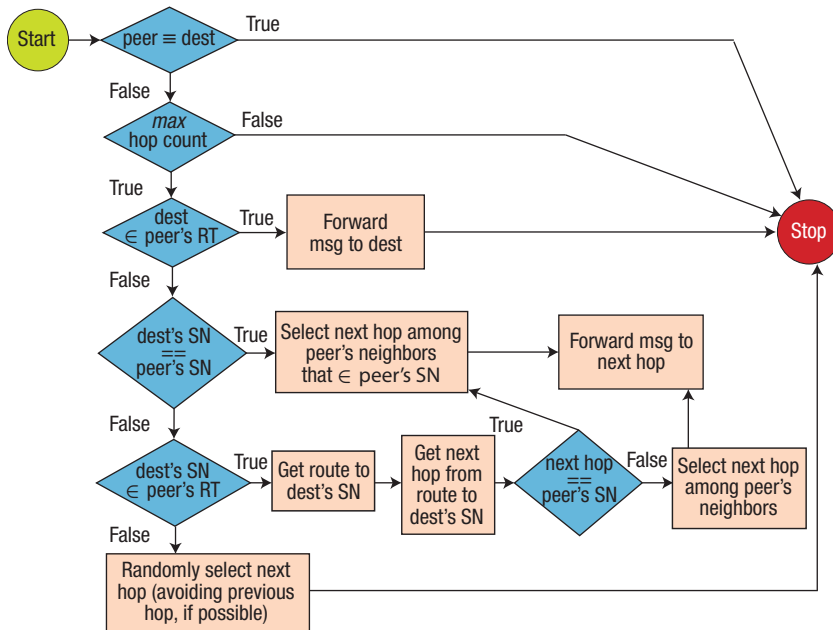


FIGURE 5. HR-based routing algorithm. Nodes build collective awareness by exchanging HR information with both direct neighbors and neighbors of direct neighbors. RT: routing table; SN: subnetwork.

NET9 is reachable through NET6, whose node 6.1 is directly reachable. Thus, node 4.2 sends the message to node 6.1. Figure 5 shows the flow of this routing algorithm.

HR-based routing—in which nodes populate routing tables with information about subnetworks—is suitable for both intra- and inter-domain scenarios. Compared to the two main intradomain routing classes—link-state and distance-vector¹¹—HR-based routing has two main advantages. First, unlike link-state routing, nodes need not know the whole network topology. Second, unlike distance-vector routing, nodes build collective awareness by exchanging recursive and hierarchical information not only with direct neighbors but also with neighbors of neighbors. Because of collective awareness, messages can be routed within the same subnetwork or from one subnetwork to another. In that sense, HR-based routing enables novel architectures, such as the Unified Architecture for Interdomain Routing proposed in RFC 1322 (www.rfc-editor.org/rfc/rfc1322.txt).

Simulation results

To evaluate the HR-based routing strategy’s success rate and average route length, we simulated an application with different networks, each with a thousand nodes. As a baseline, we also simulated a no-HR routing strategy in which nodes do not populate routing tables with subnetwork information but rather only keep a trace of direct neighbors and neighbors of neighbors. The no-HR strategy is similar to distance-vector routing, but it does not manipulate vectors of distances to other nodes in the network.

As Figure 6 shows, HR-based routing outperformed no-HR when the average node degree was suitably high. Interestingly, with low node degree values and the ER model topology, HR-based routing performed more poorly. Overall, however, with HR-based routing, a small increase in average node degree corresponded to a large performance increase.

Alternative scenario

Another network scenario that might benefit from HR-based message

routing is *n*-layered trees of subnetworks. For example, suppose that NET_x is the union of several subnetworks: NET_x = ∪_i NET_{x.i}.


In this case, for a node that belongs to another subnetwork NET_y such that there is no intersection between NET_x and NET_y, it is sufficient to know how to reach NET_x to send a message to a node that belongs to NET_{x.i}. This notion extends to any value of *n* (not just 2), as Figure 7 illustrates.

Networking and computing research communities are already considering HR-based strategies, but much work remains, particularly in finding novel strategies to efficiently maintain information that enables the simultaneous application of HR. In addition to network exploration and message routing, distributed sensing, mapping, and geolocation systems can also benefit from HR-based strategies.

Our HR-based approach is a significant first step toward collective self-awareness and self-expression in distributed systems. Groups of specialized servers with HR-aware routing tables could enable novel and highly efficient decentralized job dispatching and load balancing. Such a scenario is particularly appealing when jobs are part of complex workflows, each of which involves a different type of specialized servers.

Load-balancing and -sharing policies can be either static, using only information about average system behavior, or dynamic, adaptive policies that use continuously updated system state information. In an adaptive policy, both transfer and location policies are key elements. The transfer policy determines whether a job

is to be processed locally or remotely, whereas the location policy determines the server to which a remotely executed job should be sent. Typically, transfer policies use thresholds to determine whether the server is heavily loaded; when the server exceeds that threshold, the policy initiates the transfer mechanism.

Suitably filled HR-based routing tables would effectively support the location policy. Indeed, if servers share descriptors of their own capabilities and those of the nodes or node groups they know, they can quickly fill their routing tables with foundational information for locating suitable remote-job execution servers. To be even more effective, such static information might also be enriched by dynamic state variables that represent, for example, CPU queue length and utilization. Such strategies can be the basis for greatly increasing system efficiency. 

REFERENCES

1. F. Faniyi et al., "Architecting Self-aware Software Systems," *Proc. 12th Working IEEE/IFIP Conf. Software Architecture (WICSA 14)*, 2014, pp. 91-94.
2. M. Amoretti, "A Modeling Framework for Unstructured Supernode Networks," *IEEE Comm. Letters*, vol. 16, no. 10, 2012, pp. 1707-1710.
3. V. Jacobson et al., "Networking Named Content," *Proc. 5th ACM Int'l Conf. Emerging Networking Experiments and Technologies (CoNEXT 09)*, 2009; <http://conferences.sigcomm.org/co-next/2009/papers/Jacobson.pdf>.
4. J. Touch et al., "A Dynamic Recursive Unified Internet Design (DRUID)," *Computer Networks*, vol. 55, no. 4, 2011, pp. 919-935.

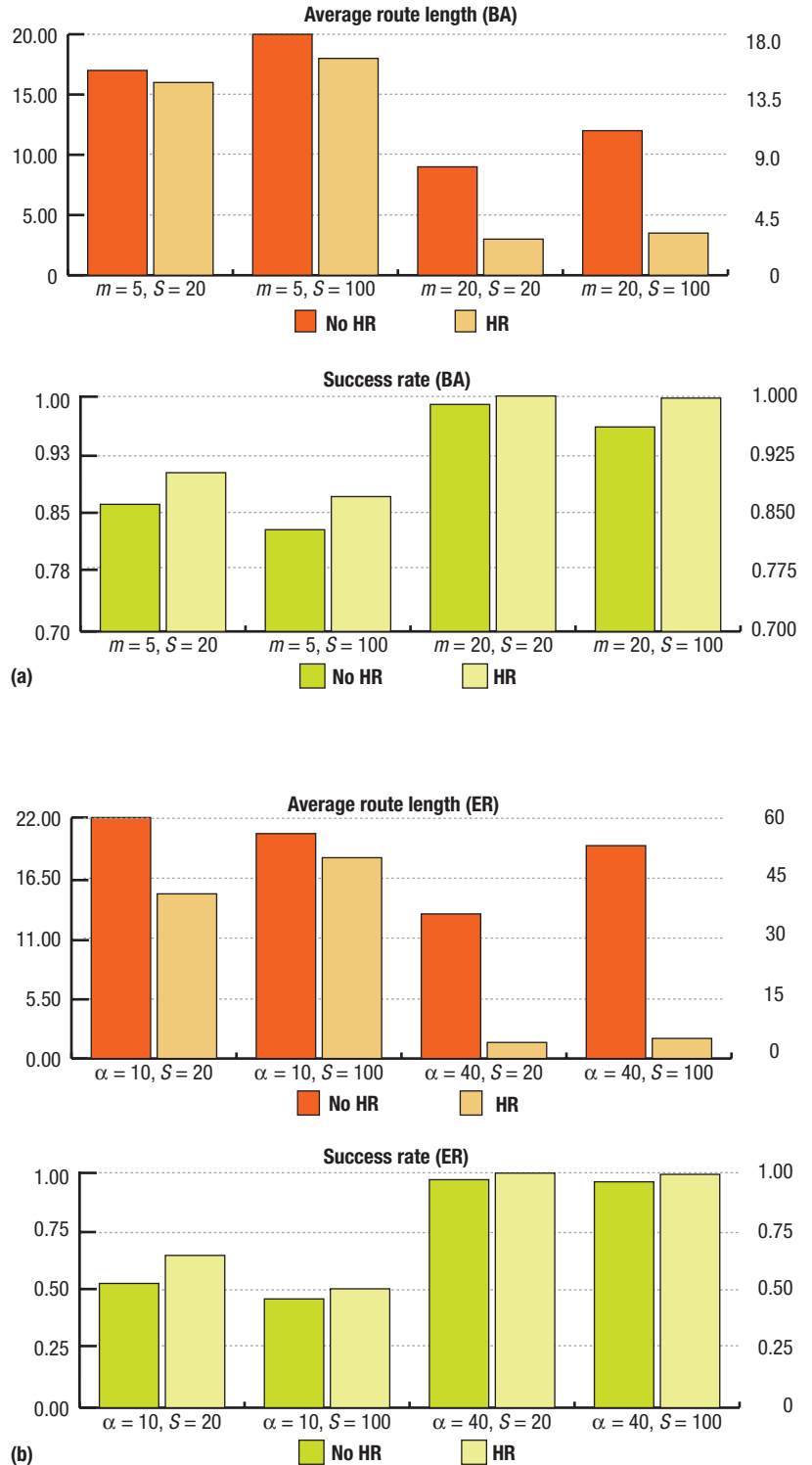


FIGURE 6. No HR versus HR-based routing in terms of average route length and success rate. (a) Scale-free (BA model) topology and (b) ER model topology. With the scale-free topology, HR-based routing outperformed no-HR, but with the ER model topology it performed worse overall. S is the number of subnetworks, m is the initial number of connections of each node for the BA model, and α is the mean node degree for the ER model. Mean values are based on 25 simulation runs.

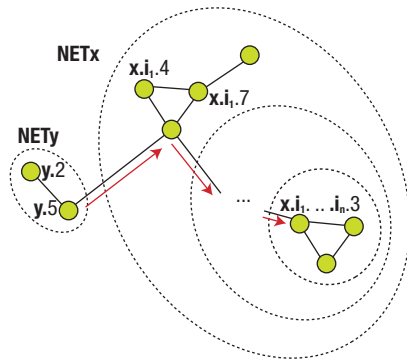


FIGURE 7. HR-based routing in an n-layered tree of subnetworks.

ABOUT THE AUTHORS

MICHELE AMORETTI is a postdoctoral research associate in the Department of Information Engineering at Università degli Studi di Parma, Italy. His research interests include modeling and simulation of large-scale distributed systems, high-performance autonomic computing, quantum computing, and wireless sensor networks. Amoretti received a PhD in information technologies from Università degli Studi di Parma. He is a member of IEEE. Contact him at michele.amoretti@unipr.it.

STEFANO CAGNONI is an associate professor of computer engineering at Università degli Studi di Parma. His research interests include soft computing, with an emphasis on the application of evolutionary computation and distributed intelligence to computer vision; pattern recognition; and robotics problems. Cagnoni received a PhD in biomedical engineering from Università di Firenze, Italy. He is a senior member of IEEE and member of the ACM Special Interest Group on Genetic and Evolutionary Computation (SIG EVO). Contact him at cagnoni@ce.unipr.it.

5. R. Van Meter, "Quantum Networking and Internetworking," *IEEE Networks*, vol. 26, no. 4, 2012, pp. 59–64.
6. A. Morin, "Self-Awareness Part 1: Definition, Measures, Effects, Functions, and Antecedents," *Social and Personality Psychology Compass*, vol. 5, no. 10, 2011, pp. 807–823.
7. M. Mitchell, "Complex Systems: Network Thinking," *Artificial Intelligence*, vol. 170, no. 18, 2006, pp. 1194–1212.
8. G. Cabri et al., "Self-Expression and Dynamic Attribute-Based Ensembles in SCEL," *Proc. Int'l Symp. Leveraging Applications Formal Methods, Verification, and Validation (ISoLA 14)*, LNCS 8802, 2014, pp. 147–163.
9. A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, 1999, pp. 509–512.
10. G. Siganos et al., "Lord of the Links: A Framework for Discovering Missing Links in the Internet Topology," *IEEE/ACM Trans. Networking*, vol. 17, no. 2, 2009, pp. 391–404.
11. F. Kurose and K.W. Ross, *Computer Networking: A Top-Down Approach*, Addison-Wesley, 2012.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

Computing
in SCIENCE & ENGINEERING

Subscribe today for the latest in computational science and engineering research, news and analysis, CSE in education, and emerging technologies in the hard sciences.

www.computer.org/cise