Wireless networks that fix their own broken communication links may speed up their widespread acceptance.

Self-Healing Networks

ROBERT POOR, CLIFF BOWMAN,
CHARLOTTE BURGESS AUBURN,
EMBER CORPORATION

The obvious advantage to wireless communication over wired is, as they say in the real estate business, location, location, location. Individuals and industries choose wireless because it allows flexibility of location—whether that means mobility, portability, or just ease of installation at a fixed point. The challenge of wireless communication is that, unlike the mostly error-free transmission environments provided by cables, the environment that wireless communications travel through is unpredictable. Environmental radio-frequency (RF) “noise” produced by powerful motors, other wireless devices, microwaves—and even the moisture content in the air—can make wireless communication unreliable.

Despite early problems in overcoming this pitfall, the newest developments in self-healing wireless networks are solving the problem by capitalizing on the inherent broadcast properties of RF transmission. The changes made to the network architectures are resulting in new methods of application design for this medium. Ultimately, these new types of networks are both changing how we think about and design for current applications and introducing the possibility of entirely new applications.

Self-Healing Networks

To capitalize on the benefits of wireless and to compensate for its challenges, much research and development to date has been focused on creating reliable wireless networks. Various approaches have been tried, but many wireless networks follow the traditional wired models and are manually configurable. This means that to join a network, a particular “node,” or transceiver-enabled device, must be programmed to direct its communications to another particular node—often a central base station. The challenge here is that if the node loses contact with its designated peer, communication ends.

To compensate for this possibility, a small army of businesses has formed that will complete exhaustive (and expensive) RF site surveys to determine the optimal placement of nodes in a particular space. Unfortunately, sometimes even this step is not enough to ensure reliability, as the character of the environment can change from day to day. This is especially true in industrial environments and has led some early adopters of wireless technology to declare the whole medium to be useless for their purposes. They may change their minds about this fairly soon, however.

The most promising developments are in the area of self-healing wireless networks. Often referred to as ad hoc networks, they are decentralized, self-organizing, and automatically reconfigure without human intervention in the event of degraded or broken communication links between transceivers. [See “Routing in Ad Hoc Networks of Mobile Hosts,” by David B. Johnson, from the *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, December 1994.]

These networks may have bridges or gateways to other networks such as wired Ethernet or 802.11, the strength of their architecture is that they do not require a base station or central point of control. It is these purely wireless decentralized networks we are addressing here.

Self-healing ad hoc networks have a decentralized architecture for a variety of reasons. The first, and perhaps the least expected, is historical. The earliest provider of funds for research in this area was the U.S. military. The requirements of the Defense Advanced Research Projects Agency (DARPA) SensIT program focused on highly redundant systems, with no central collection or control point. This makes sense even in the civilian world. Centralized networks, while optimized for throughput functions, risk a single point of failure and limit network scalability. Decentralized networks, in which each node acts as both an endpoint and a router for other nodes, naturally increase the redundancy of the network and open up the possibilities for network scaling as well. These attributes make for an attractive base on which to build self-healing network algorithms.

Automated network analysis through link and route discovery and evaluation are the distinguishing features of self-healing network algorithms. Through discovery, networks establish one or more routes between the originator and the recipient of a message. Through evaluation, networks detect route failures, trigger renewed discovery, and—in some cases—select the best route available for a message. Because discovery and route evaluation consume network capacity, careful use of both processes is important to achieving good network performance.

DISCOVERY AND ROUTING

The work done to date on discovery and routing in self-healing networks divides along a few lines, although these lines are often blurred after the first classification. As a general rule, however, wireless self-healing networks have proactive or on-demand discovery, and single-path or dynamic routing. These characteristics affect network latency, throughput, resource needs, and power consumption in varying amounts, with reference as well to the particular applications running over them.

Proactive discovery. Proactive-discovery networks configure and reconfigure constantly. They assume that link breakages and performance changes are always happening, and they are structured to continuously discover and reinforce optimal linkages. Proactive discovery occurs when nodes assume that all routes are possible and attempt to “discover” every one of them. The Internet is like this in the sense that it is possible, at least in principle, to route from any station to any other station once you know its IP address, without adding any new information to the tables in routers along the path.

This is a great strategy for the Internet, but it is usually considered impractical for an embedded network because

of its limited resources. Proactive discovery networks are “always on;” thus, a significant amount of traffic is generated and bandwidth occupied to keep links up to date. Also, because the network is always on, conserving power is more difficult.

On-demand discovery. On-demand discovery, in contrast, establishes only the routes that are requested by higher-layer software. On-demand discovery networks are only “on” when called for. This allows nodes to conserve power and bandwidth and keeps the network fairly free of traffic. If, between transmissions, the link quality between nodes has degraded, however, on-demand networks can take longer to reconfigure and, thus, to deliver a message.

Once routes have been established, they must generally be maintained in the presence of failing equipment, changing environmental conditions, interference, etc. This maintenance may also be proactive or on-demand.

Single-path routing. As for routing, network algorithms that choose single-path routing, as the name suggests, single out a specific route for a given source-destination pair. Sometimes, the entire end-to-end route is predetermined. Sometimes, only the next “hop” is known. The advantage of this type of routing is that it cuts down on traffic, bandwidth use, and power use. If only one node at a time needs to receive the packet, others can stop listening after they hear that they’re not the recipient.

The pitfall is non-delivery. If any links along the predetermined route are degraded or broken, the packet will not get through. Depending on the maintenance scheme, it can take a long time for the network to recover enough to build a new route.

Dynamic routing. Rather than ignoring the broadcast nature of the wireless medium, dynamic routing takes advantage of it. Messages are broadcast to all neighbors and forwarded according to a “cost-to-destination” scheme. Messages act as multiple objects rolling downhill toward their ultimate destination. Although this type of routing takes advantage of multiple redundant routes from originator to destination, it can also generate a lot of traffic on the network. Without modification, it can result in messages traveling in endless loops, jamming up the network.

These definitions may seem fairly rigid, but in practical implementation many self-healing networks take advantage of a blending of these characteristics. Often, with the more open-ended algorithms, applications can be developed to be more restrictive than the underlying network architecture. Research institutions, standards bodies, and corporations are building best-of-breed technologies

out of all these options, balancing the requirements of latency, bandwidth, memory, and scalability in different ways.

SELF-HEALING TECHNOLOGIES

AODV. Ad hoc On-demand Distance Vector (AODV) routing is, as its name states, an on-demand discovery protocol. [For more information, see “Ad hoc On-Demand Distance Vector Routing,” by Charles E. Perkins and Elizabeth M. Royer, from the *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.] It will discover routes to destinations only as messages need to be sent, although it bends this rule a bit by using periodic proactive “Hello” messages to keep track of neighboring nodes.

As for its routing scheme, AODV is actually a combination of dynamic and single-path routing (see Figure 1). It uses the cost-based gradients of dynamic routing to discover routes, but thereafter suppresses redundant routes by choosing a specific “next-hop” node for a particular destination. Thus, even if other nodes are within range, the chosen “next-hop” node is the one that receives the message. This choice is made to save memory and bandwidth, and comes at the cost of higher latency in the face of changing RF environment conditions and mobility of nodes, as it takes time to recover from broken routes.

DSR. Another on-demand discovery example, Dynamic Source Routing (DSR) is actually a single-path routing scheme, as shown in Figure 2. [See “Dynamic Source Routing in Ad Hoc Wireless Networks,” by David B. Johnson and David A. Maltz, *Mobile Computing*, Kluwer Academic Publishers, 1996.] DSR is a source-routing scheme, where the entire end-to-end route is included in the header of the message being sent. Those routes are developed using dynamic discovery messages, however, and multiple possible routes are stored in case of link breakage.

The “key advantage to source routing is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they forward, since the packets themselves already contain all the routing decisions,” according to Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva in “A Performance Comparison of Multi-Hop Wireless Ad-Hoc Network Routing Protocols,” [*Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom ’98)*, October 25-30, 1998, Dallas, TX]. This choice reduces memory usage for tables, as well as administrative traffic, because this,

Self-Healing

"coupled with the on-demand nature of the protocol, eliminates the need for the periodic route advertisement and neighbor detection packets present in other protocols," according to Broch et al.

The main drawback to DSR is its increased packet overhead—because the whole route needs to be contained in the packet. For nodes that need to communicate to many other nodes, there is also a high memory penalty, as that node needs to store entire routes and not just a next hop or cost.

GRAd. Gradient routing in ad hoc networks (GRAd) is our first example of totally dynamic routing, shown in Figure 3. [See "Gradient Routing in Ad Hoc Networks," by Robert D. Poor, MIT Media Laboratory, 2000.] It is an on-demand discovery scheme as well. GRAd's routing capitalizes on the potential availability of redundant routes from

originator to destination nodes to optimize for lowest latencies. To reduce network traffic once the message has made it to the destination, GRAd suppresses message proliferation loops by returning an acknowledgment. The maintenance of multiple sets of routes adds memory cost and network traffic, but the return is an increase in both reliability and speed of message delivery.

Cluster-Tree. The Cluster-Tree algorithm ["Cluster Tree Network," by Ed Callaway, IEEE p.802.15 Wireless Personal Area Networks, 2001] is the most significantly different protocol in both discovery and routing from the other algorithms described here. To initialize the network, one of the nodes is designated as the "root" of the tree. It assigns network addresses to each of its neighbors. These neighbors then determine the next layer of addresses, and so on. Routing is rigidly defined, and messages by default go up to the closest jointure and back down branches to get to their destinations. Discovery in the Cluster-Tree algorithm is proactive. Periodic "Hello" messages follow the setup of the network to ensure that links remain unbroken or to rearrange the tree if damage has occurred.

The great benefit to this algorithm is a small memory footprint, as a node has to remember only the addresses of its neighbors. This type of system is a good choice for smaller applications where all data needs to flow to a single sink. For that small footprint, however, it sacrifices speed across the network (some nodes may have to go all the way to the root to contact another node that is physically close by) and scalability.

BUILDING APPLICATIONS FOR SELF-HEALING NETWORKS

The significant compromise to the robustness of any self-healing wireless network versus a wired network, or a centralized wireless network, is the increased latency and loss of throughput to the overhead costs of network maintenance and the inherent costs of store-and-forward messaging. To recover some of that lost network performance, developers need to focus on designing extremely efficient applications—those

that take advantage of the processing power available in wirelessly enabled endpoints and that tailor the transport layer of the network to the needs of the particular application. This results in the need to do more careful code building and can mean a steeper learning curve for those wanting to use self-healing technologies. Ultimately, however, the results will be a proliferation of low-power, low-latency, highly scalable applications that will transform the technology landscape yet again.

Network application design—particularly the design of the “simple” applications most likely to run on severely resource-constrained hardware—often exhibits an unfortunate inertia. In the earliest days of embedded digital networks, developers used bus-sharing strategies such as query-response and token-passing to control traffic. Perhaps by habit, some developers try to employ these strategies even when a viable MAC layer is in place. Unfortunately, this redundant traffic control adds overhead and reduces network capacity; when the selection of a self-healing network is already squeezing network capacity, developers designing these networks should question the need to directly control access to it. These kinds of trade-off choices are the hallmark of design for self-healing networks.

One especially useful strategy for avoiding unnecessary overhead is to decentralize tasks within the network. Digital communications assume some degree of processing power at each node, and using that power to handle tasks

in a distributed way often requires no hardware changes. Some types of processing are cheaper than sending data in self-healing networks—even the simplest devices can compare data against thresholds, so it is possible to limit messaging to cases where there is something interesting to say. Rather than a periodic temperature report, for example, a temperature-sensing device can be programmed to report “exceptions,” conditions under which the temperature falls outside a prescribed range. This kind of

Self-Healing Networks

exception-based or “push” messaging can greatly reduce traffic in a network, leaving more capacity for communication. Depending on the amount of processing power available at the endpoint and the complexity of the application, a significant portion of the data processing and analysis needed for an application can be done before the data ever leaves the endpoint. Developers need to evaluate their applications carefully to discover how useful this strategy will be for them.

Developers encounter another challenge at the point where data leaves the endpoint. In extremely resource-constrained applications, using extraordinary methods may be necessary to satisfy application specifications. As a concrete example, consider the conscious relaxation of encapsulation in the classical Open System Interconnection (OSI) model in a network using dynamic route selection. Developers need to ask, “Is pure separation of layers appropriate here?”

According to good programming practice and the OSI model, developers should encapsulate routing tasks in the network layer and transmission and reception tasks in the physical layer. Any interaction between these layers should be indirect and mediated by the network layer. Unfortunately, efficient dynamic route selection often depends upon immediate access to physical data such as signal strength or correlator error rate.

In this case, performance can suffer if there are artificial barriers to this interaction. Ultimately, this suggests that the OSI networking model may need to change to suit the characteristics of these new networks, whose importance is growing all the time and should not be underestimated.

THE NEXT 18 MONTHS...

Thanks to Moore’s Law, the cost of radio frequency integrated circuits (RFICs) and microcontrollers is falling

rapidly, making possible the manufacture of very low-cost wireless communication nodes. Commercial wireless chipsets are available for less than \$5 (with a typical total module cost of \$20), less than the cost of the connectors and cables that are the hallmark of wired network systems—and that cost is still decreasing.

When these low-cost wireless nodes are provisioned with self-healing networking algorithms, the result is a network that is simultaneously inexpensive to manufacture and easy to deploy. Because the network is wireless, you don’t need to drill holes and pull cables. And because it is self-organizing and self-healing, you don’t need to be an RF expert to deploy a reliable network; the network takes care of itself.

So the stage is set. We have at hand a new kind of network that can be mass-produced inexpensively and deployed with no more effort than bringing one wireless node within range of another. What kinds of applications can we expect from these self-healing wireless networks within the next 18 months? And what changes can we expect in the long term?

Expect to see a profusion of sensor networks, characterized by a large quantity of nodes per network, where each node produces low-bit-rate, low-grade data that is aggregated and distilled into useful information. Using wireless sensor networks, bridges and buildings will be outfitted with strain gauges to monitor their structural health. Air-handling systems in airport terminals and subway tunnels will be equipped with detectors for toxic agents, giving an early warning in case dangerous substances are released into the air. Orchards and vineyards will be peppered with tiny environmental monitors to help determine the optimal irrigation and fertilization schedule and to raise an alert if frost is imminent.

Expect to see numerous wire replacement applications in buildings and in factories. Within buildings, lighting and HVAC (heating, ventilation, air conditioning) systems will use wireless connections to reduce the cost of installation and remodeling. The ZigBee Alliance [www.zigbee.org] is an industry alliance developing profiles on top of the IEEE 802.15.4 Wireless Personal Area Network devices [www.ieee802.org/15/pub/TG4.html]; some of the first profiles being developed are specifically for lighting and HVAC applications.

Expect to see a variety of asset management applications, where commercial goods are equipped with inexpensive wireless nodes that communicate information about their state and provenance. Imagine a wireless maintenance log attached to every jet turbine in a giant warehouse, making it possible to do a live query of the

warehouse to learn the presence, current state, and maintenance history of each turbine. This not only reduces the time required to locate parts, but also is indispensable for rapid product recalls.

...AND BEYOND

In the longer term, individual wireless networks will interconnect to offer functionality above and beyond their original intent. In the office building, wireless light switches and thermostats will cooperate with motion detectors and security sensors and, over time, form a detailed model of the usage patterns of the building. The building can “learn” to anticipate ordinary events and to flag abnormal events, resulting in lower energy costs and more effective security systems. In the case of a fire or other emergency, the building can alert rescue personnel as to which rooms are occupied and which are empty.

Imagine a city in which each lamppost is outfitted with a simple go/no-go sensor to indicate if its light is working properly or not. From the outset, this system can save the city substantial revenue by automatically identifying which lamps

need replacing and which are wasting energy. Imagine that each bus in the city is outfitted with a wireless node. As a bus drives down the street, its position is relayed through the wireless network to a display at the next bus kiosk, informing the waiting passengers how long they must wait for the next bus.

The important common factor in all these applications is that humans are barely involved: The devices, sensors, and networks must function without human intervention. As David Tennenhouse eloquently stated in his visionary article, “Proactive Computing,” [*Communications of the ACM (CACM)*, May 2000, Volume 43, #5], widespread adoption of networked devices can’t happen until humans “get out of the loop,” simply because there will be too many devices for us to attend to. It is the autonomous nature of self-organizing, self-healing networks that makes these applications possible.

THE CHALLENGES

Self-healing networks are designed to be robust even in environments where individual links are unreliable, making them ideal for dealing with unexpected circumstances, such as public safety systems. The dynamic

nature that gives these networks their self-healing properties, however, also makes them difficult to test. Even after multiple deployments and thorough simulation, it’s difficult to predict how these systems will work (or fail) in actual emergencies. Can wireless, self-healing networks be trusted with anything but the most trivial of data?

As with any new technology that touches our daily lives, perhaps the biggest challenge is anticipating the sociological effects of a very networked world. Trusted technologies will need to address challenges of privacy in network and data security, in regulatory issues, and in civil liberties. Buildings that know where we are will keep us safe in an emergency, but do they infringe on our rights of privacy and freedom?

Ultimately, if self-healing networks can provide truly useful services to a wide range of applications at a good

cost, all the challenges facing them—both social and technological—will be readily solved. Remembering that the best uses for technologies are often difficult to predict (after all, e-mail was an afterthought for the developers of the Internet), we can be fairly certain that the killer

app for self-healing networks is out there, waiting to be developed. □

ROBERT POOR founded Ember Cooperation in 2001. As a doctoral student at the MIT Media Lab, he developed Embedded Networking, a new class of wireless network that enables inexpensive, scalable, easily deployed connections for common manufactured goods. He has accrued more than 25 years of technical and management-level experience from Silicon Valley. He received a patent for self-organizing networks in 2000 and was awarded his Ph.D. from MIT in 2001.

CHARLOTTE BURGESS AUBURN is the marketing manager for Ember Corporation. She holds a bachelor's degree from Oberlin College and a master's degree from Tufts University. She has professional experience in creative production with the MIT Media Laboratory.

CLIFF BOWMAN is an applications engineer at Ember Corporation. He has been developing embedded solutions for industrial, defense, and automotive applications since 1991.

© 2003 ACM 1542-7730/03/0500 \$5.00

BIO