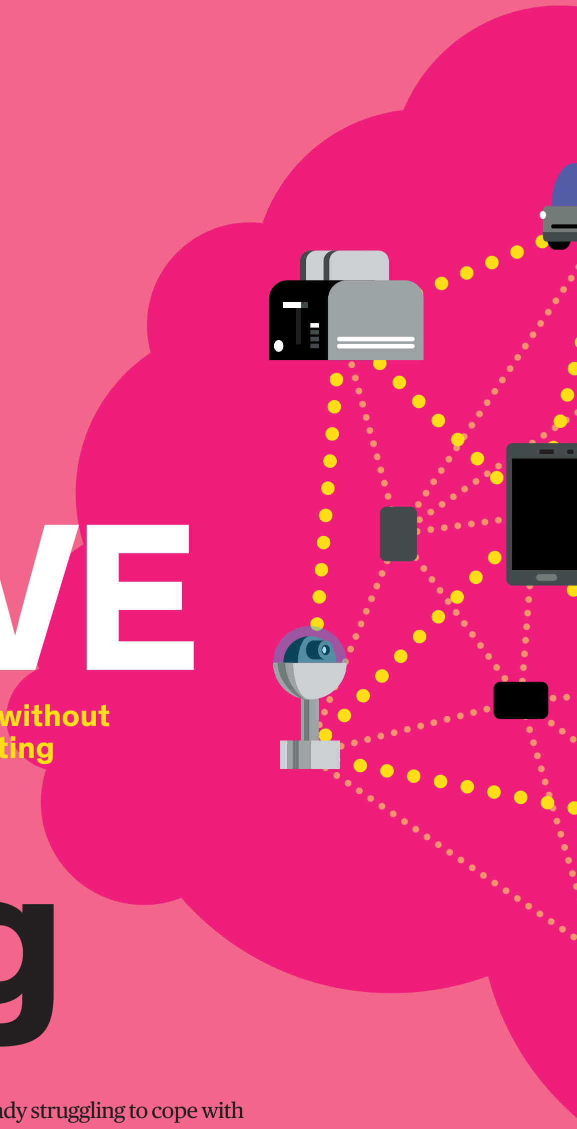
The COGNITIVE NET Is Coming

The Internet will break down without new biologically inspired routing

By Antonio Liotta

Illustrations by L-Dopa



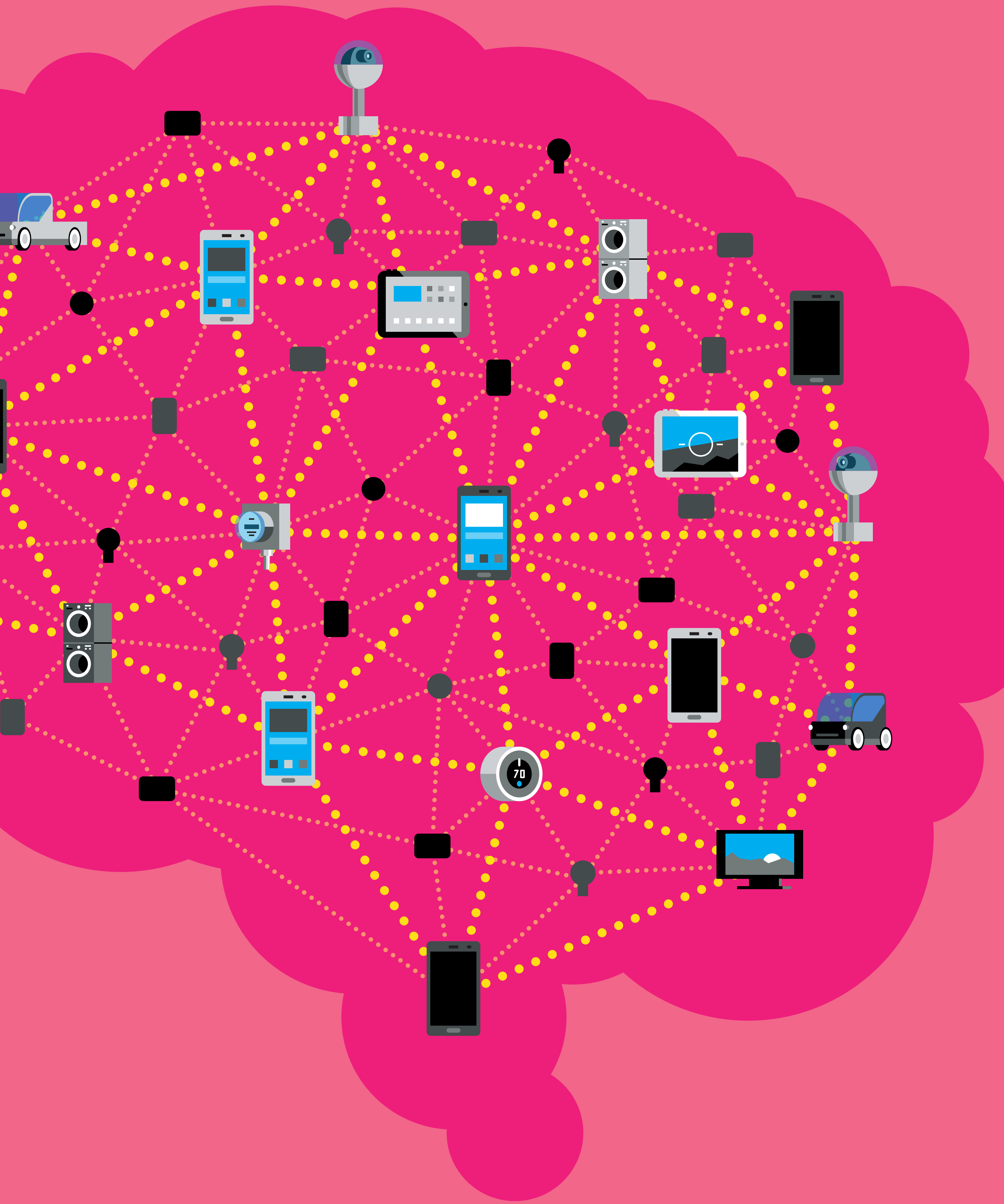
PERHAPS AS EARLY AS THE END OF THIS DECADE, our refrigerators will e-mail us grocery lists. Our doctors will update our prescriptions using data beamed from tiny monitors attached to our bodies. And our alarm clocks will tell our curtains when to open and our coffeemakers when to start the morning brew.

By 2020, according to forecasts from Cisco Systems, the global Internet will consist of 50 billion connected tags, televisions, cars, kitchen appliances, surveillance cameras, smartphones, utility meters, and whatnot. This is the Internet of Things, and what an idyllic concept it is.

But here's the harsh reality: Without a radical overhaul to its underpinnings, such a massive, variable network will likely create more problems than it proposes to solve. The reason? Today's Internet just isn't equipped to manage the kind of traffic that billions more nodes and diverse applications will surely bring.

In fact, it's already struggling to cope with the data being generated by ever-more-popular online activities, including video streaming, voice conferencing, and social gaming. Major Internet service providers around the world are now reporting global latencies greater than 120 milliseconds, which is about as much as a Voice over Internet Protocol connection can handle. Just imagine how slowly traffic would move if console gamers and cable television watchers, who now consume hundreds of exabytes of data off-line, suddenly migrated to cloud-based services.

The problem is not simply one of volume. Network operators will always be able to add capacity by transmitting data more efficiently and by rolling out more cables


70

and cellular base stations. But this approach is increasingly costly and ultimately unscalable, because the real trouble lies with the technology at the heart of the Internet: its routing architecture.

Information flows through the network using a four-decade-old scheme known as packet switching, in which data is sliced into small envelopes, or packets. Different packets may take different routes and arrive at different times, to be eventually reassembled at their destination. Routers, which decide the path each packet will take, are "dumb" by design. Ignorant of a packet's origin and the bottlenecks it may encounter down the line, routers treat all packets the same way, regardless of whether they contain snippets of a video, a voice conversation, or an e-mail.

This arrangement worked superbly during the Internet's early days. Back then most shared content, including e-mail and Web browsing, involved small sets of data transmitted with no particular urgency. It made sense for routers to process all packets equally because traffic patterns were mostly the same.

That picture has changed dramatically over the past decade. Network traffic today consists of bigger data sets, organized in more varied and complex ways. For instance, smart meters produce energy data in short, periodic bursts, while Internet Protocol television (IPTV) services generate large, steady streams. New traffic signatures will emerge as new applications come to market, including connected appliances and other products we haven't yet imagined. Basic packet switching is just too rigid to manage such a dynamic load.

So it's time we gave the Internet some smarts, not simply by making incremental improvements but by developing an entirely new way to transport data. And engineers are turning to nature for inspiration.

The Path to INTELLIGENT Routing

The future Internet will need smarter routing algorithms to handle diverse data flows and prevent failures. Although there are no tried-and-true solutions yet, early designs might follow an architecture like this one.

1 **A ROUTING DEVICE** can be any network node, such as a phone, a television, a car, a kitchen appliance, an environmental sensor, or some gadget yet to be invented. Proximal devices form "mesh networks" that off-load some traffic from the core network and bring Internet service to remote places.

2 **THE ROUTING AND FORWARDING ENGINES** determine the best pathways to get data packets to their destinations and queue them for transmission. (These engines are already built into today's "dumb" routers, but in the future they could exist as software applications rather than separate pieces of hardware.)

3 **THE AUTONOMIC CONTROLLER** directs the routing and forwarding engines by following the MAPE loop: It *monitors* internal sensor data and signals from other nodes, *analyzes* that information, *plans* a response, and *executes* it. Neighboring devices coordinate their actions in real time through control signals.

4 **THE COGNITIVE ENGINE** helps the router adapt to unforeseen changes by following the OOPDAL loop: It *observes* the environment, *orients* the system by prioritizing tasks, *plans* options, *decides* on a plan, *acts* on it, and *learns* from its actions. By sharing knowledge, devices spread intelligence across the Internet.

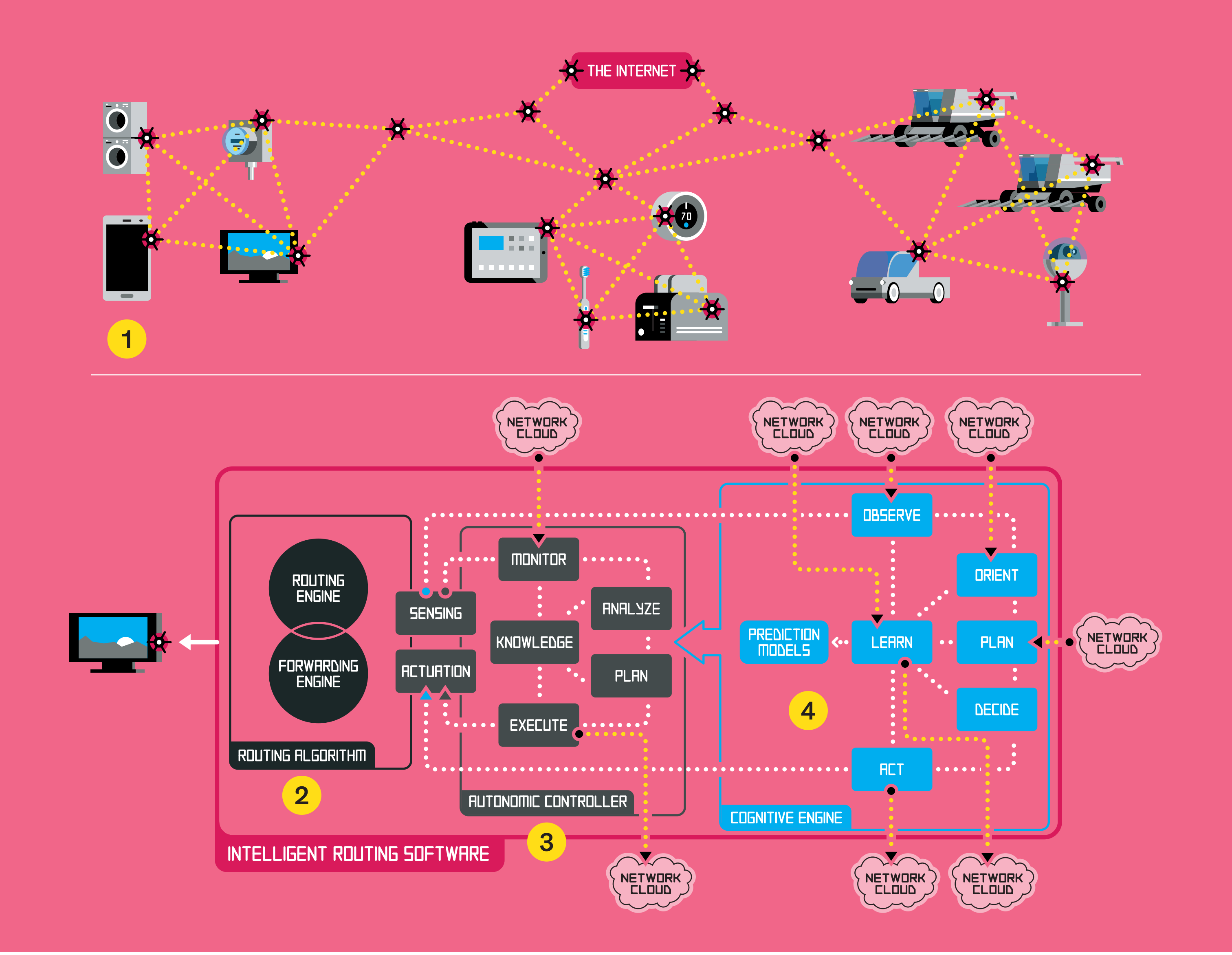
Millions of years of evolution have resulted in biological networks that have devised ingenious solutions to the hardest network problems, such as protecting against infectious agents and adapting to failures and changes. In particular, the human brain and body are excellent models for building better data networks. The challenge, of course, is in figuring out how to mimic them (see sidebar, "Networking Lessons From the Real World").

To understand why the packet-switched Internet must be replaced with a more intelligent system, first consider how today's network is structured. Say, for example, you want to watch a YouTube clip. For the video data to stream from Google's server to your smartphone, the packets must pass through a hierarchy of subnetworks. They start at the outermost reaches of the Net: the access network, where terminals such as phones, sensors, servers, and PCs link up. Then the packets move through regional networks to the core network, or backbone. Here, dense fiber-optic cables ferry traffic at high speeds and across vast distances. Finally, the packets make their way back to the access network, where your smartphone resides.

Routers send each incoming packet along the best available route through this hierarchy. It works like this: Inside each router, a collection of microchips called the routing engine maintains a table that lists the pathways to possible destinations. The routing engine continually updates this table using information from neighboring nodes, which monitor the network for signs of traffic jams. When a packet enters the router's input port, another set of chips—the forwarding engine—reads the packet's destination address and queries the routing table to determine the best node to send the packet to next. Then it switches the packet to a queue, or buffer, where it awaits transmission. The router repeats this process for each incoming packet.

There are several disadvantages to this design. First, it requires a lot of computational muscle. Table queries and packet buffering consume about 80 percent of a router's CPU power and memory. And it's slow. Imagine if a mail carrier had to recalculate the delivery route for each letter and package as it was collected. Routers likewise ignore the fact that many incoming packets may be headed for the same terminal.

Routers also overlook the type of data flow each packet belongs to. This is especially problematic during moments of peak traffic, when packets can quickly pile up in a router's buffer. If more packets accumulate than the buffer can hold, the router discards



excess packets somewhat randomly. In this scenario, a video stream—despite having strict delivery deadlines—would experience the same packet delays and losses as an e-mail. Similarly, a large file transfer could clog up voice and browsing traffic so that no single flow reaches its destination in a timely manner.

And what happens when a crucial routing node fails, such as when a Vodafone network center in Rotterdam, Netherlands, caught fire in 2012? Ideally, other routers will figure out how to divert traffic around the outage. But often, local detours just move the congestion elsewhere. Some routers become overloaded with packets, causing more rerouting and triggering a cascade of failures that can take down large chunks of the network. After the Vodafone fire, 700 mobile base stations were out of commission for more than a week.

Routers could manage data flows more effectively if they made smarter choices about which packets to discard and which ones to expedite. To do this, they would need to gather much more information about the network than simply the availability of routing links. For instance, if a router knew it was receiving high-quality IPTV packets destined for a satellite phone, it might choose to drop those packets in order to prioritize others that are more likely to reach their destinations.

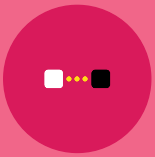
Ultimately, routers will have to coordinate their decisions and actions across all levels of the Internet, from the backbone to the end terminals, and the applications running on them. And as new user devices, services, and threats come on line in the future, the system will need to be smart enough to adapt.

The first step in designing a more intelligent Internet is to endow every connected computer with the ability to route data. Given the computational capabilities of today's consumer devices, there's no reason for neighboring smart gadgets to communicate over the core network. They could instead use any available wireless technology, such as Wi-Fi or Bluetooth, to spontaneously form "mesh networks." This would make it possible for any terminal that taps into the access network—tablet, television, thermostat, tractor, toaster, toothbrush, you name it—to relay data packets on behalf of any other terminal.

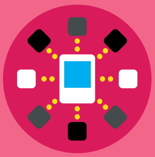
By off-loading local traffic from the Internet, mesh networks would free up bandwidth for long-distance services, such as IPTV, that would otherwise require costly infrastructure upgrades. These networks would also add routing pathways that bypass bottlenecks, so traffic could flow to areas where Internet access is now

Networking LESSONS From the Real World

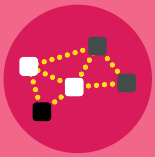
Internet engineers can learn a lot from biological and social networks



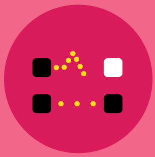
Keep pathways short, even in large networks.
EXAMPLES: Social relationships, gene regulation, neural networks in the brain
ADVANTAGES: When data can reach any destination in a small number of steps, latency stays low.



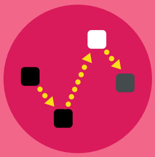
Only a small percentage of nodes should have many links.
EXAMPLES: Human sexual partners, scientific-paper citations
ADVANTAGES: Minimizing the number of hubs helps stop the spread of viruses and protects against attacks.



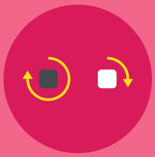
Weak links can be a good thing.
EXAMPLES: Some molecular structures
ADVANTAGES: Poor or transient links can help improve bad connections, dissipate disruptions, and bring network access to places where strong links can't be built.



Trade some speed for stability.
EXAMPLES: Traffic-control systems (including stoplights, yield signs, and speed limits)
ADVANTAGES: Controlling data flows helps prevent traffic spikes from causing network congestion or collapse.



Spread information through "gossip" rather than broadcast.
EXAMPLES: Rumors, viral videos
ADVANTAGES: Disseminating data as if it were gossip can be more efficient and less disruptive than broadcasting it.



Control and learn at different timescales.
EXAMPLES: Autonomic functions (such as breathing and digesting) versus cognition
ADVANTAGES: Real-time control lets nodes coordinate actions, while gradual learning helps the network evolve.

poor, extending cellular service underground, for example, and providing extra coverage during natural disasters.

But to handle data and terminals of many different kinds, routers (including the terminals themselves) need better methods for building and selecting data pathways. One way to engineer these protocols is to borrow tricks from a complex network that already exists in nature: the human autonomic nervous system.

This system controls breathing, digestion, blood circulation, body heat, the killing of pathogens, and many other bodily functions. It does all of this, as the name suggests, autonomously–without our direction or even our awareness. Most crucially, the autonomic nervous system can detect disturbances and make adjustments before these disruptions turn into life-threatening problems.

If all this sounds a little vague, consider the example of digestion. Say you've just eaten a big, juicy hamburger. To begin breaking it down, the stomach must secrete the proper amount of gastric juices. This might seem like a simple calculation: more meat, more juices. In fact, the parts of the brain that control this process rely on a smorgasbord of inputs from many other systems, including taste, smell, memory, blood flow, hormone levels, muscle activity, and immune responses. Does that burger contain harmful bacteria that must be killed or purged? Does the body need to conserve blood and fuel for more important tasks, such as running from an enemy? By coordinating many different organs and functions at once, the autonomic system keeps the body running smoothly.

By contrast, the Internet addresses a disturbance, such as a spike in traffic or a failed node, only after it starts causing trouble. Routers, servers, and computer terminals all try to fix the problem separately, rather than work together. This often just makes the problem worse–as was the case during the Vodafone fire.

A more cooperative Internet requires routing and forwarding protocols that behave more like the autonomic nervous system. Network engineers are still figuring out how best to design such a system, and their solutions will no doubt become more sophisticated as they work more closely with biologists and neuroscientists.

One idea, proposed by IBM, is the Monitor-Analyze-Plan-Execute (MAPE) loop, or more simply, the knowledge cycle. Algorithms that follow this architecture must perform four main tasks:

First, they *monitor* a router's environment, such as its battery level, its memory capacity, the type of traffic it's seeing, the number of nodes it's connected to, and the bandwidth of those connections.

Then the knowledge algorithms *analyze* all that data. They use statistical techniques to determine whether the inputs are typical and, if they aren't, whether the router can handle them. For example, if a router that typically receives low-quality video streams suddenly receives a high-quality one, the algorithms calculate whether the router can process the stream before the video packets fill its buffer.

Next, they *plan* a response to any potential problem, such as an incoming video stream that's too large. For instance, they may figure the best plan is to ask the video server to lower the stream's bit rate. Or they may find it's better to break up the stream and work with other nodes to spread the data over many different pathways.

Lastly, they *execute* the plan. The execution commands may modify the routing tables, tweak the queuing methods, reduce transmission power, or select a different transmission channel, among many possible actions.

A routing architecture like the MAPE loop will be key to keeping the Internet in check. Not only will it help prevent individual routers from failing, but by monitoring data from neighboring nodes and relaying commands, it will also create feedback loops within the local network. In turn, these local loops swap information with other local networks, thereby propagating useful intelligence across the Net.

It's important to note that there's no magic set of algorithms that will work for every node and every local network. Mesh networks of smartphones, for example, may operate best using protocols based on swarm intelligence, such as the system ants use to point fellow ants to a food source. Meanwhile, massive monitoring networks, such as "smart dust" systems made of billions of grain-size sensors, may share data much as people share gossip—a method that would minimize transmission power.

Autonomic protocols would help the Internet better manage today's traffic flows. But because new online services and applications emerge over the lifetime of any router, routers will have to be able to learn and evolve on their own.

To make this happen, engineers must turn to the most evolutionarily advanced system we know: human cognition. Unlike autonomic systems, which rely on predetermined rules, cognitive systems make decisions based on experience. When you reach for a ball flying toward you, for example, you decide where to position your hand by recalling previous successes. If you catch the ball, the experience reinforces your reasoning. If you drop the ball, you'll revise your strategy.

Of course, scientists don't know nearly enough about natural cognition to mimic it exactly. But advances in the field of machine learning—including pattern-recognition algorithms, statistical inference, and trial-and-error learning techniques—are proving to be useful tools for network engineers. With these tools, it's possible to create an Internet that can learn to juggle unfamiliar data flows or fight new malware attacks in a manner similar to the way a single computer might learn to recognize junk mail or play "Jeopardy!"

Engineers have yet to find the best framework for designing cognitive networks. A good place to start, however, is with a model first proposed in the late 1990s for building smart radios. This architecture is known as the cognition cycle, or the Observe-Orient-Plan-Decide-Act-Learn (OOPDAL) loop. Like the MAPE loop in an autonomic system, it begins with the *observation* of environmental conditions, including internal sensor data and signals from nearby nodes. Cognition algorithms then *orient* the system by evaluating and prioritizing the gathered information. Here things get more complex. For low-priority actions, the algorithms consider alternative *plans*. Then they *decide* on a plan and *act* on it, either by triggering new internal behavior or by signaling nearby nodes. When more-urgent action is needed, the algorithms can bypass one or both of the planning and decision-making steps. Finally, by observing the results of these actions, the algorithms *learn*.

In an Internet router, OOPDAL loops would run parallel to the autonomic MAPE loop (see illustration, "The Path to Intelligent Routing"). As the cognition algorithms learned, they would generate prediction models that would continually modify the knowledge algorithms, thereby improving the router's ability to manage diverse data flows. This interaction is akin to the way your conscious brain might retrain your arm muscles to catch a hardball after years of playing with a softball.

Network engineers are still far from creating completely cognitive networks, even in the laboratory. One of the biggest challenges is designing algorithms that can learn not only how to minimize the use of resources—such as processing power, memory, and radio spectrum—but also how to maximize the quality of a user's experience. This is no trivial task. After all, experience can be highly subjective. A grainy videoconference might be a satisfactory experience for a teenager on a smartphone, but it would be unacceptable to a business executive chatting up potential clients. Likewise, you might be more tolerant of temporary video freezes if you were watching a free television service than if you were paying for a premium plan.

Nevertheless, my colleagues and I at the Eindhoven University of Technology, in the Netherlands, have made some progress. Using a network emulator, or "Internet in a box," we can simulate various network conditions and test how they affect the perceived quality of different types of video streams. In our experiments, we have identified hundreds of measurable parameters to predict the quality of experience, including latency, jitter, video content, image resolution, and frame rate. Using new sensing protocols, terminals could also measure things like the type of screen someone's using, the distance between the screen and the user, and the lighting conditions in the room.

In collaboration with Telefónica, in Spain, we have created machine-learning algorithms that use many of these parameters to predict the quality of a user's experience when IPTV programs are streamed to different types of smartphones. These prediction models turned out to be remarkably accurate (having around a 90 percent agreement with user surveys), showing that it's possible to train networks to adapt to variable conditions on their own. In another study, we demonstrated that a network can quickly learn, through trial and error, the best bit rate for delivering a specific video stream with the highest possible quality of experience. One big advantage of this method is that it can be applied to any type of network and any type of video, whether the network has seen it before or not.

Engineers still have plenty of work to do before they can build complex intelligence into the Internet itself. Although the change won't happen overnight, it's already beginning. At the edges of the network, services such as Google and Facebook are now using sophisticated learning algorithms to infer our preferences, make recommendations, and customize advertisements. Wireless equipment manufacturers are building radios that can select frequencies and adjust their transmission power by "listening" to the airwaves. Still other engineers are finalizing protocols for creating mobile ad hoc networks so that police and rescue vehicles, for example, can communicate directly with one another.

Gradually, similar innovations will spread to other parts of the network. Perhaps as early as 2030, large portions of the Internet could be autonomic, while others will show the odd flash of actual insight. The future Net will exhibit a great diversity of intelligence, much like our planet's own biological ecosystems. ■

POST YOUR COMMENTS online at http://spectrum.ieee.org/cognitive0813