



By Neil Gershenfeld,  
Raffi Krikorian and Danny Cohen

In Barcelona about a century ago, Antoni Gaudí pioneered a fluid building style that seamlessly integrated visual and structural design. The expressive curves of his buildings were not just ornamental facades but also integral parts of the load-bearing structure. Unfortunately, a similar unification has yet to happen for the electronic infrastructure in a building. Switches, sockets and thermostats are grafted on as afterthoughts to the architecture, with functions fixed by buried wiring. Appliances and computers arrive as after-the-fact intrusions. Almost nothing talks to anything else, as evidenced by the number of devices in a typical house or office with differing opinions as to the time of day.

These inconveniences have surprisingly broad implications for construction economics, energy efficiency, architectural expression and, ultimately, quality of life. In the U.S., building buildings is a \$1-trillion industry. Of that, billions are spent annually on drawing wiring diagrams, then following, fixing and revising them. Over the years, countless “smart home” projects have sought to find new applications for intelligent building infrastructure—neglecting the enormous existing demand for facilities that can be programmed by their occupants rather than requiring contractors to fix their functionality in advance.

Any effort to meet that demand, though, will be doomed if a lightbulb requires a skilled network engineer to install it and the services of a corporate IT department to manage it. The challenge of improving connectivity requires neither gigabit speeds nor gigabyte storage but rather the opposite: dramatic reductions in the cost and complexity of network installation and configuration.

Over the years, a bewildering variety of standards have been developed to interconnect household devices, including X10,

LonWorks, CEBus, BACnet, ZigBee, Bluetooth, IrDA and Home-Plug. The situation is analogous to that in the 1960s when the Arpanet, the Internet’s predecessor, was developed. There were multiple types of computers and networks then, requiring special-purpose hardware to bridge these islands of incompatibility.

The solution to building a global network out of heterogeneous local networks, called internetworking, was found in two big ideas. The first was packet switching: data are chopped up into packets that can be routed independently as needed and then recombined. This technique marked a break from the traditional approach, used in telephone networks, of dedicating a static circuit to each connection. The second idea was the “end-to-end” principle: the behavior of the network should be determined by what is connected to it rather than by its internal construction, a concept embodied in the Internet Protocol (IP). Gradually the Internet expanded to handle applications ranging from remote computer access to e-commerce to interactive video. Each of these services introduced new types of data for packets to carry, but engineers did not need to change the network’s hardware or software to implement them.

These principles have carried the Internet through three decades of growth spanning seven orders of magnitude in both performance and size—from the Arpanet’s 64 sites to today’s 200 million registered hosts. They represent timeless insights into good system design, and, crucially, they contain no specific performance requirements. With great effort and discipline, technology-dependent parameters were kept out of the specifications so that hardware could evolve without requiring a revision of the Internet’s basic architecture.

These same ideas can now solve the problem of connecting

# The Internet of Things

The principles that gave rise to the Internet are now leading to a new kind of network of everyday devices, an “Internet-0”

EVEN SOMETHING AS SIMPLE as a lightbulb could be connected directly to the Internet, if suitably equipped with cheap circuitry that sends signals along the electrical wiring.



heterogeneous devices rather than heterogeneous networks. Extending the Internet all the way down to an individual lightbulb requires recognizing the similarities, and differences, between a bulb and the mainframe computers for which the Internet was originally developed.

## Smart Spaces

WE ENCOUNTERED the opportunity, and demand, for incorporating the Internet into a physical infrastructure through a series of installations we did with colleagues around the world. One, at a demonstration of future technologies for the White House/Smithsonian millennium events, was a smart bathroom shelf that detected pill bottles. It could remind people when to take their medication, let the pharmacist know when a refill was needed and help a doctor supervise care. Such a system could assist with health care compliance, which represents one of the greatest social and economic costs associated with aging.

Another installation, at New York City's Museum of Modern Art in 1999, used the furniture in a gallery to guide visitors through information about the exhibits. The idea was to avoid intruding on the visual and social space of the museum with conventional computer interfaces. At the opening, a museum benefactor proclaimed, "This is great, because I hate computers, and there are no computers here!"—not realizing that the furniture contained 17 Internet-connected embedded computers communicating with hundreds of sensor microcomputers.

Then came a building in the "Media House" demonstration in 2001 in, fittingly enough, Barcelona. The structural supports of this installation carried not just weight but also electricity and data. Lights and switches contained computers that allowed them to interact with one another and with other computers over the network. The associations between lights and switches were established on the fly.

## Overview/*Internet-Zero*

- Giving everyday objects the ability to connect to a data network would have a range of benefits: making it easier for homeowners to configure their lights and switches, reducing the cost and complexity of building construction, assisting with home health care. Many alternative standards currently compete to do just that—a situation reminiscent of the early days of the Internet, when computers and networks came in multiple incompatible types.
- To eliminate this technological Tower of Babel, the data protocol that is at the heart of the Internet can be adopted to represent information in whatever form it takes: pulsed electrically, flashed optically, clicked acoustically, broadcast electromagnetically or printed mechanically.
- Using this "Internet-0" encoding, the original idea of linking computer networks into a seamless whole—the "Inter" in "Internet"—can be extended to networks of all types of devices, a concept known as interdevice internetworking.

At an opening event for the Media House, one of the leaders of the high-speed Internet-2 project was on hand. He kept asking how fast data could be sent through the building. Someone reminded him that lightbulbs do not need to watch movies at broadband speeds and joked that the network of everyday devices was part of an "Internet-Zero," not Internet-2. The name stuck.

The IP processors that were developed to support these demonstrations were themselves not a research project, but the recurring interest in them led to the launch of the Internet-0 (I0) project. Those devices embodied seven principles that together extend the original notion of internetworking to interdevice internetworking.

## The Sevenfold Way

FIRST, EACH I0 DEVICE uses IP. In contrast, the many competing approaches for connecting devices introduce alternative standards. If a computer wants to communicate with one of these devices, the message must first be translated from IP into another protocol—a task that requires a special interface. Designers took this approach out of a belief that IP would be too demanding to implement in simple devices. But that need not be so. The code to run IP can be squeezed into a few kilobytes and run on a one-dollar microcontroller. The IP information adds about 100 bits to each message, which typically has a negligible impact on the response time and power requirements. In return for this modest overhead, the network avoids the cost of configuring and maintaining complex interfaces.

Second, the software is simplified by implementing the communications protocols jointly rather than separately. In a conventional computer, the tasks associated with networking are rigidly segregated. Low-level code handles the actual signals, such as the generation of electrical pulses sent over an Ethernet cable or through a telephone modem. That code sends its output to another layer of software that encodes and decodes the data. Above that are layers that supervise the sending and receiving of packets, assemble and disassemble the packets, and interpret standards for the content of the packets. Finally, the data reach the application, such as a Web browser.








Each of these layers is implemented separately as a kind of software version of a human bureaucracy; much of the code is dedicated to interlayer message passing. The layers are useful abstractions for developing the standards, so that one of them can be changed without needing to modify the rest, but that generality does not need to be preserved when they are actually executed. In an I0 device, the software implementation takes advantage of knowledge of the application.

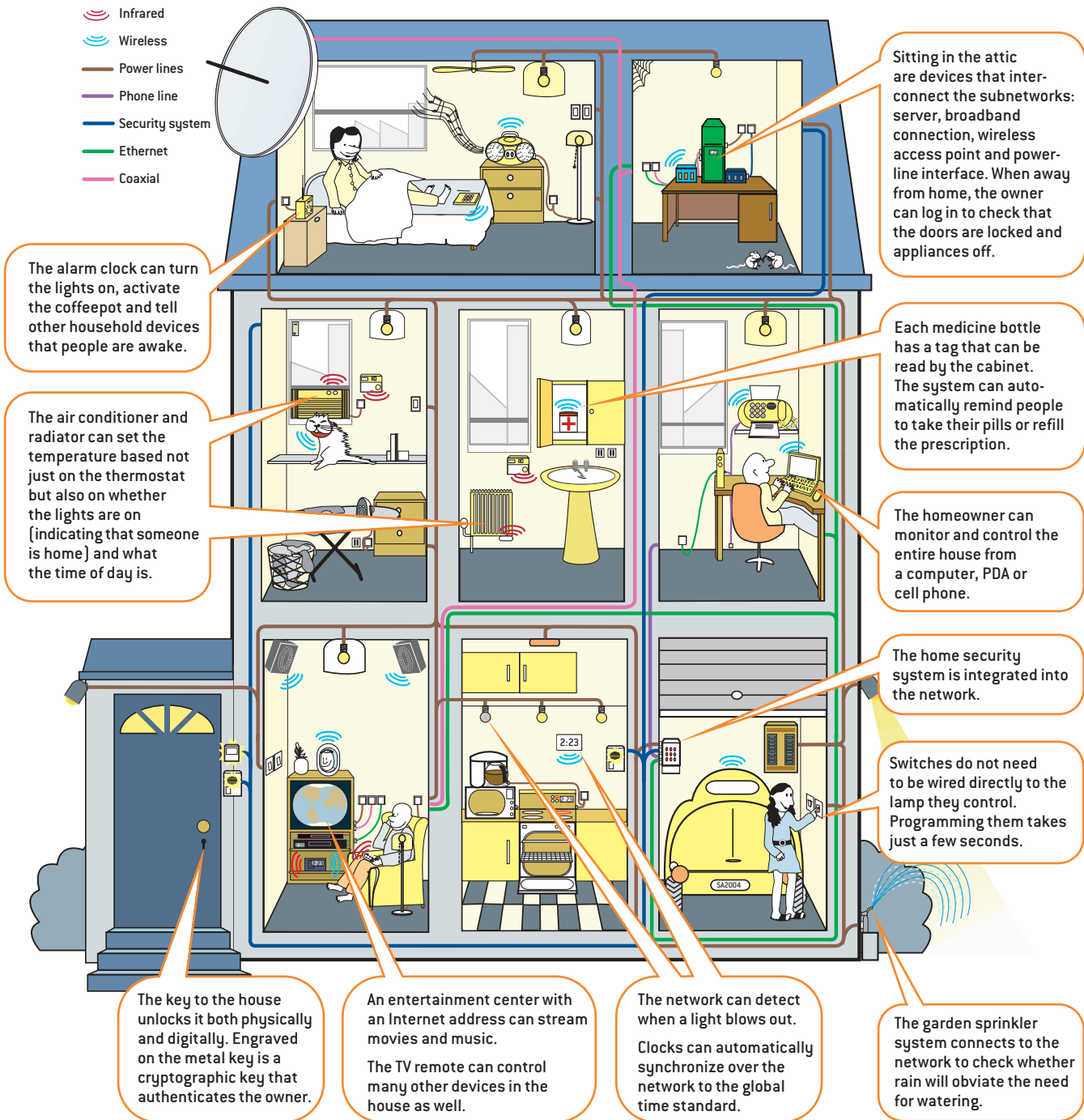
Third, two I0 devices do not require the existence of a third one in order to operate. Most computers on the Internet are either clients (such as Web browsers) or servers; the clients are useless without the servers. But each of the I0 lights and switches stores the data and routines it needs rather than relying on a central server, which would reduce reliability and increase costs. Although servers could enhance the system's value—say, by turning all the lights on or off at a particular time of the day—they are not necessary for it to function.

# One Network to Connect Them All

Internet-0 allows myriad devices to intercommunicate and interoperate: pill bottles can order refills from the pharmacy; light switches and thermostats can talk to lightbulbs and heaters; people can check on their homes from their offices. Existing technologies already allow many of these functions, but Internet-0

provides a single consistent standard. It can handle information sent through the AC power line, over a wireless connection or even engraved on a metal key, and it seamlessly integrates with the local and global computer networks. Devices can be configured by interacting with them rather than by typing on computers.

-  Infrared
-  Wireless
-  Power lines
-  Phone line
-  Security system
-  Ethernet
-  Coaxial



Internet-0 packets can be engraved onto a key (left) or printed as a bar code (below). In both cases, the vertical bars represent pulses and, once converted into electrical signals, can be put onto the network without further translation. The key engraving is a cryptographic key that allows the holder to reconfigure devices. The bar code contains "10" preceded by the address of the destination device.





## Whereabouts

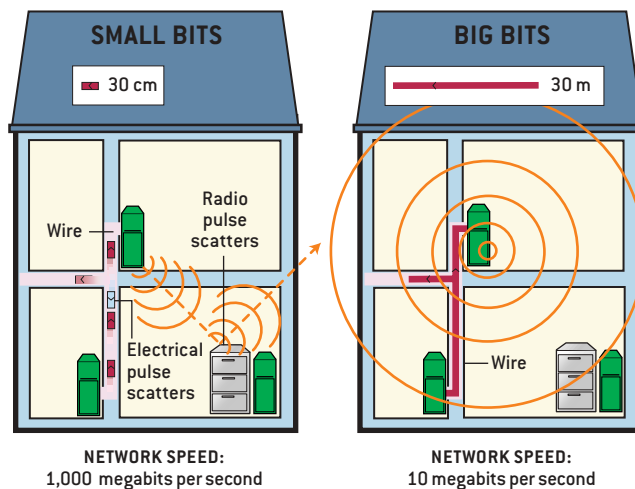
FOURTH, EACH IO DEVICE is responsible for keeping track of its own identity. A networked computer has five different names: a hardware Media Access Control (MAC) for the physical address on the local network (such as “00:08:74:AC:05:0C”), an IP address on the global network (“18.7.22.83”), a network name (“www.mit.edu”), a functional name (“the third server from the left”) and the name of a cryptographic key to communicate with it securely. Naming is one of the primary functions of servers. IO devices must be able to manage those functions for themselves when a server is not present, as well as accept the answers from a server when one is.

The most common kind of hardware address is centrally managed by assigning manufacturers blocks of addresses to burn into their products. But that kind of coordination would not be feasible for every light and switch produced on the planet. Instead a device can simply choose a random string for its address. The probability of two devices picking the same 128-bit number, for example, is just one in  $10^{38}$ . Users can assign functional and hardware names by interacting with a device—for instance, by pressing programming buttons on a light and switch that cause them to broadcast their addresses in succession and hence establish a control relation or by carrying the network address along with a cryptographic key between the devices to make the connection secure.

Fifth, IO uses bits that are bigger than the network. Bits have a physical size—they are nothing more than electrical, radio or light pulses. The time it takes to send such a pulse, multiplied by the speed at which it travels (which is typically about the speed of light), is the size. It used to be that bits were bigger than the network over which they were sent. Now they are much smaller—30 centimeters long for a data rate of a gigabit per second. If the network is bigger than that, problems arise at the interfaces between pieces of the network. Slight mismatches in the transmission properties at the interfaces generate spurious signals. Moreover, two computers that start transmitting simultaneously may not discover the collision until after they have sent many

## When Slower Is Better

When data are transmitted slowly, the size of the corresponding electrical or radio pulses is large, greatly simplifying the functioning of a computer network. Whereas small pulses reverberate off interfaces such as metal cabinets (in the case of radio signals) or wire junctions (for electrical signals), large pulses literally fill every inch of the air or wiring within a house. Therefore, they do not require hubs and other special equipment.



bits. That is why high-speed networks require special cables, active hubs, agile transceivers and skilled installers. But at a megabit per second, which is roughly the speed of a typical home cable modem or DSL connection (certainly enough for a lightbulb), a bit is 300 meters long—so large that it can span an entire building network. It no longer matters what interfaces the network has.

## On the Telegraph Road

SIXTH, THE USE of big bits allows the data that make up a packet to be represented in the same way no matter what physical medium conveys them. When bits are small, their physical representation, or modulation, must be customized to each communications channel. The modulation used by a dial-up modem is very different from that used by a cable modem, because a twisted-pair phone line and a coaxial cable differ in the amplitude, frequency and phase of the signals they can carry. But when the bits are big, the detailed propagation characteristics of each channel make no difference.

Morse Code exploits this principle. It can be tapped on a telegraph, flashed from ship to ship or banged out on a pipe. These very different channels carry the same data at the same rate using the same encoding scheme. No translation is needed. The information is conveyed simply by the arrival time of a change in the physical medium—a voltage jump on the telegraph line or the sudden onset of sound. The precise tone or amplitude does not matter.

IO is similar except that it uses “0” and “1” impulses instead of dots and dashes. Like today’s modems, an IO device sends a packet in a string of eight-bit bytes, each framed by start and stop bits. A 0 bit is represented by an impulse followed by a

NEIL GERSHENFELD, RAFFI KRİKORIAN and DANNY COHEN are researchers that seem to thrive by defying traditional disciplinary boundaries. Gershenfeld directs the Center for Bits and Atoms at the Massachusetts Institute of Technology, which gets support from the National Science Foundation. He studies the relation between physical form and logical function in everything from quantum computers to automobile safety systems, from a computerized cello he developed for Yo-Yo Ma to tools used by rural Indian villagers. Krikorian is an M.I.T. graduate student who has led the development of hardware and software for Internet-0. His earlier work in academia and industry spanned very small embedded IP stacks to very large distributed computation engines. The Internet-0 project grew out of their collaboration with Cohen, a Distinguished Engineer at Sun and one of the fathers of the Internet. He pioneered real-time interactive applications over the Arpanet, which helped to lead to the development of the Internet Protocol (IP), and started the MOSIS IC fabrication service.

pause; a 1 bit is a pause followed by an impulse; a start or stop bit is a pair of impulses. Such a scheme, called Manchester encoding, simplifies distinguishing between a valid 0 or 1 and an interfering or absent signal. In addition, the spacing between the pulses in the start bit allows the receiver to measure the transmission rate; the rate does not need to be fixed in advance. If extra noise immunity is needed, the sender and receiver can agree to use a procedure to vary the time between bytes (as is done in ultrawideband radios), which would help them separate the signal from noise while retaining backward compatibility with simpler I/O devices that use just the framing pulses.

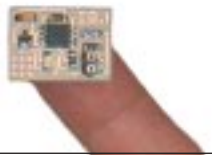
As long as the bits are sent slowly enough to be bigger than the network, the encoding can be the same across all types of physical media. The pulses could be sent through a wire, coupled onto a power line, clicked by a speaker, printed onto a page or

doctor rather than a prearranged code that the reader would have to translate into English. The information would be carried by the pill bottle rather than programmed into the reader.

If these features are such a good thing, then why were they not put into practice sooner? The problem is that communications engineers have had a longstanding bias that bandwidth is scarce and hence must be used efficiently, dating back to the days when it was. The developers of the original Ethernet were faulted at the time because it did not reach the fundamental communications limits set by quantum mechanics. That was true but irrelevant. Ethernet succeeded because of its relative simplicity.

Today networks are indeed approaching quantum limits, sacrificing simplicity for ever more spectacular performance gains. I/O reverses this trend. It is a technological case of less being more; speed is sacrificed for interoperability.

## Data could be sent through a wire, **CLICKED** by a speaker, **PRINTED** onto a page or engraved on a key—all using the same Internet-0 encoding.



INTERNET-0 PROCESSOR

engraved on a key. Each of these media would pass a different part of the pulse: a power line filters out high frequencies and a radio antenna the low ones. All that is required is for some of the frequencies in the impulse to get through. (The detailed frequency response would be useful, however, if the I/O device needed to probe its physical environment.)

This representation extends the end-to-end principle of the Internet to modulation. When a computer transmits packets using IP, it does not need to know anything about the networks that will carry the packet. Similarly, when a device uses I/O impulses, it does not need to know about the media that will carry the signal.

### Less Is More

THE SEVENTH AND FINAL attribute of I/O is the use of open standards. The desirability of open standards should not need saying, but it does. Many of the competing standards for connecting devices are proprietary. The recurring lesson of the computer industry has been that proprietary businesses should be built on top of, rather than in conflict with, open standards.

As an example of I/O in action, return now to the bathroom shelf for managing medication. Our demonstration project used radio-frequency identification (RFID) tags in the pill bottles—tiny disposable chips that are powered by the radio signals that interrogate them. We had to configure the tag reader to know what to do with the data it received. The same is true for the RFID systems now being deployed across consumer and military supply chains: an army of consultants and contractors is needed to configure all the RFID readers.

The process would be much easier using I/O. The tag would encode an IP packet—one might call it an IPID tag—and the reader would merely have to relay the packet to the network. The packet might contain the addresses of the pharmacist and

I/O is aimed at the scaling limits imposed by network complexity rather than raw performance. It is not intended to replace the existing Internet; it provides a compatible layer below it. An I/O device depends on the current routers, gateways and name servers to carry packets between I/O subnetworks. Over time, however, the distinction between I/O and the rest of the Net may recede. The protocols that run in the Internet's servers, such as the ones used to direct IP packets to their destinations, are described as algorithms—a set of instructions for finding the best path for a packet to take. But the protocols can also be understood as optimizations—a way to make best use of the available communications resources given their constraints. Recent research has shown how to solve such constrained optimizations using distributed systems rather than central processors. Thus, I/O nodes might one day be configured to solve global network management problems through their local behavior so that the higher-level Internet architecture can emerge from their interactions.

If so, the ultimate destiny of Internet-0 is not just turning on the lights. An I/O network will be indistinguishable from the computers that it connects; it really will be the computer. By allowing devices for communications, computation, storage, sensing and display to exchange information in exactly the same representation—around the corner or around the world—the components of a system can be dynamically assembled based on the needs of a problem rather than fixed by the boundaries of a box. **SA**

### MORE TO EXPLORE

**How the Internet Came to Be.** Vinton Cerf in *The Online User's Encyclopedia*. Edited by Bernard Aboba. Addison-Wesley, 1993. Available at [www.internetvalley.com/archives/mirrors/cerf-how-inet.txt](http://www.internetvalley.com/archives/mirrors/cerf-how-inet.txt)

**When Things Start to Think.** Neil Gershenfeld. Henry Holt, 1999.

Other publications are available at [cba.mit.edu/projects/I0](http://cba.mit.edu/projects/I0)