Internet 0: Interdevice Internetworking

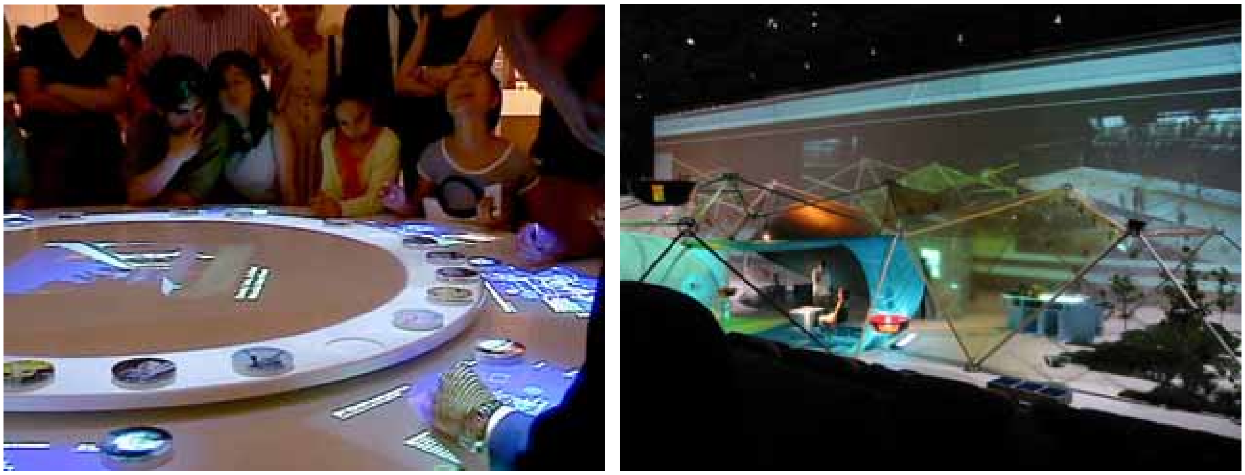
Neil Gershenfeld, Raffi Krikorian, Danny Cohen

to appear in *Scientific American*
draft: *6/25/04*

The Internet is appearing everywhere. Phones speak it, appliances process it, coffee shops and even coffee pots serve it. But we'll be doomed if your coffee pot demands the services of an IT department. As remarkable as its growth has been, Internet implementations that were appropriate for mainframes are not for connecting everything everywhere. Yet there are compelling reasons for a light bulb to have Internet access, ranging from construction economics to energy efficiency to architectural expression. Accomplishing this with the cost and complexity expected for installing and maintaining a light bulb rather than a mainframe raises surprisingly fundamental questions about the nature of scalable system design. The success of the Internet rests on the invention of "internetworking" across unlike networks; the Internet zero (I0) project is extending this insight to enable "interdevice internetworking" of unlike devices.

Internet hosts: an original PDP-10, and an Internet 0 node

We did not set out to invent a new Internet, or in fact any kind of network at all. Rather, work on new devices to interface computers with their environments required developing ever-simpler ways to connect to those environments. For example, a presentation on the future of technology for the White House/Smithsonian Millennium events showed a communicating bathroom shelf that could help seniors manage their medication, which represents one of the greatest social as well as economic costs of aging. Another, at New York's Museum of Modern Art, used the furniture in a gallery as an artistic information interface; at the opening, a beaming museum benefactor proclaimed that "This is great, because I hate computers, and there are no computers here!" (not realizing that the table she was pounding on contained seventeen Internet-connected computers, communicating with hundreds of sensor microcomputers). And on an even larger scale, a demonstration "Media House" in Barcelona sought to make the information in a structure as expressive as its physical form.



MoMA and Barcelona Internet 0 testbeds

In Barcelona around the turn of the last century Antonio Gaudi pioneered a fluid building style that seamlessly integrated visual and structural design; the Media House sought to do the same for the intelligence in a building. Computers were embedded in lights and switches, giving them each an Internet address so that their relationships could be dynamically programmed rather than fixed by a wiring diagram. Each device contained the data and procedures for its control functions, allowing them to operate as a distributed system without relying on central servers. And physical programming interfaces were provided so that, for example, installing a light and then operating a switch could associate them over the network without requiring commands from another computer to configure them. All of this was possible using just a few dollars of parts in each device, by revisiting some standard assumptions in implementations of the Internet specifications. One assumption is that they should operate in the same layered way that they are described, however a surprisingly large fraction of the resulting hardware and software is devoted to a kind of technological expression of human bureaucracy. And another assumption is that faster is better; higher-speed networks have hidden costs that have made them increasingly difficult to use in physical infrastructure.

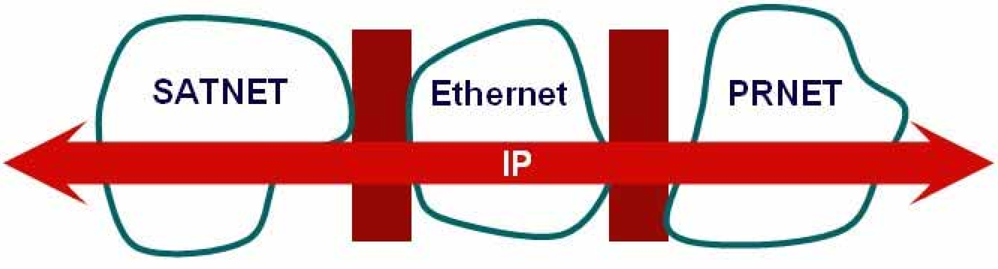
At an opening event one of the architects of the high-speed Internet 2 project kept coming back to ask how fast data could be sent through the building infrastructure. After being reminded that light bulbs don't need to be able to watch broadband movies, he was jokingly told that the emerging network of everyday devices was part of an Internet zero, not Internet 2. The name stuck. I0 is not a replacement for the current Internet (call that Internet 1); it is a set of principles for extending the Internet down to individual devices.

If a networking researcher had trouble understanding the value of slow networks, the building construction industry did not. Ever faster computer networks have paradoxically become less and less relevant to their needs. Unlike plumbing or power, both the network infrastructure and the workers required to plan, install, and operate it have become increasingly specialized. Countless "smart home" projects have sought (with limited success) to find killer applications for bringing this kind of connectivity into homes, neglecting the biggest one of them all: building the buildings themselves. In the 1997 US Economic Census, the annual revenue from making computer and electronic products was $438B, compared to $858B for construction. Even if the hardware was free, the cost of installing intelligent infrastructure in all those buildings could add up to the size of today's entire IT industry. This presents a technological challenge that demands neither gigabit speeds nor gigabyte storage, but rather simpler configurations

Given the potential size of this market, there is a long list of alternative competing standards attempting to fill parts of it: X10, LonWorks, CEBus, BACnet, ZigBee, Bluetooth, USB, IRDA, HomePlug, SPI, I^2C, ... While each has strengths, each is also encountering many of the same scaling issues that the Internet faced, leading to repeated reinvention of schemes for naming things, managing reliability, and routing across subsystems. This situation is in fact analogous to the early days of the Internet, when multiple competing networking standards threatened to devolve into islands of incompatibility.

The Internet is a packet-switched network, which means that, unlike a traditional phone switch that connects wires, it chops data up into packets that can be sent independently and reassembled at their destination. This shares resources better because it's driven by demand, and it's more reliable because packets can be routed around disturbances. Packet switching was developed by Len Kleinrock when he was a grad student at MIT in the 1960's, and was used by his officemate Larry Roberts to develop a network for the Advanced Research Projects Agency (ARPA, now DARPA) to allow researchers to share expensive computing resources, the Arpanet. Other packet-switched networks followed, including Packet-Radio Net (PRNET), Ethernet, and SATNET. A requirement for a custom interface between every type of network was a sure recipe for disaster; ways were sought for their inter-operation, later known as "Internetting".

Enter Vint Cerf and Bob Kahn, who proposed a protocol for packet network intercommunication now known as TCP for "Transmission Control Protocol". This hid the details of the networks carrying a packet, and also made sure that the packets reached their destination and were assembled in the right order. That extra reliability was important for sending files, but added an unacceptable overhead for realtime applications. Quoting Vint Cerf: "So Danny Cohen at ISI, who was doing a lot of work on packet voice, argued that we should find a way to deliver packets without requiring reliability. He argued it wasn't useful to retransmit a voice packet end-to-end. It was worse to suffer a delay of retransmission. That line of reasoning led to separation of TCP, which guaranteed reliable delivery, from IP. So the User Datagram Protocol (UDP) was created as the user-accessible way of using IP." After three versions of TCP, thus was born IP (Internet Protocol) version 4, today's ubiquitous standard.



Internetworking

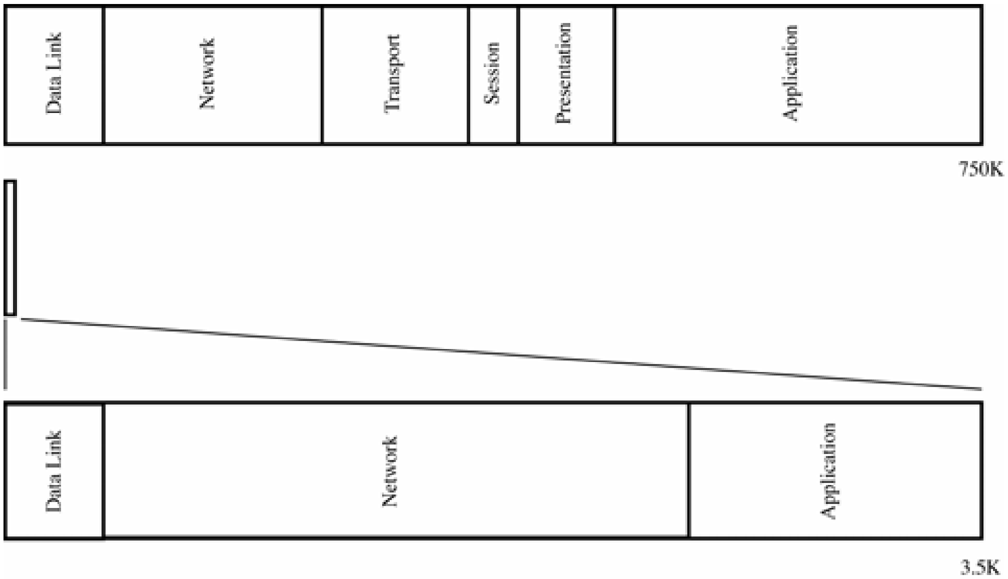
The ARPAnet was ambitiously designed to handle up to 64 sites with up to 4 computers per site, far exceeding any perceived future requirement. Today there are more than 200 million registered hosts on the Internet, with still more computers connected to those. IP, TCP, and UDP have so far survived more than 3 decades and about 7 orders of magnitude of performance improvement. One reason is that, with great effort, performance numbers were kept out of the specifications. There's no mention of any implementation details. And a second is that new applications have been developed by changing what's connected to the network, not how the network itself works; this became enshrined in Saltzer, Reed, and Clark's "end-to-end" principle.

The Internet was, however, originally developed to connect mainframes. There are some however differences between a light bulb and a mainframe, which are reflected not in the original specifications but in how they have been implemented since. Consider now the Barcelona Media

House installation, which *was* designed for light bulbs. A meeting at MIT afterwards identified 7 guiding principles that it demonstrated; no one of these is new, but their intersection is unlike both the Internet as it exists today and all of the competing alternatives for networking things. Taken together, they provide the architecture for Internet 0:

IP to leaf nodes. Each device in the Media House used the Internet protocol, rather than switching to a different standard for the last hop. Historical concerns about bringing IP to the leaf nodes of a network have been rooted in a fear that the IP protocols impose unacceptable resource requirements on both the device and the network, hence incompatibilities have been built in at the edges of the network. But the IP stack used in the Media House fit in just a few kilobytes of code running on an inexpensive microcontroller, corresponding to a fraction of a square millimeter of silicon and pennies of added cost in a custom chip. Using IP added about 200 bits to each data packet; most any kind of custom addressing scheme would need something comparable. And because Internet routers have grown from having thousands to billions of bytes of memory in their routing tables they can accommodate this extra layer of hierarchy in the Internet.

Compiled standards. It was possible to implement the Internet in a few kilobytes by recognizing that a light bulb doesn't need to do everything that a mainframe does. The Arpanet's layers were frozen in the International Standard Organization's (ISO) Open System Interconnect (OSI) network model, which defines 7 of these, from the physical communication medium through to the application. But for a given task the whole does less work than is apparent from the sum of the parts. They can be simplified by implementing them jointly rather than separately, just as a computer compiles a program written in a general-purpose language into an executable that does a particular thing. This not only removes the overhead of passing messages between layers, it's also possible to take advantage of knowledge of the application; a switch whose only job is to create control packets does not need to know how to route them. The steady march of VLSI scaling will not eventually obviate the need for this kind of optimization, because even as transistors get smaller there are still fundamental costs associated with algorithm complexity, including power consumption and device packaging.



Layers and code size for a Linux and I0 Web server

Peers don't need servers. In a world of clients and servers, small devices present and gather information for a larger machine. But centralized networks have a single point of failure; without the central server, the clients are useless. Even relatively simple devices can now hold, manage, and communicate their own state. In the Media House, each switch was responsible for keeping track of the things that it controlled, and each light for the switches that it listened to. Servers could add value to the network, aggregating data and implementing more complex control functions, but they weren't necessary for it to operate. No one device needed any other device in order to do its job.

Physical identity. A switch in the Media House had three kinds of names: an Internet address ("192.168.1.100"), a hardware address ("00:12:65:51:24:45"), and a functional address ("the switch by the door"). The first depends on which network the switch is connected to. If that network is not connected to the rest of the Internet then a random address can be chosen (so that a name server is not required), but if there is an Internet connection with a name server available then that can be used. These addresses are associated with networks rather than tied to devices because routers need to be able to use them to direct packets properly. The second name is fixed for a particular device. In Ethernet chips these are called MAC (Media Access Control) addresses; blocks of them are assigned to manufacturers to burn into their chips. Since that system would be unwieldy to centrally manage for anyone who wanted to develop or produce any I0 device, random strings are generated as MAC addresses; the probability of two 128-bit random strings being the same is just 1 part in 10^{38}. The Internet and network addresses can be associated through use of the device without requiring a server, such as the example of installing a light and then operating a switch. When that happens the light and switch communicate their addresses, and then can agree on establishing a logical connection. Or a handheld remote, which is just a portable I0 node, can be used to carry the logical identity between separated physical devices to associate them. An important application of that capability is carrying a cryptographic key to establish one more kind of name, a secret string of bits that is shared between devices based on having physical access to them. This can then be used in a Message Authentication Code protocol to encrypt, for example, the time to day, so that a switch can prove to a light that it knows the right private key to work it, but an eavesdropper can't later replay the message to control the light. As we'll see, even a conventional key can contain a mechanically-encoded I0 packet with a cryptographic key for a secure electronic lock.

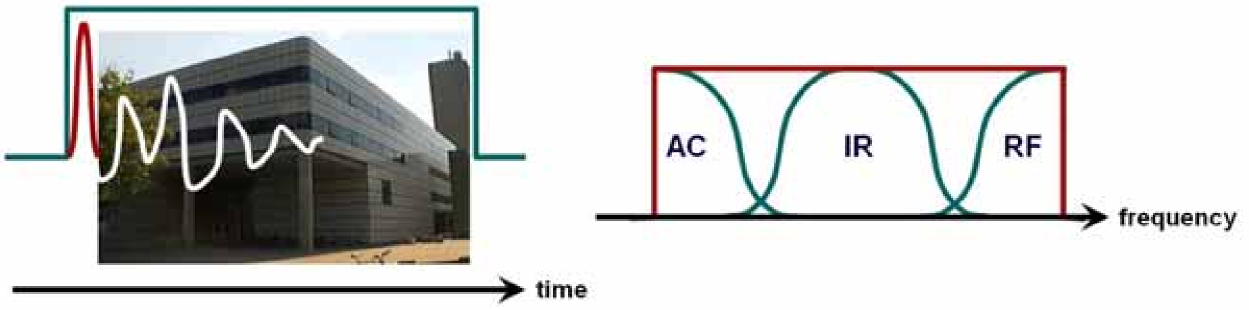
Open standards. While this should not need saying, it does. Compare the explosive growth of the Internet with the relative chaos in the US of the cellular phone network, where there are multiple redundant proprietary systems battling for the same subscribers. The same thing threatens to happen with the competing standards for connecting things; the recurring lesson in the IT industry has been that proprietary businesses should be built on top of rather than in conflict with open standards.

Big bits. A bit in a network represents a unit of information (a 1 or a 0), and is represented by some kind of traveling excitation (electrons in a wire, photons in a fiber, electromagnetic waves in the air). The disturbance has a speed, which depends on the medium but for electrical signals is typically on the order of the speed of light (~$3x10^{8}$ meters per second). That may sound fast, but if the bits are being sent at the current Ethernet speed of a gigabit per second this corresponds to a size per bit of about a foot. If a network is bigger than that then spurious bits will be created by scattering from any interfaces in the network, and two nodes could begin transmitting simultaneously and not realize it until after their bits collide. This is why high-speed networks require special cables, active hubs, and agile transceivers. If, on the other hand, a bit is bigger than the network, then it will fill the network independent of how it is configured. For a 100m building this corresponds to about a million bits per second, equivalent to a DSL connection, which is plenty for a light bulb. If bits are sent at this rate then they have time to settle over the network, which greatly simplifies how they can be encoded.

End-to-end modulation. For two devices to communicate they must agree on how to represent (modulate) information. This choice depends on the range of available frequencies, as well as the amount of noise and time delay at each of those frequencies. The frequency response can be measured by sending in a short spike and then recording the response to the impulse, analogous to hitting it with a hammer and then listening to it ring. The goal of high-speed network design is to keep that ringing as short as possible.

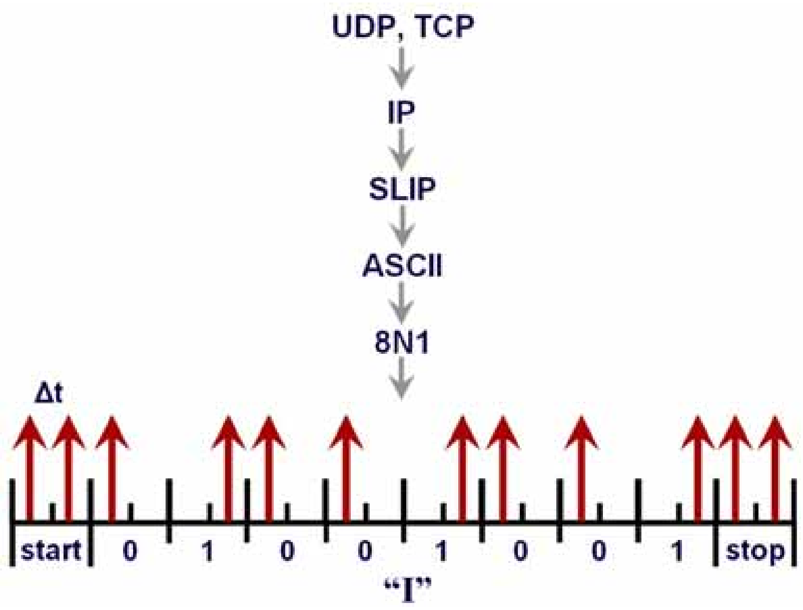
If, however, there's no need to communicate particularly fast, then the impulse response itself can be used to encode a bit by the time of its arrival. Even if the response to the impulse is not carefully controlled, its onset can be timed very precisely. This is done in ultra-wideband (UWB) radios that use advanced electronics to produce sub-nanosecond impulses corresponding to GHz signals, but even relatively modest circuits can generate transients covering the frequency response of most common communication devices. As long as successive impulses are separated by the time they take to settle over the size of the network, arbitrary inhomogeneities over that size can safely be ignored. Viewed on that time scale, a tangle of electrical wires is just as useful as carefully-laid optical fiber. If there are regulatory or physical limits on the use of a particular device, the impulses can be shaped to use specified bands of frequencies.

Transmitting information in impulses is certainly not new; it dates back to the origins of electronic communications in Guglielmo Marconi's spark gap and in Samuel Morse's code (awarded US patent number 1,647 in 1840). Originally developed for the telegraph, Morse Code could equally well be flashed from ship to shore, tapped on pipes, or puffed out in smoke. This diversity was possible because the only requirement was that the medium be able to represent some kind of distinguishable event, and because the data rate was not fixed in advance but interpreted by the receiver. The speed of light hasn't changed since Samuel Morse's day, but the resolution with which the arrival of impulse responses can be timed has.



Impulse response and frequency response. The impulse creates energy at a broad range of frequencies; an AC powerline passes the low-frequency part, an infrared diode and detector can pass mid-frequency driving signals, and a pair of radiofrequency antennas will couple still higher frequencies.

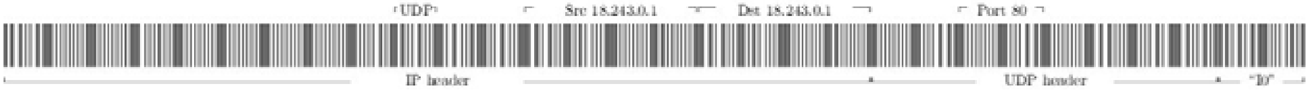
Devices that can communicate at unpredictable times (asynchronously), such as a telephone modem, commonly send each byte of data in a format known as "8N1", which means that there is a start bit, eight data bits, then a stop bit, and no extra bits are used for detecting errors (since that can be done at a higher level). This can be mapped into I0 impulses by dividing each time slot into two half-intervals, with a zero being represented by an impulse in the first half, a one in the second, and a start or stop bit using both intervals. Dividing a bit in half is called a Manchester encoding; it cuts the effective data rate in half in return for reliability because it's possible to distinguish between receiving a zero and not receiving anything at all. The original Ethernet standard used a Manchester encoding so that there would be time between bits to listen for competing use of the network.



Internet 0 impulses

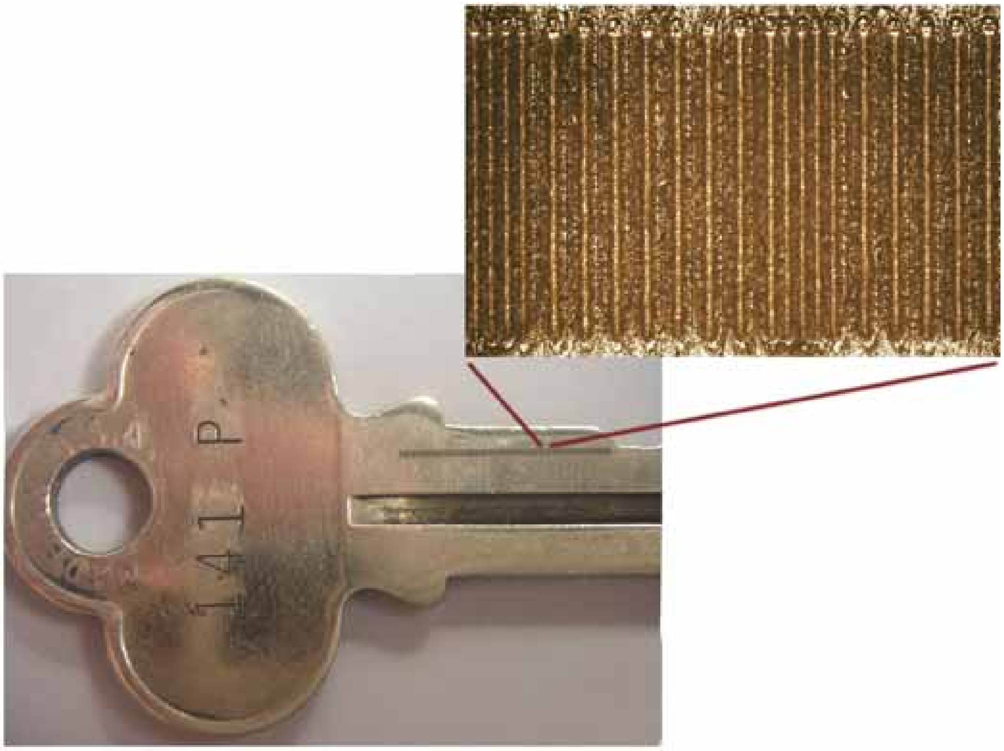
When used with I0 impulses, this kind of encoding has two significant consequences. First, the time between the start impulses can be used to set the timing for the rest of that byte. Instead of needing to know in advance that the serial data rate is, say, 9600 bits per second, the receiver can adapt to whatever rate the sender chooses. Remember that one of the core reasons the IP protocols have scaled so well is that they didn't contain technology-dependent numbers; self-clocking I0 impulses could be sent across a chip or across a solar system depending on their spacing. And second, unlike a conventional serial connection, that timing can be used to allow multiple terminals to share a channel simultaneously. Two succeeding impulses from different sources that are erroneously interpreted as a start bit can quickly be rejected because they won't be followed by the impulses for the rest of a byte.

The binary character codes for the letter I and the number 0 are 01001001 and 00110000, hence "I0" written as I0 impulses is |||--||-|--||-|--|||||-|--|-||-|-|-|-|| . Putting this into a complete UDP/IP packet becomes:



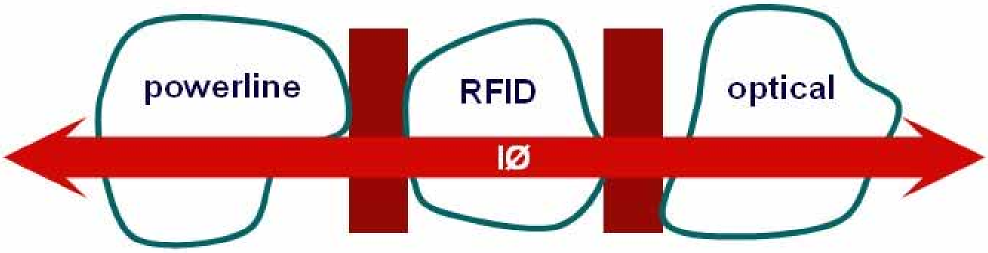
"I0" in an Internet 0 packet

This is 240 bits; although it could be compressed, at a building-scale megabit per second it takes just 0.24 msec to send (human's can't perceive response times faster than about 10 msec). It isn't just a picture of an Internet packet, it *is* one. If you fed it into a scanner, the analog signal from the optical sensor could go directly onto an I0 network as a valid packet. The representation will be exactly the same if the packet is capacitively coupled into a powerline, inductively read from an RFID tag, flashed by an LED, printed on a page, or engraved on a key. Conversion among them is done simply by changing the physical media access device. I0 isn't an application for Ethernet, Bluetooth, IRDA, ...; it's a single alternative to all of them. This is the principle of interdevice internetworking, extending the Internet's end-to-end principle all the way down to individual devices.



A key engraved with an I0 packet containing the lock's cryptographic key (photo credit: Manu Prakash)

If this is such a good thing, then why wasn't it done sooner? Communication engineers have had a longstanding bias that bandwidth is scarce and hence must be used efficiently, dating from the days when it was. When Bob Metcalfe developed Ethernet at Xerox PARC in 1973, he was criticized by a colleague because it was "not quantum noise limited," i.e., not operating at the fundamental physical limits. The colleague was right about the physics, but wrong about its implications. Ethernet was a tremendous success because of its relative simplicity and reliability. However, succeeding networking standards have indeed approached fundamental physical limits as engineers have sought to push more and more data through them. In so doing, they've sacrificed simplicity for (spectacular) performance gains. Internet 0 is a technological case of less is more: the physics of "big" bits allows speed to be traded off against interoperability across devices. The only reason to use a different protocol for each kind of device is if optimality is more important than compatibility. Rather than repeat the prehistory of the Internet and impose interface processors between dissimilar devices, as long as the data rate for network-sized bits is acceptable then a single representation can be used across all of them.



Interdevice internetworking. Rather than using different formats and protocols requiring translation, each of these links can carry the same Internet 0 signal and differ only in the physical access devices.

Because the purpose of Internet 0 is to bring IP connectivity to the leaf nodes, it does not attempt to recreate all of the functions of the Internet at the bottom of the network. To communicate globally, an I0 device still needs the Internet's gateways and routers. But after removing servers within I0 subnets it's unsatisfying to reimpose the need for them between subnets, presenting a challenging research question: can I0 nodes alone organize themselves hierarchically so that the global functionality of gateways and routers emerges from the local interactions in a system?

A suggestive hint that I0 nodes can indeed do this comes from the possibility of representing "mathematical programs" as graphical models. Mathematical programming is not a computer language, it is the language of optimization, expressing both goals and the constraints on them. These are written as formulas that people understand rather than as programs that computers understand, but they can alternatively be represented graphically by the relationships among the variables and their constraints. It's then possible to compile such a graph into mathematical messages that get passed on the graph in order to solve the optimization problem that it represents. These messages don't just carry a description of the problem to be solved, they actually *are* its solution. Ongoing research is applying these ideas to I0 nodes, blurring the boundary between I0 and the rest of the Internet.

The ultimate destiny of Internet 0 then is not just lighting light bulbs. An I0 network can not be distinguished from the computers that it connects; it really *is* the computer. Because it allows 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.

Further Reading

The UnPrivate House,
I.D. Magazine Interactive Media Design Review,
June 2000

Media House Project,
V. Guallart, ed.
IaaC
Barcelona, 2004

"How the Internet Came to Be",
Vinton Cerf,
"The Online User's Encyclopedia",
Bernard Aboba,
Addison-Wesley,
November 1993,
ISBN 0-201-62214-9

"Computer Network Development to Achieve Resource Sharing"
L. G. Roberts and Barry D. Wessler
Proceedings of the Spring Joint Computer Conference,
Atlantic City, New Jersey - May 1970

"Information Flow in Large Communication Nets",
L. Kleinrock,
RLE Quarterly Progress Report, Massachusetts Institute of Technology, July 1961.

"A Protocol for Packet Network Intercommunication"
Vint Cerf and Bob Kahn
IEEE Transactions of Communications, vol. com-22, no. 5
May 1974

"End-to-End Arguments in System Design"
J.H. Saltzer, D.P. Reed and D.D. Clark
ACM Transactions in Computer Systems vol. 2
November, 1984
277-288