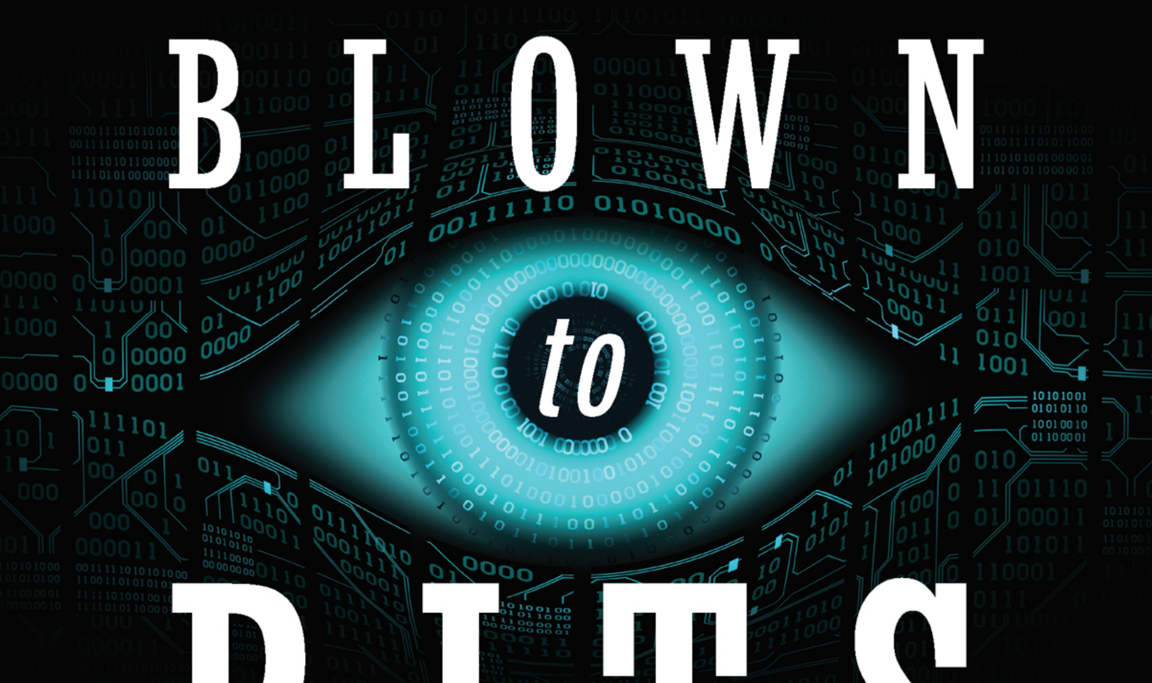
HAL ABELSON • KEN LEDEEN
HARRY LEWIS • WENDY SELTZER



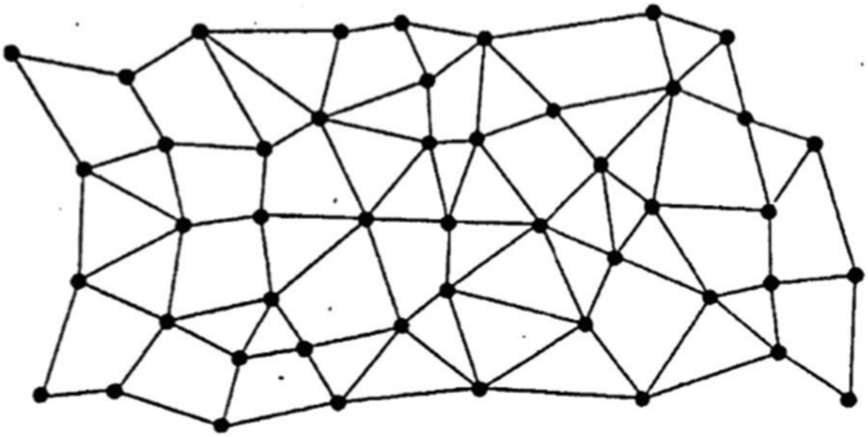
BLOWN *to* BITS

[YOUR Life, Liberty, and Happiness After the Digital Explosion]

SECOND EDITION

Blown To Bits

Second Edition



Baran contributed a second important idea: If a switchpoint went down, another route could be found that did not go through it, as long as the switchpoint itself was neither end of the communication. By setting the switches the right way, communication between two points could be established along a particular path. But knocking out any of the points along the path would then interrupt that communication. So would an ordinary hardware failure at any of those intermediate points. It was important to protect the integrity of individual communications even as the network components failed in unpredictable ways.

Baran proposed to chop communications into small chunks of bits, what we today refer to as “packets.” In addition to the “payload,” a fragment of the communication itself, the packets would contain information identifying the source and destination (much like the address information on a postal letter), and also a serial number so that the destination node could reassemble them in the right order if they happened to arrive out of sequence. With this much information on the “envelope,” the packets comprising a single communication did not need to follow the same path. If a portion of the network was unavailable, the network nodes could direct packets along a different path. Making this all work was not simple—how would the network nodes know in what direction to forward a packet?—but in principle, Baran’s idea of a mesh-like interconnection and packetized communication would meet the military requirements for survivability.

Protocols: How to Shake Hands with Strangers

Once ARPANET was operational and connected a few dozen computers, it started to become clear that what needed to be connected were not individual computers but existing computer networks. Different ways of networking computers together could coexist, as long as the networks used some common language for communicating with each other. And in the 1970s and 1980s, different kinds of computer networks did exist, each using the standards of a different computer company. IBM had its SNA (Systems Network Architecture). Digital Equipment Corporation had DECnet. Apollo Computer connected its machines in a ring rather than a branching tree or mesh. Each company touted the advantages of its networking scheme, and some of the claims were valid for particular use cases. But none of the manufacturers had any incentive to make their machines interoperable with those of other manufacturers—until ARPA declared that it would pay for no more computers unless they could be interconnected. Starting from the success of the ARPANET, Vinton Cerf and Robert Kahn designed the protocols for interconnecting computer networks.[8] That is, they designed the Internet.

The Internet *is* its protocols. The Internet is not a machine or even a collection of machines. It's not some piece of software. It is a set of rules. Any person or organization can build hardware or write software that abides by those rules and become a functioning part of the Internet.

Protocols are communication conventions, like the convention that people shake their right hands. Having everyone greet each other by shaking their left hands would work equally well, but the established convention of right-hand shaking makes it possible for strangers to greet each other with no prior mediation. Internet protocols are the conventions by which different networks shake hands in order to pass information from one network to the other. Each network can operate as it wishes internally; only at the points where networks are connected together do the Internet protocols become relevant.

The decision to make ARPANET a packet-switched network simplified the Internet design considerably. Networks were connected to the Internet via connection points called *gateways*. If a gateway behaved as it should, information would flow through it. If it didn't behave properly, that caused no harm except to cut off that network from the rest of the Internet. No computer or network of computers needed permission to join the Internet. If it adhered to Internet standards, it could be understood by others and could interpret messages directed to it.

As we look at the Internet today, it seems varied and complicated: so many different kinds of content, so many different kinds of devices, and so many different kinds of connections. But it's all built on top of a single protocol, known simply as Internet Protocol (IP). It's the job of IP to get a single packet of around a thousand bits from one end of a communications network link to the other. The bits, as delivered, may contain errors; nothing physical ever works perfectly all the time. But errors can be recognized and, if necessary, dealt with. To get packets across the network, IP is used repeatedly, bucket brigade style, with each switching point receiving packets, checking them, and then dispatching them toward their intended destination.

The simplicity of the Internet design made it possible to build protocols on top of protocols to expand the Internet's utility. The earliest uses of the Internet were for logging in to time-shared computers remotely, for moving files from place to place, and for electronic mail. All these services required the data to arrive error free but not necessarily instantaneously. No one would notice if a file transfer or an email delivery took an extra fraction of a second, but having a single bit turn from a 0 to a 1 in transit could have catastrophic consequences. For such transfers, a protocol was developed to make sure that packets sent by the source were received correctly and reassembled in the correct order. Given the unreliability of the intermediate nodes of the network, this requires some bookkeeping at both source and destination. A packet, once received, is acknowledged by sending a special packet back from the destination to the source. The source runs a timer; if a packet sent is not acknowledged before the timer runs down to zero, the source figures that the packet has gotten lost somehow and retransmits it.

The details are tricky, but they are not important to the big picture. The result is that as long as the switches are making their best effort to pass packets along toward the destination, any message sent will be received in perfect order. The protocol that ensures such perfect transmissions is called Transmission Control Protocol (TCP). Because the underlying protocol for moving packets along single links of the network is IP, TCP/IP is the everyday name for the pair of conventions that make reliable communications possible across an unreliable network.

Since there are no rules for joining the Internet, it is fair to wonder about the "best effort" assumption. Couldn't a rogue actor try to sabotage the network by adding switches that would discard or misdirect packets rather than send them toward their intended destination? Indeed, that could happen, but neighboring switch points would eventually realize that the packets were not being delivered and would start avoiding the rogues. Internet routing heals itself by learning to avoid trouble spots—not just in case of hardware failures but also in case of malice. The Internet becomes more reliable the larger it becomes and the more interconnected it becomes.

The Internet worked because once a large enough number of parties agreed to use it in the intended way, bad actors could in effect be frozen out since they were few in number.

In addition to routing information and payload, packets also include some redundant bits to aid error detection. For example, a single extra bit might be added to every packet so that all transmitted packets have an odd number of 1 bits. If a packet arrives with an even number of 1 bits, it can be recognized as having been corrupted in transit and discarded so that the sender will retransmit it. Such extra bits can't guarantee that every packet received is correct. But they do guarantee correct transmission with overwhelming likelihood, and from a practical standpoint, this process suffices to make the likelihood of an undetected error less than the likelihood of a catastrophe, such as a meteor strike, at the source of the transmission.

IP, the best-effort packet forwarding protocol, can also be used for delivering messages imperfectly but quickly. For example, think about how the Internet might be used to convey voice communications, such as telephone calls. The voice signal can be chopped up into small time slices, each digitized and sent over the Internet. But instead of using TCP, which guarantees delivery but not timeliness, a different protocol, called UDP, is used. UDP accepts some packet loss in exchange for speedy delivery. Voice tones change slowly enough from one instant to the next that packets of a telephone conversation can be scrambled a bit, and some could be omitted entirely, without causing the conversation to become hard to understand—as long as the packets that make it arrive at about the right rate.

Many other protocols have been designed for other purposes and to layer on top of these, using TCP and UDP to carry out more complex communications tasks. For example, Hypertext Transfer Protocol (HTTP) is designed for communication between a web browser on a user's computer and a web server anywhere else in the world. HTTP relies on TCP to retrieve web pages on the basis of location information such as lewis.seas.harvard.edu. So without knowing the details of how TCP operates, anyone could set up a web server that would deliver web pages in response to incoming requests.

Who's in Charge?

There are no Internet cops to force anyone to format their packets as TCP, IP, UDP, or other protocols stipulated. No one will throw you off the Internet if you put your source address where the destination belongs and vice versa. If your packets don't conform to the standards, they just won't be delivered, or they will be ignored if they are delivered.

The Internet does, however, have some governing authorities. One is the Internet Engineering Task Force (IETF), which establishes the standards for Internet protocols. The IETF is a remarkable organization. It is open to anyone who wants to join, and it makes decisions on the basis of "rough consensus and running code." In earlier years, the IETF would meet in a room and determine "rough consensus" by having members hum. Substantial majorities were evident to everyone, and individual preferences enjoyed a level of anonymity because in a large group it is hard to tell who is humming and who isn't. Because most changes to the Internet protocols are enhancements and additions that do not change anything that is already working, there is rarely a need to make a positive decision under time pressure; the IETF can defer decisions, let people talk more while tweaking their proposals, and wait for true consensus to develop.

So the Internet is *open* by design. Anyone can join the decision-making process. You would not be wrong to be reminded of the communal utopianism of the 1960s. Early IETF member David Clark famously said, "We reject kings, presidents, and voting. We believe in rough consensus and running code"—the last phrase indicating the engineer's preference for proofs of concept over concepts alone.[9] Of course, once the Internet became widely adopted, you would need to do a lot of persuading to develop a consensus to change anything that had become important to lots of people. But if you and I were halfway around the world from each other and decided to develop our own secret protocol for (say) trans-Pacific xylophone duets, we could happily program our computers to exchange IP packets that no one else would know what to do with. The IETF explains its role this way in its mission statement:

> When the IETF takes ownership of a protocol or function, it accepts the responsibility for all aspects of the protocol, even though some aspects may rarely or never be seen on the Internet. Conversely, when the IETF is not responsible for a protocol or function, it does not attempt to exert control over it, even though it may at times touch or affect the Internet.[10]

This is a remarkable statement, and it shows how badly the "Information Superhighway" metaphor breaks down when applied to the Internet. If the Internet is a highway, it is one in which motor vehicles voluntarily adhere to certain conventions so they can share the highway safely, but bicyclists and skateboarders are welcome to use the roadways, too—though at their own risk.

The Internet is open in another direction as well. Just as IP serves as the base layer for a hierarchy of protocols, IP itself is a logical, not physical, protocol. Internet packets can be transmitted on copper wire, through fiber-optic cables, or by radio waves. If you are an ordinary personal computer user buying something on Amazon, it is likely that the packets going back and forth between you and Amazon pass through all three and more, as they move from your computer to your wireless router, to your ISP, through the Internet, into Amazon's corporate network, and to one of Amazon's computers. Whenever engineers come up with a new way to move bits through physical media, they can also develop an implementation of IP that runs on that physical medium. There is even a carrier-pigeon protocol that could, in principle, be used to implement IP.

IP, the format in which all packets pass through the Internet, plays a role like the design of the ubiquitous 120V electric outlet, with three holes of specified shape and dimensions. The electric source on one side of the outlet may ultimately be a hydroelectric dam hundreds of miles away, solar panels only a few feet away, or battery storage. As long as the electricity conforms to the standards, the outlet is doing its job. The devices that get plugged into the outlet can be refrigerators, toothbrushes, vacuum cleaners, or dental drills.

As long as a device is fitted with the right plug and is designed to run on standard alternating current, it will work. In the same way, Internet Protocol acts as a universal mediator between applications and physical media.

In fact, the standardization of IP is the reason the Internet has so many uses that were not initially anticipated. Zoom and Facetime—Internet applications for connecting people via live audio and video links—were built on IP, even though there was absolutely nothing about such services in the original Internet design. The inventors of the Internet telephone system Skype—a small group of Scandinavian and Estonian engineers—just needed to adapt the Internet protocols to their purposes. And they did not need to ask the permission of the IETF or any other authority to start using Skype or to encourage others to start paying them to use it.

The Internet Has No Gatekeepers?

Now it has always been an overstatement to say that the Internet has no gatekeepers, but it is less true now than it used to be. As we will soon see, in some countries, governments are the primary gatekeepers, and in others, such as the United States, private corporations assume gatekeeper roles. Let's start with the forms of gatekeeping that have long existed.

Names to Numbers: What's Your Address?

The first fact of Internet life is that it does no good to be "on" the Internet if no one can find you. Packets flowing through the Internet have numeric addresses. Some entity has to translate the symbolic names—like cornell.edu and Skype.com—into numbers and keep track of which connecting points have which numbers.

The Internet Corporation for Assigned Names and Numbers (ICANN) is the body that decrees which numeric addresses are assigned to Cornell University or the nation of Australia. It oversees publication of electronic directories in such a way that anyone sending an email to the address president@cornell.edu or retrieving a web page from an address such http://anu.edu.au (the home page of the Australian National University) is directed to the correct place on the Internet. The translation tables, from letters to numbers, are held on Domain Name System (DNS) servers, which other computers consult in order to look up the numeric addresses to insert in the "destination" field of IP packets before they are launched into the Internet. If the Internet has a single vulnerability, it is control over the DNS servers. Does the island nation of Tuvalu get its own Internet top-level domain, like .au for Australia? (It does—and a very valuable one at that. It's .tv, and the nation, which used to make money from selling postage