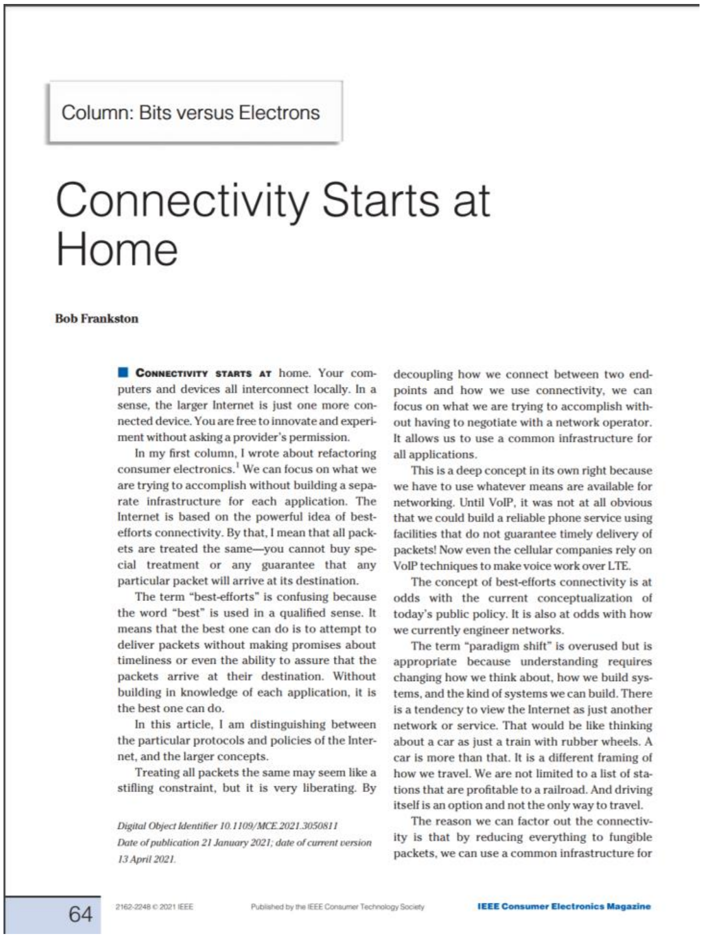
Connectivity Starts at Home



Column: Bits versus Electrons

Connectivity Starts at Home

Bob Frankston

■ Connectivity starts at home. Your computers and devices all interconnect locally. In a sense, the larger Internet is just one more connected device. You are free to innovate and experiment without asking a provider's permission.

In my first column, I wrote about refactoring consumer electronics.[1] We can focus on what we are trying to accomplish without building a separate infrastructure for each application. The Internet is based on the powerful idea of best-efforts connectivity. By that, I mean that all packets are treated the same—you cannot buy special treatment or any guarantee that any particular packet will arrive at its destination.

The term "best-efforts" is confusing because the word "best" is used in a qualified sense. It means that the best one can do is to attempt to deliver packets without making promises about timeliness or even the ability to assure that the packets arrive at their destination. Without building in knowledge of each application, it is the best one can do.

In this article, I am distinguishing between the particular protocols and policies of the Internet, and the larger concepts.

Treating all packets the same may seem like a stifling constraint, but it is very liberating. By decoupling how we connect between two endpoints and how we use connectivity, we can focus on what we are trying to accomplish without having to negotiate with a network operator. It allows us to use a common infrastructure for all applications.

This is a deep concept in its own right because we have to use whatever means are available for networking. Until VoIP, it was not at all obvious that we could build a reliable phone service using facilities that do not guarantee timely delivery of packets! Now even the cellular companies rely on VoIP techniques to make voice work over LTE.

The concept of best-efforts connectivity is at odds with the current conceptualization of today's public policy. It is also at odds with how we currently engineer networks.

The term "paradigm shift" is overused but is appropriate because understanding requires changing how we think about, how we build systems, and the kind of systems we can build. There is a tendency to view the Internet as just another network or service. That would be like thinking about a car as just a train with rubber wheels. A car is more than that. It is a different framing of how we travel. We are not limited to a list of stations that are profitable to a railroad. And driving itself is an option and not the only way to travel.

The reason we can factor out the connectivity is that by reducing everything to fungible packets, we can use a common infrastructure for

Digital Object Identifier 10.1109/MCE.2021.3050811
Date of publication 21 January 2021; date of current version 13 April 2021.

64 2162-2248 © 2021 IEEE Published by the IEEE Consumer Technology Society IEEE Consumer Electronics Magazine

Introduction

Connectivity starts at home. Your computers and devices all interconnect locally. In a sense, the larger Inernet is just one more connected device. You are free to innovate and experiment without asking a provider's permission.

In my first column, I wrote about refactoring consumer electronics. We can focus on what are are trying to accomplish without having to build a separate infrastructure for each application. The Internet is based on the powerful idea of best-efforts connectivity. By that, I mean that all packets are treated the same – you can't buy special treatment or any guarantee that any particular packet will arrive at its destination.

The term "best-efforts" is confusing because the word "best" is used in a qualified sense. It means that the best one can do is to attempt to deliver packets without making promises about timeliness or even the ability to assure that the packets arrive at their destination. Without building in knowledge of each application it's the best one can do.

In this article, I'm distinguishing between the particular protocols and policies of the Internet, and the larger concepts.

Treating all packets the same may seem like a stifling constraint, but it's very liberating. By decoupling how we connect between two endpoints from how we use connectivity, we can focus on what we are trying to accomplish without having to negotiate with a network operator. It allows us to use a common infrastructure for all applications.

This is a deep concept in its own right because we have to use whatever means are available for networking. Until VoIP, it was not at all obvious that we could build a reliable phone service using facilities that don't guarantee timely delivery of packets! Now even the cellular companies rely on VoIP techniques to make voice work over LTE.

The concept of best-efforts connectivity is at odds with the current conceptualization of today's public policy. It is also at odds with how we currently engineer networks.

The term "paradigm shift" is overused but is appropriate because understanding requires changing how we think about how we build systems and the kind of systems we can build. There is a tendency to view the Internet as just another network or service. That would be like thinking about a car as just a train with rubber wheels. A car is more than that. It's a different framing of how we travel. We aren't limited to a list of stations that are profitable to a railroad. And driving itself is an option and not the only way to travel.

The reason we can factor out the connectivity is that by reducing everything to fungible packets, we can use a common infrastructure for all connectivity, whether we are communicating images, text, etc. It means accepting our inability to ask for guaranteed delivery. It also means understanding how we can use those packets (or set of bits) to exchange ideas and references.

The traditional engineering paradigm of layering abstractions and building subsystems for a purpose is replaced by thinking in terms of resources and opportunities.

The term "Internet of Things" has become popular, but what is it? The papers I get to review tend to be focused on building with special radios and configuration. That's very old-school. The opportunity is to assume connectivity and focus on what we can do if we have connectivity and then, separately improve our capacity to exchange packets.

A Consumer Mindset

The ability to focus on the application at hand rather than the network is at the heart of consumer technology. We should think about the user experience rather than the needs of a network provider.

I was fortunate in that my first job required me to build tools for untrained users so they could craft their own solutions. It meant thinking in terms of the user experience and thinking outside narrow use cases.

I applied this principle in the implementation of VisiCalc (http://VisiCalc.us/). Dan Bricklin and I focused on the user experience. The users should forget that they were using a computer. While the 16 kilobytes (kilo, not mega or giga) limit did force tradeoffs, we never lost sight of empowering the user.

I applied this same principle to home networking. It had to work without any special setup, nor could it require a professional installer. This is why today you can go to the store, buy a router, and expect your computer to just connect - well, almost.

You still need to tap into the legacy model and get connectivity from a provider and make sure you provide the right Wi-Fi credentials. These are barriers to consumers who want things to “just work”.

An Internet (of sorts) at Home

Within my home, I rely on IP – Internet Protocol – for connectivity. Looking back, I was influenced by my experience with Ethernet (as per my first column) and took a very simple approach of fungible packet connectivity. It didn’t matter what kind of wires and radio were used – all endpoints were equally accessible. This made things very simple and enforced the separation of the application from the facilities used to exchange packets.

One problem with the IP protocol is that the addresses are issued by a provider. In the 1990s, those numbers were billed like phone numbers with a monthly fee for each one. The rationale was simple. An IP connection (via the broadband cable) was treated as a one-for-one replacement for a dialup connection.

I didn’t buy into that use case. I had been experimenting with Network Address Translation (NAT), an existing technology that would allow the entire home network to look like a single computer to the provider. The NAT allows us to innovate in our homes without asking a provider for (and, in the past, paying for) a pubic IP address for each device. It allows our connectivity to start locally rather than being entirely dependent upon a provider.

A single IP address would be shared among all the computers in the house by taking advantage of the fact that the IP address is really a 48-bit address – a 32-bit portion that is the IP address and a 16-bit port (and socket number) number. The concept dates back to mainframes in which each endpoint was a service within the computer. A known port would be assigned to service. For example, port 25 would connect to the single mail server (SMTP or Simple Mail Transfer Protocol) on the shared mainframe. Socket numbers used that field and were assigned dynamically for each connection.

Within the local (home) network, each machine is assigned a local IP address. On outbound connections it would appear to as a socket number on the shared external IP address. The downside is that your computer would be invisible to the outside world because it wouldn’t have a public address. The workaround is to set up a mapping, port forwarding so that you can map a public port to an internal address. Thus, you can map public port 22 (used for SSH – encrypted terminal connections) to port 22 on an internal machine to provide command-line access. You could also assign port 122 and map it to port 25 on a different machine.

If this seems kludgy, it’s because it is kludgy. That’s OK in that it can be contained. All you need to know is that port 122 connects you to the machine controlling your water sprinkler. You can then get on with the task of managing the sprinkler.

Another problem is the challenge of finding the IP address of your home system. The classic solution is to use a static IP address, but that requires careful management, and such addresses may not be available from your provider.

Instead, I use dynamic DNS (DDNS) names. This is an adhoc protocol which updates the DNS record via an API. Your internal system updates the DNS record when it discovers that the public IP address has changed. I purposely avoided static addresses so I could assure the vaiability of of such dynamic addresses. It helps that my public DNS file is managed by a friend who innovated in other ways to address the limits of dynamic addresses.

In a sense, the NAT is like the red/green interface to the phone network I wrote about in the last column. It allows me to innovate outside the provider’s facilities. In telephony, it took US Supreme Court decisions to legalize what many of us had already been doing on our own.

With the Internet, we didn’t need to ask for permission to innovate though it did violate the terms of service of the providers in the 1990s that said we aren’t supposed to share the connection among computers nor offer services (such as a webserver) in our home. Considering that the

Internet is a peer network, such requirements were wildly out of sync with the basic reality of the Internet.

I wasn't happy about using the NAT. If IPv6 were available, then each local network could have gotten its own IP addresses. My intent was to make the router[i] (the device between your home network and the world that also, typically, hosts the NAT) evolve into an IPv6 gateway, but that never happened because the accidental properties of the NAT became features and because it works for today's applications.

One accidental byproduct of the NAT is that, by default the internal machines aren't visible to the outside world. The NAT itself isn't a firewall but acts as one. This is a limitation that is treated as a feature.

PCs evolved in isolation, so the software hadn't been designed using the practices that made it safe for conecting directly to the rest of the world. Making them inaccessible provided a degree of isolation and safety. I had expected that problem to have been addressed when IPV6 would become the norm, and all connections would be encrypted with IPSEC.

Instead, the firewall became a feature, and increasingly complex firewall rules bedevil the network by creating obstacles in the path. I can't just make a service available. I have to figure out which rule is causing the blockage.

The other reason is the creative ways to work around the limitations. A device inside your home can register with its own external server and be available. It didn't need its own address for direct availability. Sure, that is far from ideal; but it works. Or seems to. One of the problems is that this co-evolution doubles down on use cases; thus, we get more of what works for existing applications, and it is increasingly difficult to innovate outside those design points.

In particular, we have an Internet that is more and more about the web. Everywhere the DNS names fail by default unless one remembers to renew the names. How do you create a stable community of things in such an environment? The things don't have their own identities and only work to the extent that each one has a special workaround. This means that many of the devices in your home fail if you lose your connectivity to the outside world.

I am very aware of this because I can experiment with direct connectivity inside my house where, thanks to the NAT, I have a way to experiment with approaches that apply to the larger world. I can continue to figure out how to make a more consumer-friendly Internet that is less about providers and more about innovation.

I believe that had my approach of evolving the NAT to support V6 locally gotten support we would be using IPv6 more commonly. With the NAT providing V6 tunneling over V4, we could use V6 without waiting for providers to implement it. As of now, Verizon still doesn't provide V6 to FiOS customers.

But that would only solve part of the problem. Ultimately the idea of endpoints being assigned unique addresses by a central source is at odds with the more decentralized approach I have taken within my home network. It would also presume an Internet where the endpoints are physical devices or services within well-defined systems. I see an Internet that is are more abstract. The endpoint might be "lighting," which could involve opening the shades rather than turning on a single bulb.

It's an Internet in which my medical device is an endpoint that is reachable anywhere without being located on a particular provider's network or a particular home network.

But that will have to wait for another column. I've come to realize that my view of the Internet goes well beyond today's implementation. I won't want a designated network. I want the ability to connect between endpoints, both physical and abstract while factoring out the complexity of exchanging packets.

What is clear is that there are two stories. The story of what I can do by assuming connectivity and the story of how to create more opportunity for innovation by moving the Internet from being a provided service to being infrastructure. And this column is where I explore this new landscape.

[i] Purists will point out that it isn't really a router in the technical sense but that's their problem. In this case the use of the vernacular doesn't cause collateral damage.