

(Not) Getting the Message Across

Bits Versus Electrons

(Not) Getting the Message Across

By Bob Frankston

If I'm in a doctor's office, I can wear a sensor (perhaps a bracelet) that can be wired to a monitor in the office. I can replace that wire with an Internet connection, and the doctor can continue to monitor me anywhere I travel throughout the world. All I require is the ability for a best-efforts exchange of bits. I don't need any intermediaries that implement a special medical message format or a medical gateway. I just need an unfettered way to transport generic packets.

This is the big idea behind the Internet. There are just raw packets between the two end points. Best efforts means that some packets may be lost so it's the application's responsibility to be resilient—I don't depend on special treatment between the two end points because my bits are not more equal than others'.

At least in theory. The current implementation of the Internet has some issues, which I've covered in previous columns. The important point for this column is the ability to focus on the relationships without depending on intermediaries that understand the messages. Everything is normalized to packets.

We can contrast this with the business model of telecommunications, which treated messages as freight with some charges based on the value of the message. We still have some of the old model, such as when Comcast charges Netflix for giving their content priority. But change is coming as we increasingly use IP as a transport for voice and other content.

Digital Object Identifier 10.1109/MCE.2014.2317938
Date of publication: 27 June 2014

COMING HOME

In this column, I'm looking closer to home—the devices and protocols that try to understand us by embedding purpose. By this, I mean that the chips and devices along the way need to know that I'm exchanging information about my heart rate. This makes me dependent upon each intermediary cooperating as well as standards committees having their say before I can try out a new idea. The end-to-end connectivity of the Internet is not just about access; it's as much about simple connectivity locally.

The important point for this column is the ability to focus on the relationships without depending on intermediaries that understand the messages.

At the January 2014 IEEE Conference on Consumer Electronics (ICCE), I was reminded of the problem of embedding purpose in devices and infrastructure. The moderator mentioned that there were no wireless microphones available for those asking questions in the audience. Yet we all had very powerful computing devices—our smartphones. They could act like microphones simply by creating a path for the bits from the source in the phone to the room's speakers.

This is a very simple concept. The difficulty is getting past all the intermediaries that are built for a particular purpose or are trying to add value by requiring the use of a particular message format or profile.

A traditional microphone has a wire that goes from the microphone to a switchbox so that the moderator can select which microphone is active. A wireless microphone merely substitutes a frequency band for that wire. You have to have a purposeful-built device, the microphone, as well as a wire (or virtual wire in the form of a radio band) to give people in the audience a way to be heard.

We could be creative and just place a phone call to the moderator and he could hold the phone to his microphone and create a workable path for the speech. In fact, I did just that when I spoke at GlobeCom in 2012 by having a speaker in Brazil call me in Los Angeles.

Using Bluetooth as a wire replacement might work better than holding my phone to a microphone, but it doesn't give me simple end-to-end connectivity between end points. What if I want to make a connection that isn't already supported by Bluetooth?

Once we recognize that the function of the device is defined in the software, we do not need to have a microphone. We just need generic hardware and an unfettered path for bits. This is similar to using voice over IP (or other packet protocol) rather than having an infrastructure dedicated to voice phone calls.

who understand the messages. Everything is normalized to packets.

We can contrast this with the business model of telecommunications which treated messages as freight with some charges based on the value of the message. We still have some of the old model, as when Comcast charges Netflix for giving their content priority. But change is coming as we increasingly use IP as a transport for voice and other content.

Coming Home

In this column I'm looking closer to home – the devices and protocols that try to “understand” us by embedding “purpose”. By this I mean that the chips and devices along the way need to know that I'm exchanging information about my heart rate. This makes me dependent upon each intermediary cooperating as well as standards committees having their say before I can try out a new idea.

The end-to-end connectivity of the Internet is not just about “access”, it's as much about simple connectivity locally.

At the January 2014 ICCE (IEEE Conference on Consumer Electronics) I was reminded of the problem of embedding purpose in devices and infrastructure. The moderator mentioned that there were no wireless microphones available for those asking questions in the audience. Yet we all had very powerful computing devices – our smartphones. They could *act like* microphones simply by creating a path for the bits from the source in the “phone” to the room's speakers.

This is very simple concept. The difficulty is getting past all the intermediaries that are built for a particular purpose or are trying to add value by requiring the use of a particular message format or profile.

A traditional microphone has a wire that goes from the microphone to a switchbox so the moderator can select which microphone is active. A wireless microphone merely substitutes a frequency band for that wire. You have to have a purposeful-built device, the microphone, as well as a wire (or virtual wire in the form of a radio band) in order to give people in the audience a way to be heard.

We could be creative and just place a phone call to the moderator and he could hold the phone to his microphone and create a workable path for the speech. In fact I did just that when I spoke at GlobeCom in 2012 by having a speaker in Brazil call me in Los Angeles.

This July 2014 IEEE CE Column. You should read this article on the IEEE Site ([here](#)). My previous column on HTML5 is available as <http://rmf.vc/HTML5>.

If I'm in a doctor's office I can wear a sensor (perhaps a bracelet) that can be wired to a monitor in the doctor's office. I can replace that wire with an Internet connection and the doctor can continue to monitor me anywhere I travel throughout the world. All I require is the ability for a “best efforts” exchange of bits. I don't need any intermediaries that implements a special medical message format or implement a medical gateway. I just need an unfettered way to transport generic packets.

This is the big idea behind the Internet. There are just raw packets between the two end points. “Best Efforts” means that some packets may be lost so it's the application's responsibility to be resilient – I don't depend on special treatment between the two end points because my bits aren't more equal than others'.

At least in theory. The current implementation of Internet has some issues which I've covered this other columns. The important point for this column is the ability to focus on the relationships without depending on intermediaries

Using a Bluetooth as a wire-replacement might work better than holding my phone to a microphone but it doesn't give me simple end to end connectivity between end points. What if I want to make a connection that isn't already supported by Bluetooth?

Once we recognize that the function of the device is defined in software we don't need to have a "microphone". We just need generic hardware and an unfettered path for bits. This is similar to using voice over IP (or other packet protocol) rather than having an infrastructure dedicated to voice phone calls.

In order to achieve this simplicity we must get past the idea that we need a specialized gear for each purpose. Why have a different kind of wire for SATA, USB, HDMI, etc. when they are just different ways packets of bits? Not only does each have its own specialized cables, each one also requires its own special software in the form of drivers.

When I connect a device to an Ethernet (or over Wi-Fi) I can connect to it using software on any machine. If I use USB I can't use the device unless there is a driver available for the particular operating system on the particular hardware I'm using. And if I move the device to another machine it becomes unavailable. The USB protocol has other "smarts" in assigning devices to class hierarchies and, for some, precise timing that limits the reach of USB.

One might argue that these constraints on USB give us a sense of security in limiting access to devices yet our computers are shared environments in the same way our networks are. By allowing us to focus on the relationships between an application as an end point and a device anywhere we can be explicit in our policies.

Radios

We see another face of "smarts" in the many purpose-built chips. At that ICCE conference there was a discussion of the future of mobile. One topic was the problem of having so many antennas in the phone to handle the different radios.

Just as we have a cable per purpose we have radios that are trying to help us by "understanding" each message so that it can pass it along. We see this in Bluetooth radios which have a profile for each application. Because the chips and the profiles are designed together it takes years of committee meetings to add new capabilities. Then there must be a heavy effort to adopt the new chipsets in order to amortize the costs. This is very different from the rapid learn-by-doing approach we see for Internet protocols.

Once we have the profiles established we need to use security policies built into the chips – the pairing of radios. You can see the problem when two people are trying to

share the facilities in a car – the radio will pair with just one phone. Why such a restrictions? Why not have policies which can be chosen by the user and modified as we seek to improve the user experience?

We can see another problem with Bluetooth, Zigbee and other such protocols when we have two devices that work only when they are near to each other. We become dependent upon protocol-aware gateways to extend the range and lose the transparency that has made the Internet so vibrant. You may have noticed the problem when you are using a Bluetooth headset and wander away from your phone.

This can become a deadly problem in the case of a medical device when it only works if properly paired with the phone that happens to be nearby. And it must use standard message protocols agreed upon before the chips are deployed even though it is only after deploying the chips that we gain the experience we need to do practical protocols. If a pre-paired phone isn't available, if the service provider isn't available, if there is a problem with the account or if any number of other conditions can't be met then the bits fail to get through. Why is failure the default?

This message-passing approach was similar to the Internet proxies that used to be common. Instead of a transparent connection between a corporate network and the rest of the Internet each application would be built into the connecting computer. This approach doesn't scale – you need to wait for each application to be implemented in the proxy before you can use it. Today we see use of port 80 as a tunnel through such proxies (and firewalls). This is a testament to the power of the Internet as a concept and a reminder of the need to work-around such "features".

It's easy to understand the rationale of smart chips as each one was being built to solve a particular set of problems as per the tradition of consumer electronics in selling devices whose value comes from their application.

But there is another reason – the perceived need to make systems that work despite extreme limits, especially the need to use very low power.

We can learn from the history of database. Early database systems were meant to be very efficient and took advantage of the particular characteristics of the hardware. Embedding a disk address in the database could save a few steps. In many cases a part number or user id (as with CompuServe) was a direct index into the database.

Relational databases had none of these optimizations but in return you got schemas which allowed schematic evaluation of the query and thus simply avoided doing many of the operations at all while providing database designers with the flexibility to improve the design of databases. The

query processing improvements could be shared by all rather than having to be redone for each database.

I view the radio optimizations in the same way. We don't want to ignore the particulars of radios but rather characterize them at an application level. Thus if we have a low capacity link then we need the application to adapt to such a limitation end to end. We can see a possible approach with CoAP (Constrained Application Protocol) which provides a variation on HTTP for environments with limited radio capabilities.

Bluetooth Low Energy radios have the option of providing open connectivity and passing through IP. Making that capability the norm can give us a way to enable devices to operate without having to carefully establish pairwise relationships.

This isn't to say that we can never take advantage of special cases. We can harvest power to send a short message to a nearby receiver. Such clever hacks are useful but we need to recognize that they are the exception and we need to be careful to understand the limitations. It's also important to recognize that the Bluetooth radio can be a useful resource.

But there is a problem when we have solutions tied to the accidental properties of the Bluetooth radio vs. the Zigbee radio and others as each association attempts to sell the chips by the value of the application rather than as just commodity radios.

We also have to be aware of the business model of telecommunications implicit in having chips that implement GSM, CDMA and other cellular protocols. This an issue I've written about in previous columns. Such business models create barriers to simple connectivity because they have to make failure the default in order to be able to bill for those bits that are allowed through.

Just as the radio technology is implemented in the context of business requirements, we need to push back on business models which go against the opportunities we can create with our advancing understanding of technology.

The medical monitoring application in particular demonstrates the cost of continuing business as usual. It may be dramatic to say that current policies cost lives but that is the case. We can't create a separate policy for medical devices because there is no sharp dividing line between emergencies and day to day use.

If we are to get our message across we need to move from a message-aware approach to one that brings the Internet's packet level connectivity back home to our devices!

