Beyond the Interweb

From Licklider to the Internet

I'm writing this after having a Zoom™ webinar/conversation with IEEE CE members and others.
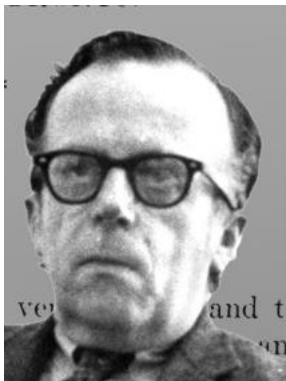

Figure 1 JCR Licklider (from Spectrum)

JCR Licklider[i] would've been thrilled to see such a powerful man-machine symbiosis having become so normal and having it work so well. Lick, as he was called, can be considered the grandfather of the Internet. He was cofounder of MIT's Project MAC. In 1963 he wrote about an Intergalactic Network[ii] that would interconnect the world's computers.

Lick was an acoustic psychologist, not a computer scientist nor engineer. His focus was on how the technology could empower people. It's telling that I came across his research while taking a class in psycholinguistics. I read a paper he wrote when he worked at IBM in 1949 about what kind of radios worked best in noisy environments. Some radios seemed worse in quiet environments were better against a noisy background.

Another mentor, Seymour Papert, studied how people learn to learn and form concepts. He was a student of Jean Piaget who studied the stages that children went through in conceptualizing the world. Or don't. As Papert pointed out people don't learn but societies do by having people who understand the new conceptual framework come to the fore.

When I first tried to explain the Internet and Consumer Electronics in my column[iii] I didn't fully appreciate how my take differed from the common view even among those well-versed in the concepts. To me it is obvious that we had transcended the original design point of a network of networks and that, at least architecturally, we could view it as a flat address space independent of the accidental topology of the earlier networks.

I had long viewed networks as simply a means and not important in and of themselves. I was first online in 1966 as part of my job – helping to build an online service for financial analysts. As with networks I view programming as a means of creating an experience. (OK, I also see it as fun, but I don't forget the larger context). Taking the terminal home that summer meant I could explore computing rather than just focus on assigned projects.

Figure 2 BangleJS platform

This is why I'm excited about my latest toy, sorry, research platform, the BangleJS watch, which is programmed in JavaScript. This makes it easy to create small applications without worrying about all particulars of the hardware.

The software happened to run on a particular device and take advantage of the display, but it's not embedded. It just happens to be using the wrist device (OK, watch if you want to use conventional wording) as a platform. Part of the function, though, is its connection to the rest of the world. A simple line of code can fetch the current weather so I can display it.

OK, I'm fudging a little since I haven't gotten the remote weather fetching to work because the device only supports

Bluetooth and no simple IP connectivity. Bluetooth is TDS (Too Damn Smart). Its profiles have application-specific knowledge. What I need is simple IP-connectivity so I can just present a destination address and be done with it instead of having to carefully pair with devices along the path. For now, I can stub out the fetch and solve the problem later. This is my normal style of kneading code till it works rather than having to design it as a whole.

This approach of kneading code into shape works well for software. I remember when I first explicitly adopted it in 1969. I was at a computer terminal and was about to sketch out the code on paper and asked myself why I was doing it when I had a computer to help me. It was a typewriter style terminal (an IBM 2741) so I couldn't draw a diagram, but I could put together ambiguous concepts, AKA, call subprocedure stubs without first defining them. It's a middle out style in which I make some assumptions and flesh them out and then rework, rearchitecture or knead the code into shape.

It was with this mindset that I approached packet networking in 1973 when we studied ALOHAnet in class. As I wrote in my 2013 column, Bob Metcalfe had to convince his advisors that Ethernet (Aloha on a coax) would perform well as a telecommunications network whereas, for me, it didn't matter as long as I could play with it and do more than I could with lower performance paths. Ethernet provided a new opportunity.

I chose the word "paths" rather than "networks" to avoid confusion. A wire is not a network – but I can use it to do networking and we can increase the reach by having a standard convention for relaying packets from one wire to another (or by using radios).

Classic telecommunications networks were built by telegraph (and later, telephone AKA talking telegraph) companies as a means of providing their own services. The telephone itself was called CPE or Customer Premises Equipment and was part of the network. In the 1950's you weren't even allowed to put a box around a phone because it interfered with the service[iv]!

Notice that I use anthropomorphic terms as in comparing subroutines with concepts. While the brain is vastly different from a traditional computer the conceptual framework is similar. "Meaning" is the result of interpretation by our brain in context rather than being intrinsic. Understanding typically requires iteration. The eye is not a simple camera that records all pixels. The eye has evolved to provide the brain with key cues.

Traditional telecommunications providers add value by assuring all bits of a message are carried intact by reserving a channel and taking responsibility for reliable transport of messages as freight. With intelligence in our devices we can preserve all bits even if a packet is lost by resending it if there is no acknowledgment.

That computer power allows for innovation. Instead of waiting for a packet to be resent we can program around the missing packet thus allowing for streaming without depending on intelligence in the network.

Having intelligence in our device allows us to take advantage of opportunities rather than requiring guarantees. The reason it works is because we have intelligent devices at the endpoints. I didn't fully appreciate the full power of best efforts until I did my home networking project at Microsoft and tried to understand why the concept was so powerful.

The value is created by innovating rather than paying a "provider" for promises[v]. This is why VoIP (Voice over IP) was discovered outside traditional telephony. If value is created by reserving channels, engineering is focused on improving the channels rather than removing the need for them. It's telling that LTE was to be data only until VoIP showed it could carry voice.

VoIP is only one example of the innovation we get when we are not dependent upon a provider in the path providing a clear channel. Traditional networking (as a service) is built on the idea of reliable delivery.

If all we need is best efforts, then we don't need a provider that owns and controls the entire path. We can composite the path out of locally owned facilities.

What we do need is the kind of simplicity I got with ALOHAnet and Ethernet in which the two computers could directly exchange packets. The reason Vint Cerf and Bob Kahn are considered the fathers of the Internet is that they found a way to extend this simplicity beyond the local network using TCP. To me the Internet was no longer a network-of-networks but a vast common space in which endpoints can simply connect.

What I did not appreciate is the degree to which the network-of-networks concept is still at the core of how people think about the Internet and connectivity.

I use the term, Interweb, for the Internet as seen through the lens of the web. That does work relatively well once things have been set up. The user is there to click "agree" and to maintain a cell phone account. The site operator makes sure the DNS (Domain Name System) entry is renewed each year. It also works for connective devices carefully engineered for a purpose or tethered to a cellular network.

It works if users accept complex onboarding procedures. It works if we accept limitations like being unable to print because the printer is on the wrong network. It works if engineers accept that carefully crafted applications only work in one context. It works if we accept the idea that we must depend on network professionals.

I do not.

Zooming Ahead

When I was introducing home networking at Microsoft my requirement was no installers. People should be able to use it out of the box. Alas, there is still some setup, but home networks show that we needn't accept all the complexity.

I composite systems out of pieces of software and hardware and can't be tied to fixed topologies at the mercy of network operators who limit me to what is profitable to them.

Once we understand best-efforts and recognize that we don't need special-networks, we are ready to embrace the potential. We get abundant opportunity to innovate.

We are ready to recognize that the givens are not given.

- ⇨ We need stable linkages rather than domain names that expire every year. The IEEE can take a concrete step in that direction by offering names that are stable, like bit.ly does, but for devices. Perhaps an IOT.IEEE domain.
- ⇨ We need to pay for the facilities as common infrastructure rather than depending upon a provider making a profit by limiting transport to subscribers in good standing.
- ⇨ We need protocols that are transparent rather than ones like Bluetooth are TDS (Too Damn Smart)

I know this is all doable because of my engineering experience, my experience studying how the brain works, and from my experience in business and economics.

Getting people to think outside the Interweb is a challenge because we tend to look for more of the same. Traditional service-based metaphors lead us to think of something that is delivered through pipes and we worry about others stealing our Wi-Fi. If we are to realize the potential for connectivity, we need to go back to first principles – the need for people and their devices to communicate. The compromises and limitations we made for 19th century telecommunications no longer apply

By taking a user-centric view and creating a simple experience we now take Zooming[vi] for granted. Let's heed this lesson and move on rather than be limited by network-

centric thinking.

Figure 3 Zoom Concert

[i] https://spectrum.ieee.org/tech-talk/tech-history/dawn-of-electronics/untold-history-of-ai-darpa-dream-of-cyborg-intelligence

[ii] https://www.kurzweilai.net/memorandum-for-members-and-affiliates-of-the-intergalactic-computer-network

[iii] https://rmf.vc/IEEECE201301

[iv] https://en.wikipedia.org/wiki/Hush-A-Phone

[v] https://rmf.vc/PromisesVsDiscovery

[vi] Yes, I know Zoom is a trademarked name and we should talk about video conferencing, but I want to emphasize how the concept has become ordinary or, well, a meme.