

Why it's hard to understand the difference between the Internet and Telecom

I've been struggling with how to get people to understand the difference between the Internet and telecom. Whatever I say seems to get lost in the conceptual abyss that divides the world of telecommunications from the concept of the Internet. I often use the example of ordering vanilla ice cream with whipped cream and nuts and then getting told the restaurant is out of cherries because the waitperson heard "Sundae" not what I said.

I came to a realization at a recent ISOC meeting when trying to get the speaker to understand what I'm saying. It's not just him but ISOC as a whole. I realize that "Internet engineers" are far closer in their mission to a Telco engineer than to a user who just wants to exchange bits. They are trying to solve all the problems in the network because they are trying to be problem solvers.

I compare it with soldiers serving on the frontier and have more in common with the soldiers on the other side than with the people they are supposed to be serving.

Perhaps it's because I started as a user in the sense of online building systems where exchanging bits was just part of the job and not the job itself. Same for all of us in the 60's when this stuff was new. The Ethernet, for example, was just a graduate student's class project and not a network and the ESS#1 (the first electronic phone switch from ATT) was just another computer system. But in the same way we identified file systems and databases as components, we identified networks. It was easy for networks since we had already been using the term for telecom services and the Arpanet took care of all that network stuff.

What we didn't really recognize was that the Internet was something very different. It wasn't totally new but it crystallized a lot of previous efforts. We had radio packet networks including the AlohaNet. It wasn't really a network – just a set of protocols used by programs. It was the prototype for the Ethernet. The idea of unreliable packets came from [CYCLADES](#) in France.

But because we used libraries like TCP for reliable connections the application software could treat it as a network just like any other network. And because we were used to having leased lines and the institution (school or business) buy capacity in bulk it didn't really matter to most users whether we were using X.25 or TCP. They act-

ed the same for the application. This was even more true in the days of mainframes when they were all system functions but even with PC's where we might load TCP as a library it was still relatively transparent.

Furthermore the overall performance did improve when we improved the network – at least as long as it was done by increasing capacity rather than retransmitting at all levels. We also had low expectations so we didn't force voice onto the nascent Internet because we saw so much opportunity and didn't want what we couldn't pay for.

Thus people confused the simulation of telecom with telecom itself and missed the fact that the complete infrastructure of telecom had been replaced by an emulation on top of a best-efforts network.

But this has worked against us because we limit our expectations to what telecom could do. Thus we simply assume that the Internet is nothing more than telecom and accept that the relations are mainframe-mainframe and short lived connections. We accept that the idea is to connect everything and that mobility is hard and the exception. After all, there's only so much a network operator can do for us.

The idea of layering adds to this illusion since we consider the network to be a layer. Layering is a nice heuristic but fails dramatically because meaning is not intrinsic but instead comes from context. This is a deep philosophical concept and deserves a whole book. In fact Robert Laughlin touches upon it in *A Different Universe* but not as an explicit goal. It's key to evolution – I use the example of a gill bone become an ear bone by reinterpreting it in a different context. <http://rmf.vc/?n=Unbaked> also touches upon it.

The network is not a layer. That was the bug that the Internet addressed by eschewing reliable circuits but, as I note with regard to TCP, that point has been lost.

We fail to see that peer to peer is really the reinvention of the Internet from the edge in keeping with the original spirit of supporting applications rather than improving the network as such. But the P2P efforts fail to scale for at least two reasons.

The primary reason is that we are still focused on the exchange of bits as the primary purpose rather than on rela-

tionships as the design point. We see this in my TI EZ-Chronos which has a nice inexpensive radio but without the protocols that allow me do more than a single hop.

The other reason is the funding model of telecom that I keep mentioning. It means we can't assume that the devices are interconnected beyond their isolated communities so we don't address the problems of interconnecting communities not do we address the issues that arise when the devices are exposed to a toxic ecology outside their protected little islands. It doesn't help that people think that the Internet is about connecting everything to everything rather than seeing that global connectivity is not a realistic constraint any more than two-way links are.

It's not that there aren't examples. After all the Ethernet is not a network and UUCP is not a layer. But it's normal to force our understanding to fit our preconceptions and simply ignore whatever doesn't fit.

It doesn't help that we've also partitioned engineering from business and find it hard to recognize that the solution doesn't lie in engineering but in the kind of relationships we think of as market or business models. Perhaps that's Apple's big strength in seeing the whole. Too bad they then lock it down so you can't easily reuse the parts.