

On the Design of TCP/IP

(Ian Peter)

Following from feedback from Internet pioneer Bob Frankston about the nethistory.info site, the following email exchange with Vint Cerf, Bob Frankston and David Reed took place, on the subject of early tcp and ip separation. I'm reproducing it here with the permission of the participants.

It's fairly long, and essentially for technical readers. I've edited the content a bit to get to the original discussion sequence.

Bob Frankston

While I am biased, I notice that the histories rarely mention David (Reed of course -- there are too many Davids though David Reed is not exactly unique either) despite his role in the end-to-end concept and the related separation TCP from IP. You can ask the other participants in your list for their comments on this.

A number of the key concepts could be found in the design of the Ethernet though they arose more from imitating the AlohaNet than because of an explicit end-to-end design point. What makes it interesting is that it the idea of putting a reliable transport didn't even come up. I remember sitting in class when Bob (Metcalfe - too many Bobs?) spoke about his project. Of course the computers, not the network, would take care of assuring that the message got through. The insight was in making this an explicit design point of the Internet since the big Arpanet had separate computers - the IMPs - that could perform all sorts of magic including assuring a reliable transport.

The defining concept of the Internet is the / between TCP and IP. Writing it as TCP/IP is unfortunate since they are very far apart. UDP/IP might be more appropriate though I just talk about TCP or IP depending on which is appropriate.

Vint Cerf

Hi Bob

As I recall, David was certainly a proponent of the end/end philosophy that drove TCP. He also thought that we should use random sequence number (64 bit?) rather than the Initial Sequence Number mechanism based on clocks - I was not in favor of the random method but it had the advantage that you didn't need a quiet time. I was uncomfortable with the potential for ISN collision though. I don't recall David's role in arguing for the split of IP from TCP - what I remember most vividly is Danny Cohen's argument for the need for a service that was fast if not reliable or sequenced (to handle real-time traffic such as voice).

David, can you fill in blanks here?

Vint

David Reed

I can fill in some blanks about splitting TCP from IP. There were a small number of proponents for specifying the Internet protocols as based on datagrams, with the idea of reliable in-order delivery being an "application layer" mechanism. This all came together at the meeting I attended in Marina del Rey. John Schoch, Danny Cohen and I each presented arguments to that effect, from different points of view. John argued based on the PUP architecture, which was an architecture based on datagrams, where streams were one option among many.

Danny argued for the idea that packet speech did not want retransmission, but instead just a sequence-numbered stream of packets, where non-delivery was an option because latency was the key variable and the application could fill gaps. I argued that non-connection based computer-

On the Design of TCP/IP (Ian Peter)

computer protocols, such as those we were developing for interconnected LANs, could be more general, and that end-to-end reliability was ultimately up to the endpoints to assure - for example, there were useful protocols that involved a packet from A to B, handing off of the request from B to C, and a response back from C to A were quite useful, and would be end-to-end reliable at the application level, while gaining little or no benefit from low level "reliability" guarantees. Other protocols, such as multicast, etc. were essentially datagram-oriented. I remember arguing quite strongly that you could support streams on top of datagrams, but by requiring a streams, you'd never get effective or efficient datagram services. Danny equally argued that reliable streams would create latency variability (jitter) where none was necessary. John Schoch argued that PUP was datagram-based, with streams built on top, and that architecture was quite effective.

The idea of the large, randomly chosen connection identifier/sequence number was a part of the technique used in the LAN-centric internetworking protocol "DSP" (Datagram/Stream Protocol, or Dave's Simple Protocol) I had developed for MIT to allow for both datagram-like and stream-like behavior to coexist, because the connection identifier would minimize the problem of collisions in sequence space so that one need not do a 3-way handshake to authenticate an incoming message. But this was a part of a more general argument I made that related to how one could achieve idempotence in protocols - the 3-way connection setup handshake in TCP was a way to prevent delayed duplicate SYN requests from having a non-idempotent effect, i.e. doing the same command twice by accident. In many cases, I argued, idempotence is best done by the application layer, since the command itself is idempotent (i.e. the "compare and swap" instruction in a processor is inherently idempotent, because a delayed duplicate command will not compare correctly). This was one of the first end-to-end arguments, i.e. an argument that the function (idempotence) should not be done in the network layer but at a higher layer, and began my conversation with Jerry Saltzer, since he was my adviser at the time.

As I recall, we 3 people, plus Steve Crocker, conspired to argue that we needed a datagram service so we could continue to do research on more general protocols, and in the heat of the argument proposed why not split out the addressing layer from the stream layer, just as we had split the TCP from the functions of the Telnet layer. (you may remember that I also was involved in changing the interaction of the TCP/Telnet layers to use a byte-oriented "urgent pointer" that was a pointer into the stream of bytes, rather than a general "process interrupt" mechanism that was being proposed, which was problematic because it embedded operating system process semantics into the network layer). In the heat of the Marina del Rey meeting, we 3 (sitting next to each other) agreed to push for splitting the TCP packet into two layers, the routing header and the end-to-end stream payload. This resulted in a sidebar meeting in the hall during the next break, where I remember it was Jon, you, Danny, me, Steve, and John Schoch, and you agreed that we should try defining how we'd split the layers, and see if the overhead would be significant. This resulted in 3 protocols, IP, TCP, and UDP (for us datagram nuts). Danny went off being happy that he could define a packet speech protocol layered on UDP, I went off happy that I didn't need to pursue DSP anymore, but could focus on how to use UDP for protocols like TFTP, which we built at MIT shortly thereafter.

Vint Cerf

Dave,

Thanks a million for this rendering - I had not recalled your involvement in the MDR event so this fills in an unintentional blank in my own recollection of this passage in Internet history.

David Reed

Glad to be helpful. As a young graduate student, this particular meeting was at the core of my interest, so the details are vivid in my memory. It was one of the best groups that I have ever worked with with

On the Design of TCP/IP (Ian Peter)

intense argument, but a focus on constructive results. This particular interaction taught me a lot about groups, about protocol design, etc.

Too bad there are so few opportunities out there for young students to participate in such processes. The "open source" community is the one area where that is still happening, and it's sad that DARPA, the current IETF, and industry don't get how to harness such things.

(in a separate email talking about French pioneer Louis Pouzin..)

Pouzin was really arguing for packet networking, not free datagrams, sans connections. X.25, which only had streams, but was a packet network, was a descendant of Pouzin's work. I think he invented the term "virtual circuit", which illustrates where his head was at. He couldn't break free from the "circuit" idea, though he liberalized it a lot. This is where Gallegher and the folks at today's LIDS got it wrong, too. They defined a network as a way to multiplex a bunch of streams/circuits onto a more efficient infrastructure. They were trapped by their mindset and their model of what a network was.

The real issues that bogged people down in those days were "flow control" and "reliability", whose very definitions (at least the definitions use by the networking community) were defined in terms of connections or stream or circuits. In fact, the term "congestion control" appears in no literature before the Internet design time-frame, because it was subsumed into "flow control".

Though it wasn't the best idea, the idea of the "source quench" message was invented to provide a crude mechanism of congestion control that could deal with datagram-only connections, and the current TCP windowing system was initially understood as only serving an end-to-end flow control function. So inventing "source quench" was a way of breaking a controversy that surrounded the idea that gateways between networks would have no way to signal buffer overflow back to the sources. It was only later that the community (Van Jacobsen played a role, I think) invented the idea that packet drops could serve as the sensor that closed the control loop about congestion with the source.

One MUST remember that we were designing the Internet protocols as *overlays* to interconnect existing networks with heterogeneous designs, but which had internal congestion control and reliability. This allowed the Internet protocols to focus on the big picture - the "end-to-end" functions.

The key goal was to make the gateways simple and stateless and stupid (that was the "best efforts" insight that Vint and Bob Kahn brought to the world, which was their unique contribution. Bob often said to me that I was one of the few who really got what he meant by "best efforts"). Many people want to rewrite the history as if the Internet Protocols were designed to be implemented directly on the wire, handling in a single layer all of the problems of wireless, wired, multicast, ... transmission. In fact, the engineering approach was to accommodate MANY DIFFERENT solutions at the lower layers, and to encourage heterogeneity. Which is why I and John Schoch shared a very different viewpoint from the BBN guys, which was different from Danny Cohen. John and I were "LAN guys" who were working on networks that were inherently message-oriented, computer-computer, multicast technologies. We wanted those advantages to shine through to applications, not turned into virtual dialup telephone calls at teletype rates. Danny was a media carriage guy, who was interested in conversational packet voice over unreliable channels.

To design a protocol with that diversity of undercarriage required us to resist all sorts of attempts by the "Router Guys" (like Tomlinson and McQuillen) to "optimize" for the same old ARPANET constraints (50Kb/s private lines connected by IMPs). Cisco's heritage is as Router Guys, and too many of the IETF guys these days are Router Guys. They are the new "bellheads".

On the Design of TCP/IP

(Ian Peter)

Vint Cerf

David, I think there is something incorrect about your rendition regarding Louis Pouzin.

Louis was the datagram guru. The other French guy was Remi Despres and he was the one who did the Reseau Communication Par Packet - a predecessor to X.25. The latter was developed jointly by Remi, Larry Roberts/Barry Wessler/Dave Horton/and John Wedlake. When Larry Roberts was building Telenet he asked what protocols to use and I suggested TCP/IP but he rejected that claiming he could not sell datagrams and that people would only buy "virtual circuits" (sigh).

Virtual Circuits were never in Louis' network world - he was all datagrams until you got to the end/end transport layer and there he introduced the voie virtuelle (virtual circuit) - not unlike TCP over IP. When another of Louis' team, Hubert Zimmerman, wrote the first OSI architecture spec, I think he had X.25 in mind as the network layer with virtual circuits built in. When I "called him" on it, he said he could not sell datagrams to the rest of the OSI community - but thought AFTER he got the X.25-based OSI specs agreed he might be able to get people to accept a connectionless addition. Eventually there was a CLNP (connectionless Network Protocol) but it was never widely implemented - nor was most of OSI except for X.400 I suppose.

Remi's work had the VCs built into the network while we and Louis preferred a datagram interface and service. Oddly enough, the ARPANET had a datagram interface but an internal virtual circuit!

Your comments about best efforts and the use of gateways to link networks with very different characteristics are spot on - we lost that in many respects when Cisco started routing IP directly and not encapsulating in anything other than PPP or the moral equivalent or over ethernet. I really liked the indirectness of encapsulating IP in a network layer.

David Reed

Vint - you are closer to Pouzin than I ever was, so I could very well be wrong. I don't remember him ever advocating pure best-efforts datagrams without circuits at the top layer, all I remember was that he pointed out that circuit functions were better done on top of datagrams. But his normal audience was the people who used circuit abstractions (comms people), so that might have shaped the context so he would never have talked about pure datagrams as a useful capability to have. I should probably go back and read some of his stuff again, because I haven't read that literature for 25 years, and it has a way of blurring together into the OSI world, etc.

Larry Roberts has always resisted the idea of moving functions out of the network, since I've known him. I remember arguing with the Telenet people repeatedly that they should think more about internetworking and less about being a homogeneous and optimized network for terminal traffic. They just never got the "Internet" idea - they were just doing the ARPANET again. They were competing with Tymnet and SNA, and seemed to think that they should move towards them.

Vint Cerf

David,

You may be correct that Louis always included VC on top of DG - he kept the VC at the edge however and not in the net. I am not sure he had applications that were purely datagram in nature -I have some old slides of his in my files from the 1970s and if I ever get to it I will unearth them (I am in Hong Kong at the moment and in the middle of three weeks of travel).

This is a great interchange so I'm archiving it - hope that's ok with all of you
Vint