The Clamor Outside as INWG Debated: Economic War Comes to Networking

John Day
Boston University

The debate inside the International Network Working Group (INWG) and the events between 1974 and 1976 that led up to the INWG 96 transport protocol were a small part of a much larger debate going on within the communications and networking industry. This article places the INWG discussions in this wider context to better understand the technical points and implications and their ultimate impact on established business models.

In a 2011 Anecdote department article in the *Annals,* Alex McKenzie provided an excellent account of the events between 1974 and 1976 leading up to INWG 96, a proposed internetwork transport protocol, as well as sound reasoning about the impact of that effort.[1] As an anecdote, McKenzie's article necessarily focused exclusively on the events in INWG (International Network Working Group), which was later designated the International Federation for Information Processing (IFIP) Working Group 6.1 (WG6.1). INWG was formed primarily by network researchers from Cyclades, NPL (National Physical Laboratory), Arpanet, ETH, University of Liege, and other groups in the wake of the First International Conference on Computer Communication (ICCC) in 1972 to provide a focus for developing protocols and tools for a new networking paradigm.[2] INWG 96 was the key element for this new paradigm. But adopting the paradigm threatened the business models of the phone companies and IBM. This article tries to place the INWG discussions in this wider context to better understand their technical points, importance, and impact.[3]

The New Paradigm for Networking

By the 1970s, IBM dominated the computer industry with 85 percent of the computer market. Unlike today, the other 15 percent was occupied by many other companies (RCA, General Electric, Amdahl, Xerox, Burroughs, NCR, and Univac) as well as the newcomers in the minicomputer market (DEC, Data General, Prime, and so forth). There was considerable variety in the hardware architectures and operating systems of these products.[4]

Similarly, the communication industry was dominated by big players, but in this case, by monopolies. In the United States, AT&T, Bell System, and Western Electric were regulated by the Federal Communications Commission (FCC), and they dominated 85 percent of the market. The remaining 15 percent of the market consisted of close to 6,000 independent phone companies (mostly cooperatives) of which only a half-dozen, such as GTE, were of any size. In most of the rest of the world, the telephone company was a government ministry referred to as the PTT (Post, Telegraphy, and Telephony).[5] The PTTs were the equivalent of AT&T and the FCC in one entity, making the rules they had to follow. Both IBM and the major PTTs, being essentially vertically integrated—that is, building the products and providing the services—wielded immense market and political power (not just influence).

Computer communications began almost as soon as there were computers. For example, in 1951 the University of Illinois had dedicated time every night via phone line to Ordvac in Virginia until they could complete building their copy of Ordvac, Illiac I. This early use of voice networks for data (by no means the first) was either terminal to mainframe or it was mainframe to mainframe, for example, using Autodin I message switching.

 Published by the IEEE Computer Society 1058-6180/16/$33.00

In the 1960s, government researchers were exploring the possibilities of networks dedicated to moving data. The story has been told many times how Paul Baran at the Rand Corporation and Donald Davies at NPL in the UK independently invented the idea of packet switching. Rather than sending a continuous data stream, the way voice was sent, data would be broken up into small packets for transfer along network paths.[6]

Oddly enough, whether this was revolutionary depends on your age at the time. If your formative years had been with telephony, then this was indeed revolutionary. Moving from a continuous stream to discrete packets is a huge change. However, if you were just a few years younger and your formative years were with computing and you know little of telephony, then packet switching was "obvious"! Consider the problem from that perspective: Data in computers is stored in finite-length buffers (in those days, relatively small buffers), and the problem is to transfer data from one computer to another. What should you do? Pick up the buffer and send it! What else would you do? And if you were sending a buffer, you wouldn't do it without taking precautions that it wasn't lost or damaged and that it wasn't being sent too fast. Packet switching is a logical solution to that problem, which explains why it was invented independently and in somewhat different forms more than once.[7]

As often happens in science, these insights seldom happen all at once in one fully formed transformation. This was no exception. The packet-switching proposals as envisaged by Baran and Davies still retained many of the properties of the traditional phone system. This is to be expected. To paraphrase Ludwig Wittgenstein,[8] we have to climb the ladder to see what is next. But first we have to build something to climb up, so we can see further. These ladders are thrown away a bit more frequently as the new idea (paradigm[2]) is in its initial flush of understanding and then become less frequent as understanding moves into what Kuhn calls "normal science."[9] It is only when viewed from the future that it appears to have been a step function.

In 1969, ARPA began construction of the Arpanet, taking its lead from Baran's ideas.[10] Soon after, Louis Pouzin, who had worked on Project MAC at MIT and was now at IRIA near Paris, saw the Arpanet on a return visit to the US. Pouzin resolved to build a packet-switched network in France to study networking, in contrast to ARPA which built what was primarily a production network to facilitate research on other topics. This was an important distinction. They called the network Cyclades, named after the Cyclades Islands near Greece, which had a form of distributed government. The group determined that for a network to study the nature of networking, they should start from the basics. This led to the insight that completed the paradigm shift.

It had become clear in the late 1960s and early 1970s that computing and communications were destined to merge.

They saw the problem more as distributed computing and in particular a resource-allocation problem. Consequently, they applied their experience from operating systems (OSs). Two examples should suffice. From OSs, it was well-known that dynamic resource allocation required orders of magnitude fewer resources[11] than the more traditional static allocation of resources to connections. As a consequence, dynamic allocation didn't require all messages in a "conversation" to follow the same path. The packets could be routed independently, making dynamic use of resources, and reassembled into the original conversation at the destination. Although that also entailed some unavoidable loss from congestion, it was reasoned that with proper congestion control it would be tolerable and in any case the computers (hosts) would always check to make sure all the data was received correctly. This meant the network didn't have to be perfectly reliable. As long as the network made a "best effort" to avoid congestion, the hosts would provide end-to-end reliability far more cost-effectively. Any messages lost in transit would be recovered by the hosts. Pouzin called this the datagram model or, more commonly later, the connectionless model. They also applied the Dykstra's concept of layers[12] at first to manage the complexity of the software and in doing so discovered that layers in networks were more general. They were a locus of distributed shared state. Higher layers tended to have greater scope. Notice how this requires a shift in perspective to look at the *whole* problem rather than the traditional separation where telecom people

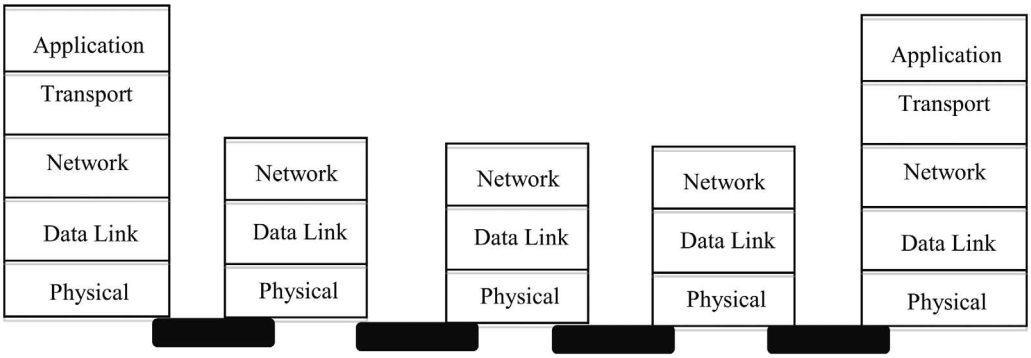

Figure 1. What became the classic five layer model proposed originally for Cyclades, where the primary purpose of the data link layer ensures corrupt messages are discarded and if necessary recovered, while the transport layer was primarily concerned with recovering loss due to congestion (whole messages) and the rare memory error in relaying. The true insight was that dynamic resource allocation in a network layer that abandoned the traditional connection would require orders of magnitude fewer resources.

considered the network aspect of the problem and computer people considered the host aspect of the problem. Pouzin was doing what science often does: establish the minimal capability with the least assumptions and then consider its properties and what is necessary to enhance them.[13] This yielded the classic five-layer model (see Figure 1) we so often see:

- a *physical layer* provides the interface to the physical media;
- a *data link layer* primarily detects errors on the physical media and hence is tailored to its characteristics and provides sufficient error and flow control over the physical media that end-to-end error and flow control can be effective;
- a *network layer* relays the packets through the network;
- the *transport layer* primarily recovers from loss due to congestion and the rare memory error during relaying, providing the end-to-end error and flow control, which ultimately creates the reliable network from unreliable lower layers; and finally
- the *application layer,* the reason for the network.

The Baran and Davies ideas were the first part of a paradigm shift (as continental drift had been), and Pouzin provided the key refinement (with plate tectonics[14]) that completed the creation of a revolutionary new paradigm. Pouzin's datagram model of using dynamic resource allocation from operating systems and building reliable networks from unreliable parts captured the imagination of the small community of researchers in the US and Europe working on networking at the time. It was the combination of Pouzin's operating system experience (different perspective) and a first working example, the Arpanet, (a good ladder) that enabled Pouzin's insights.

It had become clear in the late 1960s and early 1970s that computing and communications were destined to merge. "Convergence" was quickly becoming a watchword. Although the researchers were still working out the details of the new model, they strongly agreed about its major precepts. This was a new paradigm in the original sense of the word.[15] The new model had four primary characteristics:

- It was a peer network of communicating equals, not a hierarchical network connecting a mainframe master/server with terminal slaves/clients.
- The approach required distributed shared state of different scopes, which were treated as black boxes. This lead to the concept of layers being adopted from operating systems.
- There was a shift from largely deterministic to nondeterministic approaches, not just with datagrams in networks, but also with interrupt-driven as opposed to polled operating systems, as well as technologies such as Ethernet.
- Last but not least, this was a distributed computing model, not a telecom or data-communications model. The network was the infrastructure of a distributed computing facility.[16]

Although less fundamental, there was also a strong tendency to look for solutions that

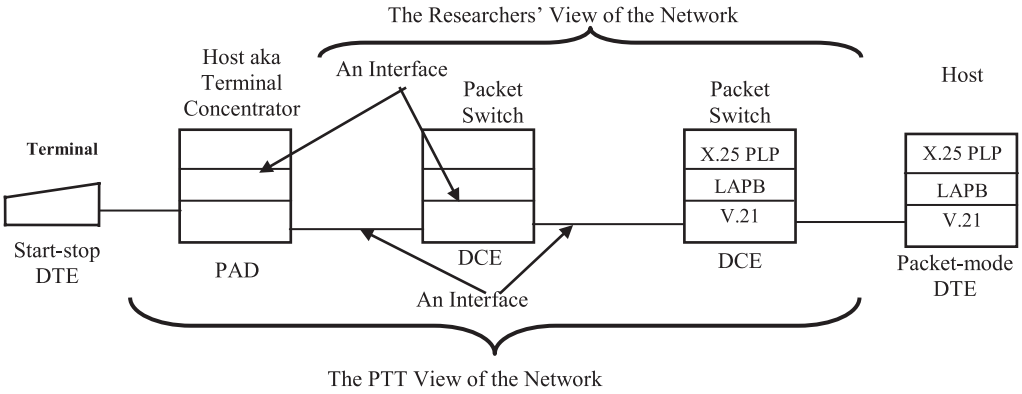

Figure 2. A simple configuration of a terminal connected to a host illustrates the difference in worldviews. The researchers' layered model (upper labels) yields a symmetric model and defines interfaces to be within boxes with no regard to who owns what, thereby market neutral. The PTTs preferred the beads-on-a-string model (lower labels), which yields an asymmetric model where interfaces are defined as between boxes and allows them to delineate which boxes belong to whom, thereby defining a market. The terminology can be read as follows: A number of terminals, *start-stop-mode data terminating equipment (DTEs)*, are connected to a host or terminal concentrator of limited capability, the *packet assembler disassembler (PAD)*, which connect to a network of packet switches, *data communicating equipment (DCEs)*, which connect to fully-functioned hosts, *packet-mode DTEs*, for example a mainframe timesharing service. If the reader finds the PTT terminology somewhat confusing and convoluted, then the author's point has been made.

yielded degenerate cases rather than special cases.[17] This was a radical change from a field that had been focused on the electrical properties of physical media to one focused on distributed resource allocation. The mathematics and the concepts between the two are completely different. An introductory networking textbook in 1975 (then called data communications) would have had 400 pages on the physical media (signals and information theory) and 50 pages on everything else. Today that proportion is exactly the opposite.

Threat to the Status Quo

To the casual reader, the characteristics of this new paradigm may seem innocent enough, almost too philosophical to have any bearing on hardnosed product development and business issues. To IBM and the PTTs, however, each characteristic represented a sharp object pointed directly at their hearts, and more importantly at their bottom lines. This was not at all what IBM and the PTTs had in mind. In fact, it invalidated their business models and was the beginning of an intense controversy that dominated networking for the next two decades and, in a real sense, continues to this day. The researchers' perfectly reasonable, but radical new networking model was counter to the traditional model of telephony and data communications of both IBM and the PTTs. They had angered not one but many 500 pound gorillas!

While the researchers in INWG were filled with enthusiasm about the possibilities of the new model, the PTTs did not disguise their disdain for the new ideas. Who were these young, mostly unkempt ivory tower types to be telling them how to build networks?! According to them, the researchers needed to go back to their labs and leave building real networks to the professionals. The PTTs preferred their traditional "beads-on-a-string" model.[18] Their model served nicely as a technical guide and conveniently to define markets by defining boundaries between boxes that determined which belong to the PTTs and which didn't (most of them did).

Figure 2 provides a revealing example of how the PTTs used this model with their first packet-switched network standard, X.25, to both guide the technology and their definition of the market. X.25 defines the interface between a host and the network. It does not define how the network works internally.

Terminals, or teletypes, accepted characters delimited by start and stop bits, hence the term "start-stop-mode DTE." DTE is anything that belongs to a customer. The PTTs defined the network's boundary such that it included the boxes they wanted to own (PADs and DCEs).[19] Even the name "packet

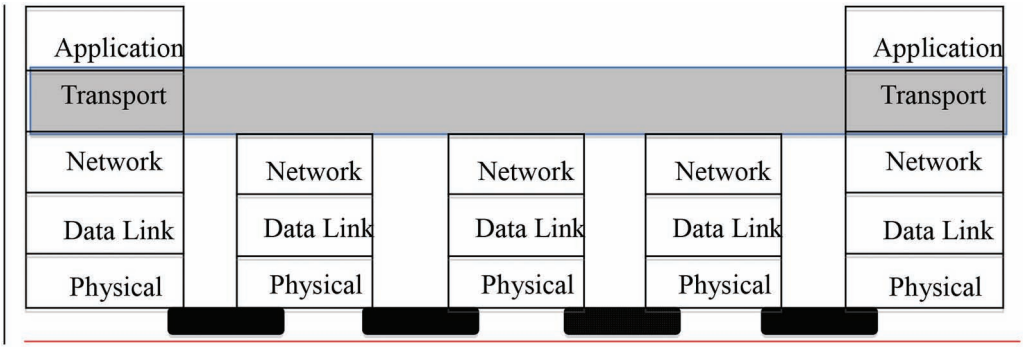

Figure 3. Datagram model. The transport layer of the new model sealed off the PTTs to a commodity business, making it impossible for value-added services to be in the network and remain the exclusive domain of the PTTs.

assembler/disassembler" was chosen to make it sound more like network equipment rather than, say, the "terminal concentrator" or "terminal front-end" terms used for the same equipment on the market at the time.[20] This was done even though this yields an asymmetric model that was a point product and a development dead end.

The research community spoke about the same boxes, but a PAD was viewed as a small host. Terminals were a local matter outside the model. Why? Theirs was a distributed computing model of peers. Terminals at the time didn't have the capability to be even a limited computing peer.[21] Thus, yielding a symmetrical model, which is open ended for new developments that will generate new revenue and hold the customer. The asymmetric model is a point solution that cannot be extended without significant change and likely requiring new equipment. This opened the door for competitors, which was not a concern for government monopolies.

The PTTs also had plans for various value-added services that would require databases and hosts to provide the service. The French Minitel's use of videotex is a prime example of what they had in mind.[22] But it was important to the PTTs that these high-margin, value-added services belonged exclusively to them and not be subject to competition. To do this, they had to be seen as being "in the network."[23] The beads-on-a-string model let them define what was outside the network and didn't restrict anything they wanted to put inside the network. On the other hand, the model put all hosts and, by implication, application services outside the network at its edge undermined their claim of exclusivity.

Although claims of exclusivity have waned with deregulation, there has been a constant drumbeat by the larger players emphasizing centralization for an inherently decentralized problem. The PTTs and their proponents talked of (or were accused of) creating walled gardens to lock in customers. Although the names change with time, the story has remained the same: timesharing with mainframes for the 70s, client-server systems for the 80s, network computers for the 90s; and cloud computing for the 00s. The centralization inherent in all of these has the same goal: to concentrate power and ensure product lock-in among a few (they hope, one) and to thwart the inherently distributed nature of the network.

The PTTs recognized that the datagram model relegated them to a commodity business of just moving bits as cheaply as possible (see Figure 3). In their world, which charged for packets delivered and connect time used, how could they charge for datagrams that were not always delivered and had no connections? Furthermore, datagrams made the network nondeterministic, not the nice neat deterministic world of telephony circuit switching they were familiar with. For the PTT engineers, software was disconcerting enough, but the nondeterminism of the datagram model was even more disconcerting.

The debate over connection versus connectionless lead to the other major technical disagreement of the period: hop-by-hop versus end-to-end error control. The Arpanet had used a hop-by-hop approach, which was taken up by the PTTs in X.25. The hop-by-hop model tried to correct errors as the packets were forwarded and ensure that none were lost. Each switch would hold a copy of the packets it sent until the packet was acknowledged by its neighbor, leading to static resource allocation. However, there were always error cases of

switches or lines failing at inopportune times that led to lost packets. As noted earlier, the new datagram model took the end-to-end approach to error control proposed by Pouzin. In this approach, the network does not attempt to be perfect, but only makes a best effort—that is, enough effort that end-to-end error control in the hosts is cost-effective. Many papers were written defending one side or the other. The PTTs argued that their networks were reliable, although they clearly weren't. There were several common error conditions that caused a reset in X.25, which lost data.

It is important to pause here to remember that this argument was not taking place within INWG. For the INWG group the hop-by-hop design, beads-on-a-string model, and value-added services in a network were an anathema: concepts of the old guard. The discussions within INWG were to create a united position to counter the PTT position.

The hop-by-hop design had several advantages for the PTTs. With no transport layer, they could continue using the beads-on-a-string model and (exclusively) offer value-added services. For their engineers, this seemed close to their traditional concepts of circuit switching (even though considerably more complicated), an approach they were comfortable with. And of course being engineers, the idea of building network switches that were nondeterministic and not absolutely bulletproof went against everything they had ever done. This debate over hop-by-hop versus end-to-end error control consumed much of the 1970s and 1980s.

In the datagram model, the high margin value-added services were in the hosts. Hosts were always "outside the network." There was no inherent distinction between hosts belonging to the customer or to the provider. It was not possible to draw boundaries to carve out exclusive markets as the PTTs had always done.[24] How could they offer (lucrative) value-added services exclusively "in the network" if the only function in the network was moving bits as cheaply as possible?[25]

Getting rid of the transport layer would allow them to keep the beads-on-a-string model and continue to use the architecture to protect their markets. The PTTs were very much against a transport layer over datagrams. Datagrams were less a worry. They built their networks. They could do as they pleased and simply not use them. But transport protocols were in customer hosts, outside their domain of control. They were a major threat to the traditional business models. However, contrary to their spin, X.25 was not reliable. Not infrequent conditions could cause a reset accompanied by the loss of data.

The PTT's solution to recovering data lost by X.25 resets was a classic of standards sleight of hand. They defined RTSE (Reliable Transfer Service Element, X.228) as an application element and argued that it was for application-layer recovery of lost data—for example, a checkpoint recovery typically used after a system crash to restart a long file transfer without beginning at the start of the file—not network-layer recovery of lost data. However, a cursory inspection of the protocol noting the time intervals it operated on were closer to packet round-trip times of a transport protocol than to the much longer intervals of file transfer check-pointing. In addition, the fact that it was always required, even when not used for transferring large amounts of data (where checkpoints would be used), indicated otherwise: It was actually a transport protocol in the application layer. By putting it in the application, the PTTs retained the beads-on-a-string model and avoided (in their minds) relegating the network to the commodity business of just moving bits. By contrast, the networking groups saw the transport layer as part of the operating system.

The new model was also a threat to IBM. Its networking product, SNA (System Network Architecture), was a centralized hierarchical architecture rooted on the mainframe and polling simple terminals, quite deterministically.[26] Even then it was recognized that, although a peer architecture could be subset to be hierarchical, it was not possible to go the other way—in other words, to "enhance" a hierarchical model into a peer model. In a hierarchical model, too many decisions are made in too many places based on the assumption that either there is a single mainframe as the master or the network is a tree. If this was to be the future, there was no way to transition SNA to this peer model. As indicated by IBM's design choices (SNA polled terminals) and that SDLC (Synchnronous Data Link Control) and their 1980s token ring support were all deterministic, IBM's engineering culture was also uncomfortable with this shift away from determinism.

However, IBM could not be as heavy handed in opposing the new model as the PTTs. The PTTs were a legal monopoly, and IBM wasn't. IBM had to be wary of doing anything that could be construed to be antitrust or anticompetitive.[27] There had been antitrust

This was a war of competition disguised as a rational technical argument among experts.

suits in the past,[28] some ongoing, and it had gotten its hands slapped for announcing products and then canceling them as well as other practices (some innocent, some not so).[29] Although the new approach was no less threatening to IBM, it had to be more subtle in its opposition. IBM needed time. Had IBM realized how its view would continue to dominate networking for the next three decades, it might have been less concerned.

The old guard held fast to a deterministic worldview. Meanwhile, the "young turks"[30] championed a worldview characterized by decentralization and nondeterminism that drew on computing and operating systems concepts that were just as unsettling to the old guard.[31] This new view was not only a threat to their business models but to their intellectual models. It would suit them fine if it just went away, and if not, they would do what they could to help it go away.

However, the other computer companies, especially minicomputer companies, saw that while this new model created problems for the big incumbents, it played to their strengths. (A comparison between dinosaurs and mammals was not uncommon at the time.) Minicomputers were being used for early packet switches and relied much more on interrupt-driven operating systems. Their engineers tended to be younger and among the first to be educated entirely in computing. Minicomputer engineers were much more comfortable with the new model. These companies strongly supported this new approach, not because they believed it represented "truth and beauty" but for hardnosed business reasons. Minicomputer companies understood that costs were going to fall for a long time (this was later referred to as Moore's law[32]) and falling prices were to their advantage. While the incumbents struggled against the corporate inertia of an installed base, the minicomputer companies could take advantage of the move to "convergence" and leverage this new network opportunity to acquire market share.

This was a war of competition disguised as a rational technical argument among experts. This war of ideas had real consequences and was raging outside INWG. While most INWG members were aware of the pressure, it was much more real to the Europeans than the Americans. Its effects on the discussions in INWG were significant.[33] If nothing else, knowing there is a faction wanting to eliminate the idea entirely inhibited the discussion of the warts or apparent weaknesses that often occur in the early stages any new idea. There is reticence to explore and understand these warts if there are those who would seize on them not to improve the idea, but to abandon it. In fact, there is a tendency to avoid the topics entirely or to argue that the weaknesses are not weaknesses. All of this gave rise to a bunker mentality. This was doubly harmful because the problems with a new idea often lead to a deeper understanding and a much improved result. The debate with the PTTs was intense and heated, causing both sides to dig in behind their positions even more. And it only intensified over the next decade.[34]

Electro-political Engineering: The Battle Lines

In the 1970s, most communication standards were developed in the CCITT (Consultative Committee for International Telegraphy and Telephony[35]), a branch of the ITU (International Telecommunications Union). In most of the world, the PTTs were government ministries. Consequently, agreements at ITU were agreements among governments, so ITU is a United Nations treaty organization. Thus, the member states of the UN are the only ones who can vote.[36] At the time, the CCITT published a set of new recommendations on a four-year cycle. All new recommendations were issued at once in the famous "colored books," each cycle having a different color.

At the same time, the prospect of computer communications offerings such as SNA had created a market for private or corporate networks. This opened the door for ISO (International Organization for Standards) to initiate projects on communications. ISO is a voluntary standards organization and national representatives are not necessarily part of the government.[37]

CCITT's Study Group VII (SGVII) was developing the connection-oriented protocol for packet-switched networks, X.25, for inclusion in the 1976 recommendations.[38] X.25

defined an interface between packet-mode DTEs and DCEs—in other words, between hosts and switches. It defined three layers: a *physical layer,* using V.21; a *data link layer* using LAPB, a variant of the HDLC (High-level Data Link Control) protocol; and a *network layer,* using the X.25 packet-level protocol. There was no transport layer protocol being developed in SGVII for two reasons: they didn't believe one was necessary, and a transport protocol was a host-to-host protocol and by definition outside their domain (another reason for not liking it).

One of the first efforts by INWG was to introduce datagrams into X.25, an alternative Layer 3 protocol. INWG realized that this was a long shot. It was late in the cycle for even a member to introduce new work to be included. There was even less chance that radically different new work from a mere liaison organization, such as IFIP WG6.1, would be accepted.

The PTTs countered with a proposal that they called Fast-Select. Fast-Select was supposed to set up, send data, and close the connection with a single packet exchange. This was not at all a connectionless datagram, and considerable time and effort was expended explaining why it was different and why Fast-Select was not acceptable. The INWG proposal was unceremoniously rejected. Ultimately, a datagram option was included in the first X.25 recommendation in 1976, but it was an empty victory given that the probability that any PTT would implement it was close to nil. Probably the most positive accomplishment of this effort was to convince SGVII to align LAPB[39] with the ISO HDLC standard from which it was derived. The differences were minor and almost arbitrary, so it was a relatively easy argument to make.[40]

Commercial packet-switched offerings began to appear as early as 1974. X.25 networks became operational in the US, Britain, Canada, France, and elsewhere as soon as X.25 was published in 1976. INWG and the computer companies aligned with it had learned that being a liaison delegate to CCITT had little influence. Not having a vote in CCITT severely limited what they could hope to accomplish. They were going to need a playing field where they had more influence. They immediately began to approach the ISO data-communication committee, TC97/SC6.[41]

INWG recognized that the real battle would be over internetworking. Basically, two approaches to internetworking were being investigated: protocol translation at the boundary between two networks and adding an internetworking layer over the different network technologies to provide a common service to the layer above. In the latter case, each network layer provided a service to an overarching Internet transport protocol, just as each data-link layer provided a service to an overarching network layer.

The INWG participants had already seen a much wider range of network types with the Arpanet, the PTT X.25 networks under development, Ethernet, SATnet, and the early Packet Radio Network, and it was not hard to imagine others in the near future. Believing (correctly) that with such a wide variety of networks that translation would become burdensome and complex, they chose the second approach.[6]

SGVII, on the other hand, was looking at a much narrower problem. For them, it was only a question of how to interconnect X.25 networks. They chose the protocol translation strategy. X.75 defined a hop-by-hop connection-oriented interface (in their terms) between two X.25 networks—a distinctly beads-on-a-string approach. A X.75 draft first appeared officially in the 1978 documents published at the half-way point in the four-year cycle. (Contributions and earlier drafts had been circulating well-before that.[42]) The SGVII approach was not going to work for the more general problem that one could already see coming in private networks.

Hence, INWG needed to have a well-developed proposal for an internetwork transport protocol. It was in this context that INWG started work on what would become INWG 96.

The Debate within INWG

The crux of all protocol designs is the coordination (or synchronization) of a distributed shared state. To transfer data reliably, a protocol is implemented as two finite-state machines exchanging messages. The messages carry not only the data, but also information to update the state machines—that is, control information. The control information is used to ensure that the data received is not lost or corrupted and is in the correct order and that the sender is not sending too fast for the receiver. Because data is constantly being transferred, the state of the two machines is seldom the same, but they should be consistent so if transfers stop they will settle to a consistent state. Designing protocols to operate end-to-end presented new problems, especially on the issue of how to achieve the necessary synchronization for

The Road to Understanding Distributed Synchronization

On first encounter, Richard Watson's proof that the necessary and sufficient conditions to achieve distributed synchronization is to bound three timers seems to come out of left field, a bolt from the blue; it's an astounding result with no precedent. Messages are exchanged, but messages are always exchanged. That isn't what ensures that synchronization is achieved. As with most "discoveries," this is not really the case. They are more a case of a developing deeper understanding than sudden enlightenment, although I would hazard to guess that there was an instance when Watson *saw* the answer before he had worked out the complete proof. That insight came from the increasing understanding of the problem that that was accruing over time in the early 1970s. This process has always moved by erratic steps.[1] Given that this problem is at the heart of all data-transfer protocols, it is important to take a closer look at how this knowledge developed.

The early data-link protocols (those that operated over a point-to-point link, such as SDLC and HDLC) provided reliable communication. They worked well. If a message was lost or damaged, withholding the ACK caused a time-out and a retransmission of the unacknowledged data to correct the problem. Propagation time was fixed and deterministic. There was not much room for variability. The link-layer protocol design was straightforward. The new transport protocols, such as those represented by INWG 39, 61, and 96, had new failure modes: messages were being relayed at a potentially variable number of routers, which could incur a variable amount of delay and messages could be dropped or could cause a loop because of inconsistencies in the distributed routing. These problems could occur any time including during initial synchronization and termination, sometimes called connection establishment and release or disconnect. This was a much less straightforward problem than in traditional data-communication networks of point-to-point links.

Initial synchronization for a transport protocol was more complicated. From a software design perspective, there was a desire to avoid timers. Timers seemed ugly and an admission that the design was falling back on a mechanism external to the protocol itself. They seemed like a cop out, a last resort rather than solving the problem. It would be more elegant if one could find a sequence of messages that would achieve synchronization without having to resort to timers. What sequence of messages alone could achieve synchronization? They had not been required with data-link protocols, so why did they keep turning up in transport protocols?

The first investigation into the problem came with Dag Belnes's paper.[2] Belnes correctly assumed that if the worse-case solution was understood, the problem would be understood: How many message exchanges does it take to deliver a single message reliably? Belnes started with an exchange of two and detailed what could go wrong. After testing three-way and four-way exchanges, he found that a five-way exchange is mostly reliable as long as neither side crashes. To protect against a crash, it looked like a timer was needed. This was getting pretty messy: five messages exchanged to deliver just one.

Carl Sunshine worked out the details for TCP (INWG 39) for what is now known as the three-way handshake for initial synchronization.[3] Of course, when taken in the context of a TCP connection with SYNs at the beginning, two FINs at the end, and n messages sent in between, Belnes' result is simply the degenerate case. When there is a sequence of messages, the data and ACKs form a continuing three-way handshake followed by the two FINs on the end to close off the connection, a bit like a chain stitch. So there is always an ongoing five-way exchange: If $n = 1$, it is Belnes' five-way exchange.

Phil Merlin wrote his thesis on formal description techniques, proving the properties of distributed synchronization with Petri nets.[4] Merlin proved that timers were always required for synchronization. (There were

the feedback mechanisms of the protocols to work reliably. Without distributed synchronization, the protocol can fail to make progress or, worse, deadlock.

The protocols that had been developed for data communications up to this point had operated over point-to-point links (opposite ends of a single wire). This simplified the problem or, perhaps more correctly, avoided some of the more subtle cases that could occur when operating over a network.[43] Solving this was a matter of considerable research in the 1970s (see the sidebar). There are other issues to be settled when designing a protocol (such as how large sequence numbers should be and whether to use fixed or variable window flow control), but this is by far the crux of the matter. The other issues are distinctly secondary.

For the people in INWG, there was general agreement on the big issue of synchronization. As McKenzie points out, the differences between the two INWG proposals were not

also papers on this problem by Andre Danthine and his students at the University of Liege and a paper by Hubert Zimmermann at IRIA, and Michel Elie at Bull, but I can't find my copies to cite them.)

Richard Watson was looking at this problem at about the same time. The fact that, even with the three-way handshake, hazards (or races) could still occur nagged at Watson. There had to be a hazard-free solution. There was something we weren't seeing. The more he thought about the problem, the more it kept coming back to the problem of time: the time to know that messages had either been delivered or disappeared from the network. In that case, he had to carefully determine what contributed to how long messages could be in the network.[5] Watson determined that there were three: maximum packet lifetime, the maximum time to wait before acknowledging, and the maximum time to exhaust retries. With that it was clear that as long as the upper bound on these was known and when a new cache of state began, synchronization for reliable data transfer was assured. A special initializing and termination sequence of messages was unnecessary.

Why hadn't someone determined this earlier? Aren't the bounds on these timers necessary in data-link protocols? They are. But the deterministic nature of a point-to-point link (propagation time) masks their importance. Data-link protocols with feedback were a degenerate case, not the general case. The greater variance with relaying experienced by transport and the uncertainty of a message arriving had exposed potential hazards. The existence of the hazard convinced Watson that a simpler explanation was there and, as it turned out, so was a more general property, a deeper understanding. Then it was necessary to understand the implications of the results for protocol design.

It is interesting to also note that, while all of these results were reached in the context of the problem of achieving distributed synchronization for reliably transferring data, the heart of the problem is the synchronization required for the feedback mechanisms of retransmission control (ACK) and flow control.

Another important thing to note is that the publication dates for the papers I have cited here are fairly meaningless. There was a very small group doing network research, and for the most part, they all knew each other. These researchers were exchanging drafts and documents among themselves long before publication. Publication dates then have to be viewed more as the upper bound on when everyone could know the results, rather than the lower bound.

Of course, these exchanges were not perfect, so we can't be sure that everyone was seeing everything from everyone else about the same time. Still, given the labor required to produce a journal article then and the delays in publication, the journal publication dates and even thesis dates were often long after the results were generally known by those working in the field. Even so, it is clear that this work all took place in a short period of time. For some people, the results that influenced their understanding depended on the order in which they saw them, rather than the order the results were available in published journals. One cannot assume synchronization of knowledge.

References and Notes

1. For a classic example, see I. Lakatos, *Proofs and Refutations*, Cambridge Univ. Press, 1976.
2. D. Belnes, "Single Message Communication," *IEEE Trans. Comm.*, vol. 24, no. 2, 1976, pp. 190–194. This paper was circulating earlier.
3. C. Sunshine, "Connection Management in transport Protocols," *Computer Networks*, vol. 2, no. 6, 1978, pp. 454–473. Sunshine's thesis was circulating earlier.
4. P. Merlin, "A Study of the Recoverability of Computing Systems," PhD dissertation, Univ. of California, Irvine, 1974.
5. R. Watson, personal comm., June 2016.

substantial, and they boiled down two narrow issues: how fragmentation was handled in terms of packets or bytes and whether the communication was a stream or what was called a "letter." A bit of explanation will indicate why these were not that important.[44]

One concern in connecting networks together was that networks had different (and often small) maximum packet sizes (250 bytes was not uncommon). This would make it necessary to fragment packets in transit across different networks and then reassemble them. The different packet sizes arose primarily because of physical limitations in both bandwidth and computing. INWG 39 (the TCP-based proposal) and INWG 61 (the Cyclades TS-based proposal) had different approaches to fragmentation. Any fragmentation scheme depends on how the packets are identified, usually by a sequence number, so that the fragments can be reassembled into the original packet. There were three

primary concerns relevant to the sequence number and fragmentation:

- minimizing space in the message header required the support of fragmentation (especially in 1975),
- the processing overhead associated with fragmentation and reassembly, and
- how long it would take for the sequence number space to roll over.

The time must be long enough that all packets and their acknowledgments have disappeared from the network before a sequence number can be reused. Neither INWG 39 nor INWG 61 was completely satisfactory.

INWG 39 used sequence numbers that were the number of the first byte in the packet. (With byte sequencing, if the first packet has sequence number X and carries Y bytes of data, the sequence number of the next packet will be $X + Y$). Fragmentation was a simple matter of counting the bytes in a fragment and giving the next fragment the appropriate sequence number. This was an elegant and simple solution to the fragmentation problem, but it had two disadvantages: First, it consumed sequence number space quickly, and with even a moderately high bandwidth-delay product, it would be necessary to stop sending before an ACK was received. (It would not be possible to keep the "pipe" full.)[45] Second, because there was no requirement for retransmissions on the same byte boundaries, reassembly could be more difficult.[46]

INWG 61, on the other hand, sequenced packets. Packets were numbered sequentially, instead of using bytes. In this case, fragments were identified by an extension to the sequence number that enumerated the fragments in multiples of a given minimum fragment size. What was a reasonable minimum fragment size? The smaller the minimum, the larger the field required to enumerate the fragments. If very small sizes were temporary and would fall by the way as new technologies developed, the field would be too long and there would be unnecessary overhead. If the field was always present and the packet wasn't fragmented, the field was unnecessary overhead. If the field was optional, then it required significantly more complex processing.

Clearly, the packet sequence numbers of INWG 61 would take longer to roll over than byte sequence numbers. For a 32-bit sequence number, the INWG 61 sequence number would roll over after 2^{32} packets were sent, whereas the INWG 39 sequence number rolled over after 2^{32} *bytes* were sent. Even at this early date, the prospect of satellite channels with large bandwidth-delay products raised the concern that the 32-bit byte sequence numbers in INWG 39 would be too small. A bigger sequence number meant not only more packet overhead, but also required "long" arithmetic that was not always available and would require simulating it in the critical path. Also the ACK field would have to be the same size as the sequence number. Making one longer makes the other longer as well. This was especially an issue for INWG 39, which had a fixed-length single header format that was already larger than comparable protocols. Doubling the sequence number from 4 to 8 bytes would have increased the header from roughly 28 to 36 bytes.[47] At a time when a 250-byte maximum packet length was not uncommon, this was a real concern.

The debate over the letter concept was even less critical. Here the issue was about how the data stream was structured. INWG 39 proposed a byte stream. The application would transfer some amount of data to the transport layer, which might be sent as one or more messages. The receiver would deliver whatever data it had when the application at the destination did a "read." If the application needed to know where certain boundaries were in the data, such as between application messages, the application had to delimit them.

INWG 61 proposed letter integrity that was maintained from sender to receiver. In this case, if the application passed an amount of data (the letter) to the protocol, the protocol might fragment the letter into multiple packets or combine it with others into a single packet, but the letter was the unit delivered at the destination, meaning its identity would be preserved. In other words, the protocol can slice and dice the letter in any way it wants, but when all is said and done, it should clean up its mess and deliver the same thing it was given by the sender. This is just good software engineering.

The two views are essentially different perspectives of the same problem: the stream structure of INWG 39 is a protocol implementor's preferred view, whereas the letter structure of INWG 61 is the protocol user's preferred view. Later, the letter concept became the OSI concept of SDU (Service-Data-Unit) defined for all layers.

Both of these issues were classic engineering trade-offs, which engineers love to argue about! And as McKenzie indicated, neither

of these were that important and certainly not fundamental to the architectural direction. It was much more important that the network researchers resolve their modest differences and present a common front to the much more powerful PTT and IBM opposition, clamoring outside. When no agreement could be reached to choose one or the other of the proposals, a synthesis was proposed: INWG 96.

Slow Down, You Move Too Fast

The two proposals before INWG necessarily represented the understanding at the time. In 1975, all these concepts were new. The entire field was only about six years old, and the five-layer model was only three or four years old and had not been thoroughly explored. It was generally believed that there was more to consider, especially above transport, where it was felt that the surface had just been scratched. There were only a few examples of these kinds of transport protocols, probably less than a half a dozen, and there had been little time for major research to understand the fundamentals of networks and protocols. There was still much to learn. (It should be noted just how few researchers were working on these problems. The entire networking community between Europe and North America in 1976 was maybe a few hundred, and probably well under 100 were concerned with transport protocol issues.)

Two observations about science are important here: One, there does not seem to be any way to rush new insights. Putting more people on a problem seldom yields results any faster. Indeed, in some cases it can actually slow progress. It simply takes time for ideas to incubate.[48] Second, the longer a problem is studied, the simpler it tends to get. The variety of solutions becomes smaller; complete generality is found to be unnecessary, as understanding is clarified.[49] Some aspects are found to be non-cases, better solutions are found, conventions become dominant,[50] or some problems turn out to be different than first thought. These observations are especially true when exploring a new paradigm and sorting out what carries over from the old paradigm, in what form, and what no longer matters.[51]

The INWG researchers were definitely in this problem-solving mode. Even a cursory browse through INWG 96 shows that things had not yet settled down.[52] While much is right, in some cases more right than current practice, there is much more in INWG 96 than one would expect today.

However, the work was being moved from research to standard far too quickly. Neither INWG 39 nor INWG 61, nor the synthesis in INWG 96, were the complete answer. There were still fundamental insights to be made. More important than the issues debated in INWG were the key insights provided by Richard Watson's seminal work done in 1978[53] and demonstrated in the protocol Delta-t.

Watson proves that the necessary and sufficient condition for synchronization in all data-transfer protocols—synchronization for the feedback mechanisms for retransmission and flow control—is to set an upper bound on three timers: maximum packet lifetime (MPL), maximum time to acknowledge (A), and maximum time for retries (R). In other words, there is no need for a special initial exchange of packets to create the synchronization.[54] This is an astonishing result that ranks with Baran and Davies' invention of packet switching and Pouzin's insight about datagrams as the seminal discoveries in networking, perhaps even more important, given that its implications are far deeper than simply defining the conditions for synchronization.

The perspective Watson takes to get to this result is as intriguing and significant as the result itself. And Watson has such a delightful way of putting it! "All connections exist all the time. Caches of state [also called state vectors or transmission control blocks] are maintained only on connections that have exchanged data recently!" In this case, "recently" is $2(\mathrm{MPL}+\mathrm{A}+\mathrm{R})$, or $2\Delta t$. If there is no traffic for $2\Delta t$, the state maintained for synchronization is discarded. If there is more traffic later, the state is simply recreated. The ports (the local handles used by the application and the protocol to refer to the communication) for the connection are still allocated. This is saying that port allocation and synchronization are decoupled and distinct. They are independent. All three of the protocols considered by INWG conflate port allocation and synchronization.[55] This has several major implications:

- All properly designed data-transfer protocols are soft state. (Yes, some data-transfer protocols today are not purely soft state. This is software. It is always possible to do it badly and have it work well enough that no one notices.)
- This results in a much simpler and more robust protocol.[56]
- It yields a protocol less vulnerable to attack.[57]
- It avoids the necessity of heavyweight solutions like IPsec.

- Last but not least, it defines the bounds of what is networking versus remote storage: If MPL cannot be bounded, then it is remote storage. That is, it's a file transfer, not networking. It is networking only if MPL can be bounded.

Incorporating these results into a transport protocol yields a much cleaner fit in the architecture.

Remember synchronization was not a matter of much discussion in INWG. Yet it was the major piece of the puzzle that was missing, a piece with significant implications. Should they have been looking at it? It is doubtful it would have happened any sooner. It simply had to wait for that flash of insight that can't be predicted and can't be rushed. It is unlikely (and seldom the case) that more people working on the problem would have seen it any sooner.[58] It just needed time to incubate and people with the time or inclination to think about it. Watson wasn't pursuing this problem because the other protocols didn't work well enough. He was trying to understand the fundamental properties of synchronization, not just find something that seemed to work (see the sidebar).[59] Once he had the insight and the results, he developed a protocol based on the results to test his results. Because Watson's work came two years after the INWG 96 vote, a decision to choose any of the three INWG proposals was in a real sense too early. INWG wasn't even debating different approaches to synchronization.

The pressure from the PTTs pushed the networking researchers to get a transport protocol in place soon, any transport protocol. And at that point, there was more than a little personal commitment behind each of the proposals. The question was more whether or not there would be a protocol proposed; at that point, which one was secondary. That pressure did not give anyone in the INWG group the luxury to contemplate deeper insights. A transport protocol proposal was necessary to counter the PTT directions. As is common, the perception was that this technology was much closer to widespread deployment than it was. There was a strong belief that there was one chance to get it right. When, in fact, it would be another decade or more before that happened.

This is not to say they weren't warned. In a 1974 INWG paper and later publication,[60] Frank Kuo counseled caution: "It is the author's view that it is important to prevent the issuance of premature standards by these groups [CCITT, ISO, and INWG]. Packet communication networks are still in an evolutionary stage. Early Standards could only freeze the development process." This is incredibly prescient statement, given that they ended up doing just that.

What Happened Next?

As noted earlier, by 1975, the computer industry had learned that they would have no influence in CCITT. They moved to ISO,[61] where they had a freer hand, and after considerable resistance from ISO's own old guard, the work on OSI (Open Systems Interconnection) was finally begun in March 1978, as TC97/SC16.[62] INWG 96 was a liaison contribution to the early work on the transport layer and formed the basis for what became Class 4 Transport. At roughly the same time, CCITT SGVII began its own reference model effort. With the seemingly untenable prospect of competing models and no hint of telecom liberalization on the horizon, least of all in Europe, SC16 made the probably necessary and at the same time fatal error of negotiating a joint project with CCITT (finally formalized in 1983).

The agendas of the two organizations could not have been more different. The PTTs had no intention of relinquishing any of their domain to anyone. However, the war would be more indirect, not about markets but over the supposedly technical issue of connection/connectionless. The battleground was first the OSI Reference Model (how to eliminate it or contain connectionless as much as possible or make it unattractive), then the transport layer (was one necessary?), and finally the network layer (how would connectionless standards be isolated?).

The Reference Model and the transport layer (being computer to computer) fell under SC16, where computer companies held more sway as opposed to SC6, the traditional data-communications committee, which had more PTT influence. After a three-year battle only narrowly decided (after a promise of a US walkout), it became clear that connectionless would be in the Reference Model (October 1983), but as unattractive as possible by constraining where connection/connectionless could interwork.[63] There had been a small initial victory in that the outline for the Reference Model included a transport layer.

Consequently, the PTTs fallback strategy was to make the transport protocol standard unattractive by loading it up with five optional and distinct protocols (called TP0

through TP4)[64] of which only one, based on INWG 96, was necessary. The other four options were added due to political pressure primarily from CCITT. It is worth noting that SGVII insisted on including Class 0, which had nothing but a placeholder header and a one-to-one mapping to the network layer—in other words, no transport layer. But it did require the use of RTSE.

However, it is a credit to the OSI Transport Layer Group that, in the midst of all this controversy, they were able to incorporate new research results into INWG 96 by incorporating the most important of Watson's results, explicitly bounding all three timers and decoupling port allocation from synchronization. They did stop short of discarding state if there is no traffic for $2\Delta t$.[65] Hence, TP4 has the beneficial advantages noted earlier.

The Internet designers did include an explicit mechanism to bound MPL with the time-to-live field in IP, but it is difficult to know whether this was based on Watson's results or if it was arrived at independently. There was no attempt to bound the other two timers. TCP suggests bounding MPL and does require an implementation to wait 120 seconds (an estimate of MPL) before discarding state when a connection is closed and indicates what can happen if it isn't, but it assumes that normal operation will bound the other two timers and port allocation and synchronization are not decoupled.[66] Oddly, contrary to the stereotype of the conservative standards group, OSI was more receptive to new research results than the self-described cutting-edge DoD researchers.[67] In the end, Watson's results were almost entirely lost and are not treated in any of the vocational network textbooks used in universities today. Meanwhile, SC6 became the focal point of the connectionless/connection battle with both sides committing major manpower to it. This was the crux of the matter.

Also in 1983, IBM began to run full page ads in journals like *Scientific American,* showing the seven layers of SNA.[68] SNA had always had five layers, a testament to the "flexibility" in interpreting the OSI Reference Model. The ads said, OSI was good for data transfer, but did not have network management. What IBM was stonewalling in OSI? Network management, but that is another story. OSI definitely had a two-front war to contend with.

The constant battle to limit connectionless was a major distortion and impediment to the trial and error necessary to working out a better understanding of these new ideas. Needless to say, such inquiry fell by the wayside fairly quickly. Any problem would be turned into a call for abandoning the whole approach. In OSI, which had to accommodate both, it led to artificially increased complexity. For the Internet far from the front lines of the battle, there was no question about connectionless. The Internet had no such constraints and was free to more deeply understand the connectionless concept and find new insights about its nature. Instead, the Internet community pursued connectionless with all of the zeal of a convert, applying connectionless in ways that had never been intended,[69] seeing themselves as an embattled elite, becoming more insular and subject to group think.[70] Meanwhile, the Internet research community was not addressing the real problems confronting networking. They were in a position to leapfrog the OSI effort solving problems that were too mired in politics to be addressed in OSI, but they didn't.

The intensity of the debate generated a bunker mentality among both connectionless factions (the Internet and the OSI connectionless supporters[71] versus the PTTs). Everyone hunkered down, fearing that any proposal for compromise from the other side was some sort of ruse that would later be turned against them to eradicate their view. Positions became more entrenched. There was no way to carefully investigate the fundamental nature of connections and connectionless that might have lead to a synthesis of the kind Arpanet researchers had already pulled off with several seemingly irreconcilable concepts.

One could argue that this was not the work of a standards committee and that would be correct. It is what researchers should do. Oddly enough, during this period, more new developments were created and adopted by the standards groups than the DoD researchers. But research tended to be (and still is) directed to more engineering questions to maintain the Internet, not to consider the fundamentals and what would be needed. The technical problems appeared to be oil and water, and the intensity of the debate[72] ensured the problems would stay that way.

Conclusion

McKenzie's conjecture about what might have happened had DARPA not gone its own way is sound:

> I once thought that if the research community were united behind one End-to-End Protocol, we might have avoided most or all of the

> incredibly wasteful ISO Open Systems Interconnection activity. However, the OSI effort was probably an inevitable clash between computer manufacturers and PTTs, and of both against IBM.[1]

He is right on both counts. It was far more important for the research community to present a common front against the status quo of the PTTs than to fragment over minor differences that ultimately didn't matter. But getting acceptance in a wider venue always takes more time. That wasn't necessarily bad, as Kuo had warned. There were additional benefits to be gained by a united front. There were still fundamentals to be worked out and research to be done that could not and should not have been attempted in a standards committee. Had DARPA stayed in the game and participating in the ongoing efforts, but made it clear that it only wanted the connectionless side of the architecture while forging ahead with new research, it could have acted as a counterweight to the PTTs. The outcome could have been very different.[73] It would have been possible for a united research front to address some of the issues that there had not been time to consider in the Arpanet. The stagnation that now plagues network research might have been avoided. OSI was bogged down in politics, thus providing an example of the general rule that standards are no place to explore new concepts.[74] The Internet designers and promoters had no such constraints and had the opportunity to move ahead, continuing the development of the nascent networking concepts beyond the initial work from the mid-1970s. Instead, the Internet froze in the face of the PTT opposition[75] that was going on in OSI, relying on Moore's law and band-aids (under the rubric of "small incremental change") to solve its problems.[76]

The only alternative would have been for the computer companies to take the view that the trends were on their side and made the bold decision to not enter into a joint project with the PTTs in the 1980s. It is certainly the case that the lack of any kind of common ground between the computer companies and the PTTs doomed OSI. A purely computing-led OSI effort with the Internet as its research vehicle would have had more commonality in its vision and better technological solutions.

But there are a myriad of reasons why this could never have happened. First, it is simply not in the nature of committees charged with getting the minimal agreements necessary to maximize their members' individual market share to take strong stands, especially in such a competitive and uncertain environment. There were immense political pressures, especially in Europe where with no hint of liberalization it appeared that an onerous networking environment completely controlled by the PTTs would arise and ensure their continued dominance. Perhaps there should have been more faith in the PTT's track record of building products with very near-term market windows. The PTTs were woefully unprepared for the truly disruptive nature of the technology changes that were setting up to roil both the computer and communications industries. There would have been time to work out the concepts rather than freezing them so early as Frank Kuo had warned in 1974. There was much more at stake than minor details about a transport protocol. Even if ISO had not set up a joint project with CCITT, there would have been supporters of a PTT worldview in the OSI discussions. After all, it was the only environment most Europeans knew and the environment their products were built for, not to mention the discomfort with nondeterminism. The same arguments were bound to have occurred. Although they probably wouldn't have played as great a role as they did. And DARPA's decision not to accept the majority's decision would have also undermined any effort without the PTTs. This is especially problematic given how small the differences were between INWG 39 and INWG 61.

The real lesson here may be that in the heat of discussion people tend to focus on details that don't matter rather than the first-order effectors that do, an observation borne out by much research on organizations behaving under high degrees of uncertainty.[77] That was certainly the case here.

At the same time, it is not uncommon for seemingly small distinctions to be a major inflection point in a design: Make the right choice and the future is bright and simple. Make the wrong choice and the future is condemned to the hell of constant patching. It is sometimes hard to know which is which.

Acknowledgments

I would like to acknowledge the insights from discussions with Andrew Russell, in general and especially on IBM antitrust and the early packet-switched services; with Lyman Chapin, when I discovered Watson's

tracks in OSI TP4; and with Fred Goldstein on the timing of X.75 and the many colleagues I have bored with these ideas over the years.

References and Notes

1. A. McKenzie, "INWG and the Conception of the Internet: An Eyewitness Account," *IEEE Annals of the History of Computing*," vol. 33. no. 1, 2011, pp. 66–71.
2. As described in T. Kuhn, *The Structure of Scientific Revolutions*, Univ. of Chicago Press, 1962, an oft-cited but less often read work that notes that when a shift occurs old words take on new meanings, many previous "issues" become nonissues, and communication between practitioners on opposite sides of the shift becomes difficult. Networking saw all of these and more.
3. I was involved in INWG during this period, but not directly in the transport protocol discussions. In our group at Illinois, Gary Grossman and I split our participation: Grossman concentrated on transport, while I focused on the other two INWG projects. Although Grossman and I often discussed the transport protocol work, I was only an observer of the events covered by McKenzie's paper. In fact, for most of what is described here, I would have to classify myself as a firsthand observer, more than a participant affecting the outcome. That changes with the paragraphs that discuss when INWG 96 was contributed to OSI. Although still observing the forces at play, I was also a participant as the US chair of the ANSI OSI architecture committee and rapporteur of the OSI Reference Model in SC16 and later SC21 and held other roles in and out of the standards process, but was not directly involved in the transport layer work, other than to prevent the US from voting "Yes" on the first transport protocol ballot, which is another story.
4. This is a fairly common path in evolution: When a new "field" opens up, all of the major "body types" are tried early on and then whittled down to a much smaller variety. We see this in nature (all of the phyla were created in the Cambrian explosion), airplanes (initially there were a wide range of fuselage shapes, but now all airliners look the same), and computing (where there were a large variety of word sizes and now all word sizes are powers of two, if not multiples of 16). Networking has probably hastened this trend by forcing greater commonality across systems.
5. In many cases, the acronym is more common than what it stands for. For example, in the Arpanet, the gray box in the machine room was an IMP. Few thought of it as an Interface Message Processor, and many would have been hard-pressed to expand the acronym. We see this often, and sometimes it is betrayed by the fact that what an acronym stands for changes.
6. J. Abbate, *Inventing the Internet*, MIT Press, 1999.
7. Note that "statistical multiplexing" (invented about the same time) is really nothing more than packet switching at a finer granularity—that is, a small number of characters rather than packets—even though its early proponents insisted it wasn't the same at all. (This is another example of the data communications (beads-on-a-string) and distributed computing paradigms at work here.)
8. L. Wittgenstein, *Tractatus Logico-Philosophicus*. Routledge & Kegan Paul, 1922: "6.54 My proposals serve as elucidations in the following way: anyone who understands me eventually recognizes them as nonsensical, when he has used them—as steps—to climb up beyond them. (He must, so to speak, throw away the ladder after he has climbed up it.)"
9. Kuhn, *The Structure of Scientific Revolutions.*
10. It is interesting to note that, in all of the projects and start-ups that Paul Baran and Larry Roberts undertook after 1970, they all took a connection-oriented approach.
11. This had been shown rather dramatically in a paper by Peter Denning, "A Statistical Model for Console Behavior in Multiuser Computers," *Comm. ACM*, vol.11, no.9, 1968, p. 605.
12. E. Dijkstra, "The Structure of THE Operating System Multiprogramming System." *Comm. ACM*, vol. 11, no. 5, 1968, pp. 341–346.
13. L. Pouzin to J. Day, personal comm., July 2010.
14. As an example of a paradigm shift not being a single step, the theory of continental drift wasn't completely wrong. Continents did move. Plate tectonics was a refinement of the idea that the continents floated on plates, and it was the plates that moved. Getting to the paradigm shift in geology was not a single step. Most seldom are, and networking was no exception. Similarly, acceptance of the paradigm shift is also not a step function.
15. A paradigm shift is equivalent to a change in axiom sets. It is seldom used this way, especially in industry where it is more often a marketing term where any little technique is labeled a new paradigm.
16. This may not have been apparent in some of the early work, but it was the goal everyone was working toward. The focus initially on applying operating system ideas of resource allocation, layering, and replicating operating system functions: device drivers (Telnet), file system (FTP), job scheduling (RJE), an editor (Net-Ed),

the work of the USING group, the National Software Works project, the distributed database work at Computer Corporation of America and with NARIS, a land use planning system at the University of Illinois.

17. In computing, a special case is an exception (usually an extreme) to a set whose members all obey a common rule, while a degenerate case is one that appears an exception, but which still obeys the rule. Often this happens when the model for solving the problem is shifted from the obvious or naïve approach to a new perspective. For example, shifting the center of the solar system from the Earth to the Sun makes the motion of the planets a degenerate case, or my overly cute aphorism that a virtual circuit is just a very long packet shifts the model to make a circuit an extreme case that still fits a general model that all packets are routed independently. Or with Telnet, where a problem everyone else saw as asymmetrical host to terminal protocol was recast as an elegant symmetrical protocol.
18. I coined this phrase "beads-on-a-string" during the INWG years and have used it in lectures and papers ever since. It based on my discussions at the time with Europeans and on Pouzin's talks on the PTT issues. It was clear that the connection/connectionless issue was not the only aspect of the architecture differences that needed to be recognized. The phrase was chosen to capture the fact that the PTT proposals were often illustrated as wires stringing boxes together, rather than layers as a locus of distributed shared state. Even today, the preference for network diagrams as boxes connected by wires often obscures the root cause of problems.
19. For details on the PTT approach, see Abbate, *Inventing the Internet*; T. Rybcznski, "Commercialization of Packet Switching (1975–1985): A Canadian Perspective," *IEEE Comm. Magazine*, vol. 47, no. 12, 2009, pp. 26–31; and R. Despres, "X.25 Virtual Circuits – Transpac in France – Pre-Internet Data Networking," *IEEE Comm. Magazine*, vol. 48, no. 11, 2010, pp. 40–46.
20. At the debut of the Canadian Datapac X.25 service in Toronto in 1976, the project's director was asked during a Q&A session what he would think of companies building PADs to attach to their network. With atypical candor, he replied emphatically, "Not very much!"
21. For historians and/or younger readers, most peripherals, such as terminals, during this period had electronics consisting of logic gates (such as AND, OR, NOT, NAND or NOR, J-K Flip-Flops) but did not have a programmable CPU. Their design could be mystifying. To wit, an engineer in our lab trying to diagnose a problem with a printer interface was overheard to exclaim in frustration, "It has to have logic!" His coworker replied, "No, just gates."
22. Videotex was a sort of text-based Web with limited graphics, justified as a replacement for the telephone book. It let users book travel, hotels, and such. Here, to use today's terms, the only "cloud" was operated by the PTT. Following an established trend, the earliest, most popular, and profitable aspect of Minitel were the so-called "blue channels" (pornography).
23. Today, we would say "in the cloud." During this period, this was seen as a threat by the monopolies or the big players to own the Net. With the much more sophisticated marketing of today's big players, their hype successfully dazzles net neutrality activists to embrace the cloud as the ultimate good. They don't notice their nice jail cell.
24. Being monopolies, their market strategy could be characterized as, "If we build it, they have to buy it."
25. And who says net neutrality is a new issue!
26. It is interesting to note how, up until this time, the PTTs and IBM had carefully evolved their products so neither encroached on the other.
27. IBM had a worse problem. IBM products were never at the technical forefront. It had established its dominant market share by aggressively selling to the person who signed the check, almost ignoring the engineers who would use the equipment. In the 1960s and 1970s, the person who could sign for millions in computers seldom had much computing expertise. As the field matured and computer-savvy managers moved up the corporate ladder, this would change. The precipitously falling prices caused by Moore's law only accelerated the process. When I first made this observation in the late 1970s, I was told I was crazy. IBM would always be the dominant computer company. By the late 1980s, IBM was in big trouble. There was reorg after reorg and massive layoffs.
28. S.W. Usselman, "Unbundling IBM: Antitrust and Incentives to Innovation in American Computing," *The Challenge of Remaining Innovative: Insights from Twentieth-Century American Business*, S.H. Clarke, N.R. Lamoreaux, and S.W. Usselman, eds., Stanford Univ. Press, 2009, pp. 249–280.
29. The claim was that it had been done on purpose as a feint to throw off the competition.
30. This phrase was used more than once to describe them (us). "Prima donna" was also heard more than once, while "bellhead" was cast the other way.
31. At the time, this was primarily a difference in age cohorts. But over time that explanation does not

seem to hold; it seems that some people are just not comfortable with nondeterminism, asynchrony, and similar concepts.

32. Although it wasn't called that yet, and the doubling of capacity and halving of price every 18 months was not yet clear, most could see the progression from transistors to integrated circuits (ICs) to large-scale integration (LSI) and VLSI coming. The reason ARPA funded Illiac IV in the late 1960s was as much to push LSI technology as to build a supercomputer. The network researchers could see where this was going. This is another advantage the young crowd associated with the Arpanet had: they were seeing and acquiring experience with things that wouldn't reach product for a decade or more.
33. In these early days, the European members of INWG were probably much more aware of what was at stake than the Americans.
34. The PTT school of thought was represented by ISDN, Frame Relay, ATM, MPLS, and IMS.
35. In 1992, ITU said it reorganized to streamline the organization and eliminated the designation CCITT. Others said it was because CCITT was becoming much better known and more influential than ITU.
36. With deregulation, this has been relaxed somewhat in recent years, but old habits die hard.
37. For more details on INWG's interactions with CCITT and ISO, see A. Russell, *Open Standards and the Digital Age*, Cambridge Univ. Press, 2014.
38. Words are important in standards. ITU produces "recommendations" because, as a treaty organization, if it produced standards the signatories of the treaty would be legally bound to conform to them. With mere recommendations, however, conformance is not obligatory. ISO, on the other hand, being a *voluntary* standards organization produces standards to which conformance is voluntary. In the end, an ITU recommendation has the same intent and weight as an ISO standard.
39. LAPB (Link Access Protocol B). There were a whole string of LAP-x protocols in CCITT recommendations. HDLC was being developed in ISO and was derived from an IBM protocol used in SNA, called SDLC. This is another case where the acronym is more important than what it stands for.
40. Being different just to be different didn't help CCITT SGVII's public image at all.
41. L. Pouzin, "TC 6 Contributions to ISO," INWG 122, 15 June 1976.
42. The reader should keep in mind that the linearity indicated by publication dates, especially of standards, does not necessarily reflect the highly parallel and concurrent real events that occurred. Early drafts were often circulated and discussed well before the dates on the final versions or even initial ballots.
43. The biggest difference was the round-trip time. It is important to know when to expect a message to be acknowledged as correctly received. For a point-to-point protocol like HDLC, this is simply the time it takes to traverse the wire twice, plus a little time to generate the ACK message. There isn't much reason for it to vary much. In a network, each message may go through a different number of routers and be delayed a different amount at each one. Hence, the variation in round-trip time is much greater and the message may be discarded due to congestion. All of this implies much greater uncertainty.
44. There was a third difference that was thought to be important: In INWG 39 all packets had the same but a longer header format, while INWG 61 had different header formats for data transfer, synchronization, acknowledgement, flow control, and so forth. It was thought that parsing a single header format would require less code and be faster. If this approach is followed closely, it results in more complex code in the main path. The solution is to treat it as if these were different message types without the benefit of reducing the overhead. But the greater overhead was of considerable concern to those outside the DoD who could not afford the high-speed lines for their production networks.
45. Today most traffic has to use the extended sequence number option.
46. It can be relatively straightforward if the packet can be reassembled into a single large buffer (not always easy, especially in the 1970s when memory was always precious). It becomes more complex if the buffer strategy utilizes a chain of common sizes and reassembles the packet as a chain of buffers.
47. Some readers will think this is wrong because TCP only has a 20-byte header. This is correct. This was before IP was separated from TCP. When they were separated, the combined header length of TCP and IP was 40 bytes, so doubling the size of the sequence number would make it 48 bytes!
48. When the insights do come, they are seldom from where one expects. Given what Fleck says, one should almost expect them to come from the periphery rather than the core: L. Fleck, *The Genesis and Development of a Scientific Fact*, Univ. of Chicago Press, 1981; originally published in German in 1935.
49. I wish I had a dime for every time the claim has been made that "X is a whole new approach to computing," only to find the longer it is studied,

it becomes apparent that it isn't all that new or different.

50. One example is not using word sizes that aren't a power of two.
51. Working through this in any science is not simple. Often viewed after the fact, it may look like a simple linear story ("the influence of T.S. Eliot on the Shakespeare" effect, see D. Lodge, *Small World*, Penguin Books, 1984), but it certainly wasn't at the time and did not appear so to those involved. As with any human endeavor, the story is messy, with twists and turns, blind alleys, and many contributing threads, which of course is what makes it so interesting.
52. Issues like receipt confirmation and transport connection recovery are examples of functions, which were eventually realized to be nonissues and not part of the protocol.
53. R.R. Watson, "Timer-Based Mechanisms in Reliable Transport Protocol Connection Management," *Computer Networks*, vol. 5, 1981, pp. 47–56. Watson indicates the work was done in 1978 and known in the community by 1980, if not earlier: R. Watson to J. Day, personal comm., Mar. 2009.
54. In terms of TCP, SYNs and FINs are unnecessary.
55. To be technical, Watson's results imply that port-IDs and connection-endpoint-IDs should be distinct. All three INWG protocols overloaded a single identifier with those semantics and, in the case of INWG 39, with application name semantics as well.
56. G. Gursun, I. Matta, and K. Mattar, "On the Performance and Robustness of Managing Reliable Transport Connections," tech. report BUCS-TR-2009-014, Boston Univ., 2009; www.cs.bu.edu/techreports/pdf/2009-014-reliable-conn-mgmt.pdf.
57. G. Boddapati et al., "Assessing the Security of a Clean-Slate Internet Architecture," *Proc. 7th Workshop on Secure Network Protocols* (NPSec), 2012, pp. 1–6.
58. One reviewer suggested that this was a case of Brooks' Mythical Man-Month theory, which discusses the software engineering in large projects. Brooks argues that as the number of people working on it increases linearly, the communication among them goes up combinatorially. (F.P. Brooks, Jr., *The Mythical Man-Month*, Addison-Wesley, 1975.) This isn't that. This is a more subtle phenomenon: With some problems, it simply takes time for that flash of insight to occur to reveal a simplification or a deeper result. More people working on it seldom makes it happen faster. Derek De Sola Price first pointed this out, probably in D. De Solla Price, *Little Science, Big Science*, Columbia Univ. Press, 1965.
59. This is when computer scientists did computer science, rather than just finding something that works.
60. F. Kuo, "Political and Economic Issues for Internetwork Connections," *ACM SIGCOMM Computer Comm. Rev.*, vol. 5, no. 1, 1975. Also see INWG 52, Jan. 1974.
61. The PTTs and telecom equipment vendors had CCITT. They had no reason to play a major role in ISO, which was mostly working on programming languages, magnetic-tape formats, character sets (ASCII, IA5), and the like. But IBM had initiated efforts on SDLC/HDLC, so CCITT had a liaison just to keep an eye on things.
62. For an excellent discussion of the events leading up to the first meeting, see Russell, *Open Standards and the Digital Age.*
63. This was the product of a committee compromise.
64. There were three for X.25: one as seen by SGVII, one for SGVIII, one for the users of X.25; then one for the Germans and the only really needed that was based on INWG 96.
65. At this writing it is unclear why. It could be as simple as discarding the state as something that was not a great savings in space, especially if it would just have to be reallocated. Or possibly some delegates were uneasy with the idea of having ports allocated but no "connection" in place. (It does require the kind of leap of faith that committees are not known for.) TP4 does have a "keep alive" function that would effectively mean that state would never be discarded.
66. It is unclear that they saw Watson's work as anything other than a competing proposal or that they recognized that this was an implication of Watson's work.
67. It should be noted that changes were made to TCP, but they were to fix problems that arose. Nothing that worked "well enough" was changed simply because it was better. This could be a difference between the master craftsman versus scientific approach or simply a case of "not invented here." Regardless, textbook authors have preferred the myth to the facts.
68. Russell, *Open Standards and the Digital Age.*
69. In CS, there has been a common phenomena when something is found to be useful or successful (usually in the marketplace), there is a tendency for calls to use it for everything. It seldom turns out to be the case. In this case, examples of the extremes were protocols such as Trivial File Transfer (worked well in relatively reliable Ethernet environments but not in general) or SNMP. There were even arguments for having the application do it. In fact, one of the original arguments for a transport protocol was so the application didn't have to do it.

70. L. Smolin, *The Trouble with Physics*, First Mariner Books, 2006.
71. Most of the OSI connectionless faction, especially the Network Layer Group, were also participating in the IETF. There they were an influential but small group.
72. Several places in this article have emphasized the intensity of the debate. This cannot be stressed enough. The debates were heated and loud, and some nearly came to blows. The ugliest of political games were played to get an advantage. Academics in recent years have become fond of the word "tussle" to describe this. One can tell from the choice of the word that they were not involved on the front lines. "Tussle" doesn't come close to describing what took place and seeks to trivializes the intensity of the debate and what was at stake.
73. This is not an argument to resurrect OSI or that OSI was right all along. The OSI model had several fundamental flaws primarily introduced by the PTTs that made it untenable in even the medium term.
74. Although OSI was able to make some significant advances in networking, they generally occurred only when it could be done before the PTTs realized it was an important topic.
75. If not that, then what? It was clear that that the Internet would not adopt anything that wasn't TCP/IP. Then why didn't they complete their architecture? Surely not because they thought it was complete.
76. Russell, *Open Standards and the Digital Age*, also has an interesting interpretation of this.
77. M. Cohen, J. March, and J. Olsen, "A Garbage Can Model of Organizational Choice," *Administrative Science Quarterly*, vol. 17, no. 1, 1972, pp. 1–25.

John Day has been involved in research and development of computer networks since 1970, when his group at the University of Illinois was the 12th node on Arpanet. He has developed and designed protocols for everything from the data-link layer to the application layer, managed the development of the OSI Reference Model, and was a major contributor to the upper-layer architecture. He is the author of *Patterns in Network Architecture: A Return to Fundamentals* (Prentice Hall, 2008). He has also published widely on 17th century China and the maps of Matteo Ricci. Day currently teaches at Boston University and is president of the Boston Map Society. Contact him at day@bu.edu.