An Interview with

GÉRARD LE LANN

OH 420

Conducted by Andrew L. Russell

on

3 April 2012

Paris, France

Charles Babbage Institute
Center for the History of Information Technology
University of Minnesota, Minneapolis

Gérard Le Lann Interview

3 April 2012

Oral History 420

Abstract

Gérard Le Lann describes his first experiences with computers in the 1960s, and his work on the Cyclades project in France and the Arpanet in the United States in the 1970s. He explains how the field of distributed computing came into being in the 1970s, the conflict between the respective advocates of "datagrams" and "virtual circuits," and his collaboration with Vint Cerf and Arpanet designers at Stanford University. The interview concludes with some reflections on the management of innovation in France and the United States.

This set of nine interviews conducted with Tilly Bayard-Richard, Najah Naffah, Louis Pouzin, Marc E. Levilion, Michel Gien, Jean-Louis Grangé, Gérard Le Lann, Rémi Després, and André Danthine was funded by the ACM History Committee with a fellowship on "European Contributions to Computer Networks: An Oral History Project."

Russell: This is Andy Russell, I'm here with Gérard Le Lann,[1] and we are going to talk about his activities in networking and some other things. I want to begin by asking you about your education and your first encounter with computers.

Le Lann: My first encounter with computers was when I was studying at what we call an engineering school in France – *École d'ingénieurs*. That's after you do the *baccalauréat* when you are around 16. Either you join the university or you try to get admitted in an engineering school. The most famous one is *École Polytechnique*. There is a ranking, in the order of 40 engineering schools at the top. So I joined one. It's called ENSEEIHT. And that's where I got my education on computers, programming, languages, compilation, and all that. When I went through that, I had to do what young French people had to do – join the National Services. So I joined the Navy. I learned a lot about computers in the Navy because at that time, Charles de Gaulle, our President, had decided to launch the new military program, and the program involved the first nuclear submarine built by the French Navy (named *Le Redoutable*). That was the first time I had to develop software for a real-time mission-critical multiprocessor. That was the system in charge of controlling the ship and its armament. This project was really high-tech, and I learned a lot of things that were not taught in universities or schools at that time, especially real-time computing. I worked on U.S.-made (TRW) computers, those used by the U.S. Navy, bought by the French Navy. That's how I got acquainted with computers. Later on, after my national duties, which were over in 1969, I decided to go abroad. But I didn't go very far. In fact, I ended up being hired by CERN, the European Organization for Nuclear Research located in Geneva, which is just across the border with Switzerland. It was a very interesting experience because there you had something like 20 nationalities working together – top-level scientists, engineers, and nuclear physicists, which was a working context totally new to me. And that's where I got faced with computer networking for the first time in my life. That was in '69, '70, '71, and '72. At that time CERN was probably one of the very first organizations in the world – there were similar organizations in the U.S. – that had to design a computer network in charge of tracking

---

[1] Jean-Louis Grangé was present during the entire interview. Near the end, Michel Gien also joined the group.

the data that was collected in a particle accelerator, which was a loop. It was big. We had to install many mini-computers all around the loop that would just follow the particle flux and preprocess the pictures taken within the accelerator. This data was for the physicists to understand what was going on in the accelerator. I was in fact involved in something that was later on called computer network protocols – these mini-computers would ship data to a mainframe located in the CERN data center when their buffers are full. Processing the data while coping with the speed of particles was a very interesting challenge. <laughter>

Russell: It's interesting because both are real-time but are very different contexts – military contexts and then a pure research context.

Le Lann: Yes.

Russell: And then you moved to Rennes and IRIA?

Le Lann: In '72, I realized that I really wanted to do research in computers, computer science, broadly speaking, whereas at CERN, the primary discipline is nuclear physics. As a researcher in computer science, I didn't see a bright future for my discipline. So I decided to quit, which is something totally silly given the level of salaries at CERN. Nevertheless, I left. While looking for where to go next, I approached my previous colleagues at the French Navy, and one of them told me, "You should talk to someone named Louis Pouzin who is starting a new project located at IRIA (former name of INRIA)." It turns out that IRIA was familiar to me since in '69, the Navy had asked me and 3 other "Lieutenants de Vaisseau" to stay with them six additional months (beyond the nominal period) in order to complete the project for the French nuclear submarine. We said, "Okay, we like the place, and the work is interesting, but we are now entitled to be paid as regular engineers. So what are you suggesting?" They offered, "Find some organization, negotiate your salary, and we will pay them back for keeping you for the first six months." And IRIA was our choice. To summarize, I have been with IRIA temporarily in '69, prior to joining the Cyclades project end of '72. In summer of '72, I meet with Louis Pouzin, who tells me, “Your profile is interesting. You’re welcome to the Cyclades team if you want to join us. Now, given your name, you are from Brittany, right?” (That’s a western part of France.) “I don’t have a team in Rennes, the capital city of Brittany. What about going there and creating a Cyclades team?” I said, “Okay, interesting challenge.” That’s why I joined the University of Rennes first. The other appealing perspective offered by Louis was the possibility of moving to the States to be immersed in one of the ARPANET nodes. That was my second motivation for joining the Cyclades team. I really wanted to go to the U.S. and see how this ARPANET thing looked like for real.

Russell: I was going to ask how you first heard of the ARPANET. I think it’s the first time you’ve mentioned it in our conversation. Was it something you heard about through Louis?

Le Lann: Through Louis, yes. At the University of Rennes, I embarked on building an event-driven simulator for the Cyclades protocol. And of course I was reading the reports published by DARPA, on NCP, the early ARPANET protocols, routing, and so on, the very first official ARPANET documents. And that created a very interesting situation in the Boston airport actually, when I came first in my life to the US in ’73, carrying these documents with the DoD logo on the cover page (I was searched as if I were a spy, funny). We knew that there were problems with the Cyclades protocols. They were blocking for hard-to-understand reasons. I told Louis, “I have never worked on simulation before, but I think that simulating the protocols would give us the time dimension and see what the problems are.” “Okay, do it.” I was lucky enough to pick up Simula-67 for programming the simulation models – a very nice language by the way. It was a real pleasure. And magically, looking at the dumps from the simulation runs (packet losses, buffer overflows, concurrency, and so on, were simulated), we understood what was going on. For example, these protocols would confuse a failed connection with its reincarnation, both ends being ignorant of the fact that they were disagreeing on which connection was in use. Or losing track of which packets sent were actually delivered at the other end. This simulation work was published after I left Stanford.[2] It turned out that when I moved to Stanford in '73 – I'll get back to that later – working with Vint, we found out that the NCP protocols did not work well for the same reasons. That was the time of the inception of the error-and-flow control mechanisms, of the sliding window scheme, now at the heart of TCP/IP

Russell: Your crucial step was to make a model and then act from the model instead of…

Le Lann: Yes, rather than struggling with an implementation, you create a set of models that match the software processes of interest, what we called the Transport Station (TS) in Cyclades terminology. You need at least two of them for modeling the communicating TS. And there were other processes for modeling the network losing packets randomly, or deterministically, you know, "I want to lose this specific packet and see what happens." <laughter> And then you activate these models/automata. We could play with at least three automata at the same time and watch them on the time axis, changing states – sending, receiving – not sending, not receiving. And because you trace everything, you just have to read the outputs generated by the simulator. Simulation outputs at that time were a bit clumsy. But after awhile, you get used to it. That's how we found out what had to be modified in the protocols specifications. In fact, when the real causes are identified, you tell yourself, "It was obvious!" It's like the early days of GPS, which was not working initially as expected (way off the targeted accuracies), until the designers found out the culprit (a flawed assumption), an obvious overlook in retrospect. I don't want to get into technical details, but the references used for a connection between a sending TS and a receiving TS were something like, "So far, I have received n packets from you, and I'm prepared to receive another three". Unfortunately, the n packets received from a sender may not be the same n packets the sender has sent, because there are variable delays and the network might lose packets. Therefore, the TS were in disagreement. Clearly, lack of some sort of common time referential is not a good design approach in such systems. (There is an analogy with relativistic physics.) Absolute

---

[2] G. Le Lann and H. Le Goff, « Verification and Evaluation of Communication Protocols », Computer Networks, vol. 2, 1978, North-Holland Pub., pp. 50-69.

physical time could not be an option either. What else then? Ah, logical time! Unique identities! Here is a stream of packets (a "letter"). If you would number them sequentially, a packet with logical timestamp (integer) X cannot be confused with a packet carrying another logical timestamp. This packet might not arrive, but then you know. If you have received X+1 and X-1, you know there is one packet missing, and which one is missing. Positive-acknowledgment-and-retransmission protocols are based on this simple idea. At the same time you do error control, you can do flow control. A receiver would send {J, N} to a sender S, meaning "J is the highest identity belonging to the string of consecutive identities/packets I have seen so far, and you are allowed to send N packets beyond J". To S, receiving pair {J, N} only once or multiple times (due to repetitions needed for coping with packet losses) could not lead to confusion. Flow control is accomplished via 2 sliding windows that are kept synchronized, a sender throughput being determined by numbers such as N returned by a receiver.

Russell: This was completely foreign to the thinking of telecom people, I would assume.

Le Lann: Yes, you are absolutely correct. To put it simply, the telecom people were educated, were thinking, in terms of synchronous communications.

Grangé: If I may, the main difference was that this mechanism you describe was an end-to-end mechanism, which was totally ignored by the packet-switching network. As opposed to the telecom approach, in which each node has a trace. In each node, the link, the circuit, is built. They have some memory reserved and possibly some computing power reserved, and it makes it terribly complicated to establish a circuit, to kill a circuit. And you can have a circuit that is partially established, partially closed, and partially open. It's just a mess. So that is the essential difference between the two things. Here the only entities that know about the communication are the both ends.

Le Lann: Yes, what Jean-Louis is saying describes a consequence of the design model the telecom people had in mind. They were used to what was called "*continuité galvanique"* in French (continuous copper circuits) in the glory days of *téléphonie*.

Simple. There is a sender that sends out a pulse on a wire (copper, optical fiber later on), which propagates to the receiving end, with nothing in between. It's synchronous. If there is no one at the receiving end, the pulse is lost. You must have a sender and a receiver present at the same time, so that pulses can be delivered. Now, replace "pulse" by "packet". What the telephone people did – what Jean-Louis described – was to mimic exactly the same model of continuity inside a packet-switching network. You see why we had this "asynchronous packets vs. synchronous circuits war".

Grangé: It was called a virtual circuit, by the way.

Le Lann: They were not aware of the fact, which is nowadays so obvious, that we were heading for a communications world which is fully asynchronous. Fortunately, I can send you e-mails when you are not logged on your computer. Cloud computing is yet another more recent example. You can work with clouds. You don't even know where they are. So who is present is a meaningless question. <laughter> The telecom community did not really like the concept of networks which could store-and-forward packets, even for short amounts of time, the time needed for traversing a router that does dynamic routing. Now we have what's called delay-tolerant networks, where you store the messages within the network because you know the receiving end is not reachable due to mobility, or due to other concerns that might have to do with security. For example, you don't want to propagate a message because you're not sure about who might hear it. This notion of having synchrony encapsulated within the logic of asynchronous protocols was totally unfamiliar to the telecom community.

Russell: I imagine in the time period we're talking about – the early 1970s – these concepts would have been unfamiliar to most people in the computing world as well.

Grangé: I think the situation was very much reflecting the fact that there were two different communities – those people who had been doing *téléphonie* for many years, and those people like us. We knew that establishing some communication capabilities between computers meant resource sharing – dynamic resource sharing – which is exactly the contrary of the notion of a virtual circuit where you allocate resources in a very fixed and rigid way. Right? So it was two different worlds really.

Le Lann: I think you're referring to something that's very true. That even we – computer scientists – were unfamiliar at that time with what's now called distributed systems. These systems emerged in the late '70s. And this was the very first school on such systems. I'm sure it is the first one held in Europe anyway. <Le Lann shows a poster from a conference, "The Design of Distributed Processing Systems," held in Nice, France, June 26th-July 7th, 1978, organized by IRIA.> That is also an example of one of the first international conferences on distributed systems. <IEEE ICDCS 1979, Huntsville (AL), USA>. Back to the scientific question that you raise. The reason why you are right is simple: the early network protocol designers were pioneering with two – and only two – (end) processes. These protocols, which were developed in the Cyclades team, in the ARPANET community, happened to be a particular case of the more general asynchronous protocols and distributed algorithms that appeared later on. We did not grasp this reality immediately because we, computer scientists, were unfamiliar with the notion of processes "living" in two different worlds, communicating together via channels prone to viable delays and losses. Again, even though it's a bit incorrect, setting up an analogy with relativistic physics makes sense because meanings of events and states depend on the time it takes to travel, on how they travel and the "universe" they cross. That was already the real challenge with two entities communicating via a packet-switching network. In the second half of the70s, once acquainted with packet switching and knowing how to make sure that what is sent is received and there is no ambiguity between a sender and a receiver, we opened the box: what about an unknown number of communicating entities and what can we say about the global state of such a system in the presence of failures? Pretty soon, papers appeared, notably this paper, which is probably the first example of a formal impossibility proof in asynchronous networks.[3] The Fischer-Lynch-Paterson impossibility result applies to networks where delays are unknown – delays are finite, but you don't know their upper bound(s). Despite assuming

---

[3] M.J. Fischer, N.A. Lynch, and M.S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process", Journal of the ACM, vol. 32 (2), April 1985, pp. 374-382.

no message losses, processes cannot reach exact agreement (consensus) in the presence of process crash failures. That had been demonstrated previously for the particular case of two communicating processes – the ack of the ack problem. So it took us some time to understand the very nature of the problems arising with distributed computing and the fact that knowledge of current global distributed states is impossible. If you are willing to "pay for" computing cycles, communication delays, etc., then you can construct a posteriori what has been some past global state. But it's not sure that this "some past state" you're going to build will be the state you're interested in. Algorithms permitting to build consistent past histories or snapshots of distributed systems appeared with the advent of distributed databases in the early 80s. But in the 70s, that was *terra incognita*. We were discovering new problems. It was even worse for the telecom community.

Russell: Do you suppose that your time with CERN helped you think about these things? You've made the analogy with physics.

Le Lann: Not really. The main concern at CERN was not theoretical computer science. We had no problems with assuming Newtonian physics. Contrary to the story about GPS. It took two years to understand why they couldn't get the accuracy they were shooting for, until someone said, "Guess what, our design model should have been relativistic physics!" At CERN, I was in a project where the main issue was to keep pace with the circular accelerator. When the particles (moving at a speed not far from light speed) hit certain obstacles (in the 70s, grids inserted in the accelerator) or other particles, sub-particles which result from collisions are traced, physical traces translated into data, which data is stored in a nearby "real-time" mini-computer. The particle beam comes back pretty soon. So the main concern was how to handle data acquisition with these mini-computers we had in the early 70s at CERN and their 128 Kbytes memory. Buffers fill in very quickly, and you better have protocols that move data out somewhere else (a "big" mainframe computer) pretty fast. In fact, this main concern – building a network that will keep pace with events occurring in a particle accelerator – has led me to learn much about networking issues.

Russell: You mentioned earlier that you knew about the ARPANET, and you wanted to immerse yourself in one of those nodes. How did the opportunity to get to Stanford come up?

Le Lann: That was before I got interested in distributed systems. I moved to Stanford in June ’73, after a visit I’d made in April ’73 in the US, when I also visited Bob Kahn, DARPA, in Washington. I was very impressed. I have to tell you this now because one of the questions you listed in your e-mail was, “What were the differences between France and the U.S. in terms of managing technology and projects at that time?” Well, it is different in France now, fortunately. But listen. In ‘73, I got some budget from a French public organization (say D) in support of my research. And to be frank, I still don’t know why I got this money. One day late ’72, I had a visit from a representative of D in Rennes. He asked, “You are a member of the Cyclades network, what do you want to do? And how much money do you need?” I started explaining the details of the protocol simulations. Pretty quickly I realized that this gentleman was bored. He said, “I have the money. Now you have it. And send us the final report please.” Of course, I was very happy. But I was surprised also, since this project supervisor did not seem to care much in some sense. A few months later, I met with Bob Kahn. I didn’t know much about the ARPANET community at that time, so I thought, “He is the boss, he must know about how to share/allocate the budgets. But maybe he doesn’t know much about communication protocols”. Ha ha! Big mistake! I was talking to someone who knew more than I knew about communication protocols, and who could open his… *Comment est-ce qu’on dit un tiroir?*

Grangé: Drawer.

Le Lann: Drawer. Thank you, Jean-Louis, and use the “public checkbook” for allocating sizeable budgets to U.S. universities and high-tech companies. The same person had the Knowledge – capital K – and access to nationwide Money – capital M. Which was something absolutely impressive for a French citizen at that time – it was either K or M, not both! So, early ’73, I visited BBN first, in Cambridge, then DARPA, then Leonard

Kleinrock (UCLA), and Vint Cerf (Stanford), and I came back. I talked to Louis, telling him, "Well, thank you for the tour. It was quite interesting. I think I'll join Stanford." In fact, with Vint, it was magic. We happen to be born almost the same day. And we got together quite well. It was really, really great to be there for one year. What was great, though, in addition to the personal considerations, was that Vint was absolutely passionate with what he was doing, with what was accomplished at one of the very early ARPANET nodes. I remember sitting down in his house in Palo Alto for hours and hours after dinner, looking at these results from the simulation runs I had completed, and talking, talking. And we also discussed the datagram concept from Louis. In one of the documents I've sent to you – the 1974 paper by Cerf and Kahn on TCP, the origins of the sliding window concept are explicitly referenced. Somehow, the Cyclades people have "exported" it, to be instantiated into TCP. At the end of my stay in Stanford, I had completed the writing of the first French book on computer networks, which I taught when back at the University of Rennes. And I had been exposed to the very early work on local area networks – Bob Metcalfe (Xerox Park) happened to visit us. Also, at that time, at the University of Irvine, there was a team who was developing the very first token ring.

Russell: Dave Farber.

Le Lann: Dave Farber. Bravo! I happened to read a paper from them. So, I returned to France with the idea (of course) of continuing my work on Cyclades – and then Internet later on. But I had the seeds of distributed computing already in my brain. And I was so much interested. So I left Cyclades in 1978, stopping my research on large computer networks and communication protocols, and started my research on distributed systems. So many interesting theoretical, entirely new, problems. Everything had to be done "from scratch"!

Russell: I want to learn a little bit more about how you spent your time at Stanford as well. You mentioned, for example, meeting with Bob Metcalfe…

Le Lann: Indeed, we have to talk about the events that were happening outside mainstream ARPANET.

Russell: Before we do that, and while we are still talking about Stanford, can you tell me what you remember about the origins and genesis of the Cerf/Kahn TCP paper?

Le Lann: The paper appeared in '74 in the IEEE Transactions on Communications.[4] It went through the reviewing process quite quickly because the authors were already "reasonably" famous… <laughter>. Besides building on the massive work conducted by the Arpanet community, authors had direct and indirect discussions with the Cyclades team in France, and others (in the UK, notably). Cerf and Kahn have "concealed" the knowledge and experience gained through the Arpanet innovations, as well as what emerged from interactions with those European teams which had built their own packet-switching networks. Interconnection of heterogeneous networks was the main target. Accidentally, I was at Stanford when the paper was written, and Louis used to meet regularly with Bob and Vint. So I guess all that explains why the Cyclades work is explicitly referenced in the paper, notably the datagram concept and the sliding window mechanism.

Russell: Who else was there? The paper is a product of conversations with others in the ARPANET community, I assume. Were you the only international person there?

Le Lann: We were a total of 3 non US persons in Vint's Arpanet team, myself included. A plaque has been installed at Stanford University in 2005, with the names on it.[5] Regarding the Cyclades team, I have been the only one.

---

[4] V.G. Cerf and R.E. Kahn, "A Protocol for Packet Network Intercommunication", IEEE Transactions on Communications, vol. 22 (5), May 1974, pp. 637–648.

[5] http://mercury.lcs.mit.edu/~jnc/plaque.html

Grangé: Well, we all traveled a lot to the U.S. I did a lot of travels to Canada as well and other countries. But I think you [looks at Le Lann] were the only one to have stayed for a long time.

Le Lann: At an ARPANET site, yes.

Russell: Because at the time – and this is something I've talked to Louis about, and I'd like to talk about with Hubert and some of the others – there was INWG, the International Network Working Group. And the dream from the start was to merge the two or at least come up with a consensus protocol. Many of these papers were circulated within INWG, even before publication. Were you part of INWG as well? Did you go to the INWG meetings?

Le Lann: Not really. Maybe one or two. But it was not my… No. I was not involved, not very much interested into these international standardization things.

Grangé: That was mainly Hubert and Louis.

Le Lann: Yes. In fact, looking back, I wrote something about it (in French, I've sent it to you). There, I talk about ISO/OSI, also. It's a personal bias. I've never been convinced that spending so much time in so many meetings with standardization bodies is the most efficient way of making things happening. I know someone has to do it, and I'm very glad that some other people were willing to do it because I was not very much interested. So I'm not criticizing. Rather, I'm criticizing myself in some sense. But it takes so much sweat to convince a group of people in a room compared to convincing one person, who understands, and who says, "I buy it. Let's do it". I find the second option more interesting. Now, looking back… What is the real output of all that? We still have TCP/IP. What about the other proposals which were to be "influencing" TCP/IP in order to get a common, worldwide-agreed protocol? They disappeared. The reason why this could be the only outcome was not very well understood at the time… It's easier to say now. I never discussed this with Jean-Louis or others at the time because I didn't see it myself. What looked really smart originally was the reasoning according to which a protocol other than "plain TCP/IP" was needed, since TCP/IP would be a proprietary product, which is not admissible because an international protocol has to be universally usable, free of charge, by everyone on this planet. But that reasoning became flawed when, surprise, in 1983, our U.S. colleagues had the very good idea to put TCP/IP in the public domain. We were not familiar with this concept at the time, so much accustomed to proprietary software/protocols versus standards. And there was this third kind, called public domain! Metcalfe and Boggs did the same thing with Ethernet. Despite few technical weaknesses, TCP/IP has won because of that, being well-tested and promoted by a powerful community.

Russell: It was interesting to hear Marc Levilion talk about this from IBM's perspective, about all of this effort, all of these meetings to go to in OSI, and AFNOR, and so on. And then in the meantime, this thing – TCP/IP – just appears, and wow, it's free!

Le Lann: <laughter> Exactly. I could tell you the same story with Ethernet. Ethernet was very smartly designed, and backed by Digital Equipment Corp, Intel, and Xerox (Intel developed the early chips). Xerox knew they would make money with "things" connected to Ethernet, not with Ethernet per se. They said, "We own the patent, but you can own the right to develop Ethernet stuff. You just have to pay $200 <laughter> to buy the documents from the IEEE, since Ethernet is 802.3 standard, and pay us some marginal royalty." And then we witnessed the same reactions we saw *vis-à-vis* TCP, from competitors: "It is their thing. We have to do something." So IBM started working on the token ring. The automated factory community – General Motors, the MAP initiative – decided they would bet their future on token buses. We were asking, "Why all that? Ethernet is based on very simple ideas. It works fine. What is the problem?" [They said] "Ah, real time! It's not real time." Wonderful! We had solved the problem at INRIA, with a deterministic variant of the original Ethernet (patented and deployed on French Navy units and in civil plants and sites), which consisted in replacing the stochastic binary back-off scheme of the original patent with a deterministic tree-search algorithm.[6] That amounted to changing 5% of the original specification. But what we kept reading and hearing was, “Token rings, token busses, are better.” Six years later, these token passing things had disappeared. It took me some time to understand why I was invited so often by IBM France, to give lectures on my work. In 1977, I had published one of the very early papers on distributed computing – which paper is here, describing the concept of a token passing protocol aimed at a virtual ring, built atop a meshed network of arbitrary topology.[7] IBM wanted to check whether this paper could be a “killer” for their patents. They already had to enter into an agreement with a Swedish citizen who had implemented a ring based on the same idea. This idea that it’s only standardizing bodies that make things happening, well, it’s the case from time to time. It is fine for well-tested technology. But when you are pushing for new stuff, innovative ideas, this is not the most efficient way of changing the world.

The other technical observation of relevance is that the ISO/OSI work was based on a strict layered architecture model. In the 80s, looking at how difficult it was to get agreement between IBM, Digital, and the others, regarding what should be this transport protocol meant to replace TCP (i.e., 1 layer only), you could wonder how long it would take to reach a general agreement on an architecture of 7 layers! Moreover, you could question whether this nice decomposition in layers would make sense in the end because you had the implementers telling you that they did care about layers. They wanted to optimize the code and aggregate three layers into one piece of software, just to give you an example. And this is even more the case now. When you look at some criteria which were ignored in the 80s, such as mobility and security – a quality of service which makes a lot of sense nowadays, I would say – and you read papers published on these topics for the last 12 years, you see something which is called cross-layer. Cross-layered architectures violate the layer concept. Why? Let me give you a simple explanation regarding mobile networking and the 3 lowest layers. Layer 3 takes care of routing, layer 2 encompasses the link and the MAC sub-layers, and these layers don’t interfere with

---

[6] French patent INPI 84 16957, November 1984, US patent number 4,847,835, July 11, 1989 (G. Le Lann and P. Rolin).

[7] G. Le Lann, « Distributed Systems: Towards a Formal Approach", Congress of the International Federation for Information Processing (IFIP), 1977, North-Holland Pub., pp. 155-160.

each other if you follow the OSI bible. If you do that, then you develop networks for mobile nodes that will be incredibly inefficient, if they can work at all. Why? Because the time at which you should submit a message to the MAC layer depends on where you are, if you want avoid unbounded contention on wireless channels. A geo-localization driven MAC-layer protocol means that the MAC layer depends on layer 3! That is particularly true with mobile sensor networks or ad hoc vehicular networks. You cannot send a message at the MAC-link layer at any time because there are other nodes like yourself moving around which want to send messages as well, and if these messages are sent at the same time, you create congestion due to destructive collisions occurring in radio networks. So you see, we are far away from the simple OSI model! When two layers meant to be independent must be "crossed", you know there is something wrong. Suspicions we had in the 80s about the ISO/OSI model turned out to be correct.

Russell: I had cut you off earlier, when you were talking about some of the other things that were outside of the mainstream story, like Ethernet.

Le Lann: Ethernet was being prototyped at Xerox PARC, south of Palo Alto, at the time I was with Stanford. We talked with Bob Metcalfe a number of times. Vint Cerf was very excited, and so was I, with this fancy idea derived from the Aloha radio network in Hawaii, developed under the leadership of Norman Abramson. Rather than having the plain ether, you would "simply" install a cable. This idea of having a passive bidirectional communication channel on a cable – and contention handled as a multi-access game – was totally new. Mainstream ARPANET was not considering this problem since a point-to-point or an end-to-end link is not multi-accessed. You only have to coordinate two entities. On Ethernet, you have a broadcast medium and, at any time, some unknown number of competitors wanting to send messages. How to make sure that they can be synchronized so as to avoid repeated collisions, which collisions entail wasted bandwidth and no communication at all? That was a new problem (distributed mutual exclusion), a very interesting one.

Russell: They started thinking about satellite as well, but you said you didn't interact too much with the ARPA satellite people?

Grangé: No. I don't know exactly why. As I mentioned, we worked mainly with Canadian people. Actually, the questions we were looking at were more related to specific satellite communication properties based on broadcasting, high-bandwidth and long propagation delays. I think what we tried to do is to design experimental protocols that were specifically adapted to those characteristics. As far as I can remember, I don't know if we had any cooperation or relationship with American teams on this subject. I think Americans at that time were more interested in using communication satellite for *téléphonie* and point-to-point communications, not necessarily high-speed.

Le Lann: I may be wrong, but I think I remember the main practical reason was linking Norway with the ARPANET. The first ARPANET node installed outside the U.S. was located in Norway for obvious cooperation purposes of another kind at the time. That was the first time packet switching was experimented on satellite links. This type of networking was perceived as yet another threat by the French PTT and the big telephone monopolies, which were making money with "rented long haul circuits." <laughter> When the Cyclades team expressed its interest, "And now we would like to do packet switching on a satellite!", the response from the French PTT was, "No way. It's too expensive anyway. You will never have it (the rented line)."

Russell: What about packet radio? I know you mentioned packet radio in the U.S., but in France?

Le Lann: In France, packet radio experiments came much later, after the experiments in the US. I was not involved in PRnet, a packet radio network that was developed in '77. Since I used to come back regularly to the US for attending conferences or visiting Stanford again, I happened to learn about the experiment in the Bay area. The idea was to have a network of mobiles moving around and communicating with the ARPANET via a radio gateway.

Russell: This was the problem that led Cerf and Kahn to TCP/IP, the story goes: three separate types of packet networks that they wanted to interconnect.

Le Lann: [Refers to printed copy of ARPANET map] Norway, here, ARPANET and PRnet there. You are right. Add to this the emergence of local area networks–something like the Dave Farber-Irvine ring or Ethernet. I thought that there was something very important for the future there. Others believed that wireless nets had no future. We know now that there are standards for all kinds of mobile packet radio networks, pioneered with PRnet.

Russell: In our correspondence before this interview, you mentioned some things about the French authorities, those responsible for R&D investments and the atmosphere in France. As you've described it, certainly it was different from the ARPA approach.

Le Lann: A different perspective and approach, no doubt. At that time. Might still be the case in some circles, but the general mindset in France has evolved, fortunately. The major difference I think lies with the management of innovation… <Welcomes Michel Gien to the room, and speaking in French> So I guess the main difference is the way innovation is managed. In France, unfortunately, we did not have the equivalent of DARPA. When I met with Bob Kahn and Vint Cerf, I learned how NSF and DARPA managed their R&D support, financially speaking, and how they decided which topics had to be supported for the next five or ten years, why this topic is over, meaning they are not going to support research any longer on that. This is accomplished via university experts, essentially. So, as a researcher, you are assessed by your peers. They know what you are talking about. And when they meet, they scrutinize proposals according to scientific criteria. This leads to the appropriate choices most of the time. Unfortunately, we did not have the same processes in France, nor this kind of committees. Decisions within the Ministry of Research, or the Ministry of Industry, were made by people who receive recommendations from the next layer up. How these recommendations are formed, nobody knows. Too often, they depend on what is believed to be fashionable.

Back to our topic, networking. It’s only in ’91 that the existence of Internet was officially recognized by the French authorities. In France, in ’91, a few researchers were conducting research on networks and distributed systems without any official support – Cyclades had been ended in’89. Those who wanted to do research on networking had to find their own budgets. And then, all of a sudden, in ’91, because some top-level member of the French government said he had been presented the Internet, and he liked it, every public laboratory claimed, “Oh, great! We have been doing networking for many years. We have a lot of experience!” So you see, it was pure politics, no real visionary perspective. The Nora-Minc Report is a very good example. In 1978, among various white papers aimed at the future of “télématique,” President Giscard d’Estaing – not an expert in computer networking, of course – picked up a report written by these two brilliant <laughter> technology forecasters, who argued that France had to invest in a video device (the Minitel) which would permit every citizen to get access to info services via “the” telephone network. We have bright people coming from the very top of our engineering or administrative schools. Those who join the political crowd for their career are those who influence strategic decisions, made out of their own past and current culture. Unfortunately, unless you are a professional, no culture can be up to date given the speed, the *vitesse*, the pace of technology innovations. Decisions made out of politician considerations in innovation-driven areas are always late by one battle, as we say in French. And there is no real continuity in terms of investments. Typically, every sizeable research project/team supported by public money is launched for three years, with a budget that covers the first 18 months only. There is a review at midpoint, and if reviewers are happy with what has been done, the project/team receives the second slice of the budget. And then, after three years, it’s the end, the team is disbanded. People involved in start-ups, in creating new businesses, or researchers, have to struggle with a very ugly problem. You set up some skilled workforce; these people understand what the ultimate target is; and they know how to proceed. But if you don’t succeed in getting a brand new contract at the end of these three years, then the only thing you have to do is to dissolve the team and start again later on. This is not knowledge conservative. You keep re-building teams all the time. How can you compete with another country which has long-term sliding plans for supporting innovation? Pure theoretical research today translates into applied research five years later, and irrigates business 12 years later. We have a hard time at importing this model in France.

Gien: There is a new program being started by the Obama Administration about technologies merging, biotechnology and computer and… I just read about it a few days ago. It's a huge program. It makes me think about the ARPANET programs to revive the whole industry, computing industry, including research. It's a kind of way to converge many different programs to a common theme. It's around bringing technology to people, so it also matches biotechs and many nano-technologies and new communication mediums and all kinds of… Big data, that's what they call it. It's big data at big scale. How to use all these algorithms and all these data to make all the fields in the lifetime better for people. It sounded to me like some kind of a big long-term program that will allow convergence of energies in various fields.

Le Lann: And there is another difference between how this happens in the U.S. versus in France, or how this is managed and doesn't happen in France. Why is it that Facebook appeared in the U.S.? Remember: Arpanet was not developed to do Facebook or YouTube. Nevertheless … I was talking about planned R&D supported by governmental bodies. But there are individual "miracles" with French authorities. Sometimes you find someone who is really smart and who makes the right long term decision – this happened to me a few times. Unfortunately, this is more the exception than the rule. It is more *ad hoc* in France, depending on who is in charge. The other difference is in the ecosystem. Could Google or Facebook be born in France? Not a chance. There are ecosystems in the States which favor these kinds of undertakings. In France, too often, you have to "fight with" people who do not want to listen because what you are telling them (which they understand, and they suspect that you might be right) is so much disturbing. They have things to do tomorrow, and they don't want to take chances.

Russell: "Risk averse" is a term that comes to mind.

Gien: It's the aversion to change. And so I was discussing at lunch with a professor at CNAM. It's a kind of university for people who are returning to university, in the evening, after they are done with their job. CNAM was created, I think, just after the second world war or something like that. And it's a wonderful thing because it's really about turning workers into engineers. A worker in the factory who wants to get back to education can do that. And they're supported after the working hours and all this, and they graduate. And he was telling me that nowadays, with this big economic crisis on us, everybody talks about educating their workforces to change their directions. He said that nothing can happen in his organization. It's totally frozen because people have been installed, and they're here to stay. And when people come with new ideas or re-orientations, they disturb the day-to-day regime. And that brings in all sorts of problems. They prefer to ignore it. And so it doesn't change. That's something that is very strong in our country. It's not only France actually, but in particular in France. Whereas in the U.S., people are much more… <simultaneous talking>. They have a lot of opportunities for creating new things. Many of them fail. A few of them succeed. But there's dynamism. In France, it's difficult to bring in dynamism.

Grangé: So we have much less failure as well. <laughter>

Gien: Much less failure. Well, we have a gradual, slow failure. <laughter> That's why people here like it <simultaneous talking>… That's why people like these big programs because they cannot get a direction, and so it doesn't spin off naturally. It's difficult. We were exceptions, you know, with the Cyclades project and all the start-ups. Now they try to bring this kind of new culture, there is the dynamism in the young people, in the students, but there is no dynamics in the financial markets to support that anymore. <simultaneous talking>

Grangé: We were the exception, but we got killed anyway! <laughter>

Le Lann: I can give you a particular example, which is current. Ground transportation. If you are in that business, you have read– and maybe your boss has told you to read the news, that there is something called autonomous cars. Automated driving is in our future. Depending on your philosophy, you can go for full autonomy (Google's choice for the moment), or for coordinated automated driving. No matter what, you know that there will be sensors, radars, devices, cameras, and communication technology – packet radio – installed in future vehicles. Vehicles need to communicate together, in order to avoid bumping into each other in non-line-of-sight conditions, for example. Two years ago, one French car manufacturer publicly said, "We don't believe in vehicle-to-vehicle communication." Two years ago. The IEEE and ETSI have been working on a standard for V2V communications for the last 5 years, and that statement was made just before Google would make their first announcement of what they'd been doing in their backyard, secretly. Granted, their driverless car is able to operate correctly under certain specific conditions. Nevertheless, this is the beginning of a new industry. A first observation: It is not happening in France. Second observation: This is prompted by a company that has nothing to do with car manufacturing. How is that possible? Google interested in cars! Everyone was surprised. When you think a little bit, then you understand. You see where the matching is. And where is the business for Google. Implications are far-reaching.

Another example. I'm currently reading a report which is meant to be delivered to CNRS, on the future of electricity production and distribution in France (smart grids, etc.) There is a chapter on electric cars. Why is it that Google has chosen a Toyota Prius for demonstrating autonomy? Huh, a hybrid car… Fashion or something else? Something else! Google has a subsidiary called Google Energy. This company is acting as a future distributor of electricity to whoever is willing to buy electricity from them, energy in the future. This is a by-product of energy-centric concerns related to their huge business data centers and server farms. And there is a "link" between future energy networks (another industry revolution) and future vehicular networks. So we have here two examples of a company which was started for specific businesses in the cyber space – indexing, searching data, data mining, etc. – and which ends up being also as a potential big player in the cyber-physical world. This is the visible part today. I'm not sure there are many people in France who knew about the existence of Google Energy two years ago. Why is it that in this report for CNRS, there is no chapter devoted to the essential role cyber

networks and asynchronous store-and-forward “energy packets switching” technology will play in future energy networks? Are we going to miss this (r)evolution? Google has this potential, this momentum for looking at long term innovations in different domains, based on their own will. No public intervention. They are not begging for money. In France, too often, we have this bad habit of begging for money – how the solar panel business was started (ending in a failure) is a recent example.
In the very early 80s, when we patented the deterministic Ethernet technology (fielded by the French Navy and then later on by French and European space agencies), we visited the States, to meet Intel, AMD, and other VLSI chip manufacturers, for exploring the possibility of producing deterministic Ethernet chips. Intel offered us not to return to France, but rather to stay in Santa Clara, and be in charge of a start-up they would launch. They were offering a lot of money. Maybe I should have accepted! Another company we visited was located in Tucson (AZ), a subsidiary of a big French group, the latest attempt made by French companies to be in the chip business. [They said after our presentation] The meeting ended with, “It was very interesting, and we have to think. But where is the money?” We asked, “What do you mean, where is the money?” Response, “We are not going to invest our own budgets. You must have money provided by the French government for us to do that.” Do you see the difference with Intel?

Grangé: It’s less and less true because there is less and less money!

Le Lann: Yes. That’s why it is not happening so often now. Don’t forget that we were alluding to public money. French entrepreneurs have understood that times were changing. Which is good. Let’s get back to networking.

Russell: Yes, we only have a few minutes left. Do you have some final thoughts on networking, on your experience with Cyclades or afterwards? I feel like we only scratched the surface here.

Le Lann: It’s hard to go into details. It would take more than one hour and a half. What’s fascinating is that people use networks as simply as they drive their cars. TCP/IP has been around for 30 years – despite some drawbacks and deficiencies, but it's still there. It's like thermal motors, invented a century ago. People drive their cars, and 90 percent of them don't understand the principles of combustion engines. It's the same with networking. Maybe they know the name "TCP", but they don't know what's inside TCP. And they don't care. So much has been added atop basic and innovative technology. There is this historical dimension, which I like, that researchers and engineers do change the world. So, yes, seminal work on networking has resulted into a real revolution. We've been happy, lucky, to be part of it. And basic protocols are still with us, for how long we don't know, but… Maybe when we are dealing with intergalactic networks, we'll have to replace TCP. Mobility also involves a lot of changes. For example, in intelligent vehicular networks, you cannot de-couple issues of safety, security, and privacy, when you design networking protocols. I don't want to be traced when I'm travelling in my automated car. However, my car will be doing beaconing, and message broadcasting, because this is mandated by the law (safety authorities). Other cars have to be aware of my existence. I cannot hide myself. But I don't want to be spied upon. How to reconcile these totally contradictory goals? As regards messages exchanged in the course of safety-critical scenarios, acceptable latencies for successful deliveries are very, very, small. You don't have the luxury of waiting for acknowledgments. In other words, you cannot bet (your life) on positive-acknowledgment-and-retransmission protocols… Again, we are invited to invent new protocols!

Russell: Thank you, I see that we are out of time but perhaps we can take this up again another time.