The Networking Philosopher's Problem

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

This article is an editorial note submitted to CCR. It has NOT been peer reviewed. The author takes full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

While computer networking is an exciting research field, we are far from having a clear understanding of the core concepts and questions that define our discipline. This position paper, a summary of a talk I gave at the CoNext'10 Student Workshop, captures my current frustrations and hopes about the field.

Categories and Subject Descriptors

C.2.0 [**General**]: Data communications

General Terms

Algorithms, Design, Management, Reliability, Security

Keywords

Education, interdisciplinary research, network architecture

1. INTRODUCTION

This is an exciting time in computer networking. The Internet is one of the most influential inventions of all time—a research experiment that, within our own lifetimes, escaped from the lab to become a global communications infrastructure. The network has ever wider reach, with nearly two billion users today and a future that promises to connect more people (in more regions of the world) and more computers, phones, and sensors. We see seemingly non-stop innovation in applications (from the Web to peer-to-peer, to social networks, to virtual worlds) and link technologies (such as fiber optics, WiFi cellular networks, WiMax, sensor networks, and so on). Computer networks have become increasingly diverse, including data-center networks, vehicular networks, ad hoc networks and overlays for circumventing censorship, and smart phones connected by cellular networks. That's pretty awesome.

So why is it, every two years when I teach my graduate networking class, I go through so much angst about what to cover and, more generally, what my chosen field is really all about? This paper is part of "how I spent my summer vacation" before teaching my graduate networking class at Princeton in the fall of 2010. The paper is also an outgrowth of a talk I gave at the ACM SIGCOMM CoNext'10 Student Workshop[1], and a great discussion with several of the audience members after the talk. In this paper, I try to capture my frustration in trying to crystallize the essence of our field, edifying comments I've heard over the years from colleagues in other research communities, and my "best effort" to define networking as a field of scholarly inquiry—beyond its clearly exciting connections to other problem domains (e.g., the smart grid and automobiles) and disciplines (e.g., distributed systems and programming languages).

2. WHAT (I HOPE) NETWORKING IS NOT

What *is* networking? Is it just a *plethora of protocols acronyms*, mostly consisting of three or four letters. (Except for IP, which is special enough to have just two letters, a two-letter wonder much like those celebrities that go by a single name, like Madonna and Cher[2].) Networking is so often known and (not) loved through its acronyms, for everything from routing protocols (e.g., BGP, OSPF, IS-IS, RIP, EGRP, PIM, MPLS, and LDP) to applications (e.g., HTTP, SMTP, FTP, NNTP, POP, IMAP, SIP, RTSP, RTP, and RTCP), from the transport layer (e.g., TCP and UDP) to enterprise networking (e.g., VLAN, ARP, MAC, DHCP, VTP, NAT, and STUN) and wireless networks (e.g., CDMA, 3GPP, OFDM, UTRAN, RRC, UMTS, eNB, GPRS, RNC, MSC, DIFF, SIFF, P-CCPCH, HS-DPCCH, and CD/CA-ICH), just for starters. Is this really what we are all about?

Or, are we a *heap of header formats*, for Ethernet frames, IP packets, TCP segments and UDP messages, and application-layer messages? Did you know that you can build a colorful version of the TCP and IP headers using Lego blocks? (This is actually quite challenging since, while Lego blocks come in many colors, the important *one byte* fields are only available in six colors!) An electrical-engineering colleague of mine at Princeton once said to me, "I took a networking class in college. I fell asleep at the start of the semester with the IP header on the screen, and woke up at the end of the semester with the TCP header on the screen." 'Nuff said. Networking *must* be more than this.

So, perhaps we are a *big bunch of boxes* that perform various functions on packets, flows, or TCP connections? Routers, label-switched routers, gateway routers, route reflectors, and route servers. Switches, bridges, base stations, firewalls, and NATs. Packet shapers, WAN accelerators, intrusion detection systems, packet sniffers, scrubbers, proxies, load balancers, and so on. Some of these boxes differ only in subtle ways, such as which bits in the packet (e.g., the destination IP address, the destination MAC address,

[1]Slides at `http://www.cs.princeton.edu/~jrex/talks/conext-student10.ppt`

[2]Yes, I know I am showing my age. Who the heck is Lady Gaga anyway?

or the TCP/UDP port numbers) are used to index a table that determines how to forward, drop, or modify the packet. Aren't these boxes just marketing terms—"market"-tecture rather than architecture?

Or, are we really defined by a *ton of tools* we use to measure and manage networks. We've got a family of familiar measurement tools like ping, traceroute, nslookup, tcpdump, wireshark, iperf, ntop, trat, mrtg, dig, dummynet, and nmap. We also have a cohort of configuration tools, like syslog, net-snmp, and rancid. (A colleague in the formal methods community, on seeing my slides for the CoNext workshop said, "Rancid? Are you serious? Any field that respects itself wouldn't have a tool called *Rancid*!") In addition, we have bunch of open-source platforms for security (e.g., bro and snort), routing (e.g., Xorp, Quagga, and Bird), and packet processing (e.g., Click, OpenFlow, and NetFPGA). These are all good things, but they are not the whole story.

Another colleague of mine once told me, "You networking people are very curious. You really love your artifacts." I suppose this really puts the nail on the head. My interpretation of this comment is that we often emphasize the practical implications of our research as much (if not more) than the intellectual depth. We often stop short of crystallizing the intellectual "nugget" of our research, even when the work is a true gem. Another (closely related) interpretation is that we devote a great deal of effort into backwards compatibility with the existing Internet, so we can incrementally deploy our solutions. Networking is a somewhat strange area of systems research, where we don't really have control over the whole system, so we are forced to clarify exactly where and when our new ideas "fit in."

Anyway, my colleague went on to say, "You know, that wasn't a very nice thing for me to say. I do think it is true. But perhaps it is just a phase you are going through." I thought this was an interesting point, too. I think this means that, right now, much our research takes place in the context of a huge, successful artifact—the Internet—that we want to better understand and improve, before we can fully devote ourselves to foundational research on how to design and analyze large, federated networks. Actually, I think we *do* really love our artifacts, so there's an element of truth here, but I think perhaps there's more to it. I don't think networking is "just the (arti)facts."

3. INTERDISCIPLINARY RESEARCH

I sometimes think that perhaps networking is not a scholarly discipline at all, but rather (just) a rich problem domain for other disciplines. And perhaps that's okay. Another colleague of mine (in theoretical computer science) once enthusiastically approached me in the halls of AT&T Research (where I worked at the time) to ask me, "What are the top ten classic problems in networking. I would like to solve one of them and submit a paper to SIGCOMM." I looked at him quizzically, and then replied, "We don't work that way. I don't have such a list." He said, "Then how do you consider networking a discipline." I said, "Well, perhaps networking is not a discipline, then." At least not by that definition.

Of course, networking really *is* a rich domain of interesting and practical problems, and theoretical techniques from more-established disciplines have contributed immensely to our field. Just to give a few examples:

- **Algorithms and data structures:** Over the years, we've seen a wealth of influential research on streaming algorithms for packet processing (e.g., longest-prefix match, multidimensional packet classification, and traffic measurement) and distributed algorithms for computing paths (e.g., Bellman-Ford, spanning tree, and distributed hash tables).

- **Control theory:** The analysis of TCP dynamics and load-sensitive routing has relied on control theory to understand how hosts or routers adapt based on delayed feedback about network conditions.

- **Queuing theory:** Kleinrock's early work applied queuing theory to demonstrate how packet switching can achieve significantly better efficiency than circuit switching for data traffic, through statistical multiplexing.

- **Optimization theory:** Many researchers apply optimization techniques to compute the tunable parameters in distributed protocols (e.g., tuning OSPF/IS-IS weights to perform traffic engineering). More recently, researchers also use optimization theory to design and analyze protocols, such as TCP congestion control and network load balancing.

- **Game theory and mechanism design:** In recent years, game theory has given us a rigorous way to reason about the influence of economic incentives on protocol behavior (e.g., stability and incentive-compatibility issues in interdomain routing). Game theory is also used to study ways to incentivize sharing of resources (such as CPU, bandwidth, and power) in mobile adhoc networks.

- **Formal methods:** The formal methods community has provided automated ways for us to design and verify protocols, and find subtle bugs in existing protocols, using tools like model checkers and theorem provers.

- **Information theory:** Research on network tomography allows us to infer network properties that cannot be directly observed from limited measurements (e.g., inferring traffic matrices from link loads, or link performance from path-level measurements).

- **Cryptography:** Cryptography provides important building blocks for designing secure protocols and mechanisms. In addition, theoretical cryptography is a powerful tool for designing and analyzing secure protocols, and proving negative results on the security machinery necessary to achieve a particular goal.

- **Programming languages:** New programming languages can raise the level of abstraction for programming and configuring networks, leading to fewer bugs, vulnerabilities, and configuration errors. For example, declarative languages nicely separate the user's intent from the underlying (distributed) operation of the system.

- **Graph theory:** At the most basic level, a computer network is a graph consisting of network elements and the links interconnecting them. Graph algorithms are widely used to compute paths and analyze measurements of network topologies. Graph theory also provides valuable metrics we can use to capture the properties of network topologies.

Frankly, even if networking were *only* an interesting application domain for these (and other) bodies of theory, we'd have an unending supply of fascinating and important research to do. I strongly believe that great research can happen when you bring a rigorous body of techniques (i.e., a "hammer") to a faithful representation of practical problem (i.e., a "nail"). This kind of research is surprisingly difficult to do well, but so valuable when you do. I often recommend graduate students in networking make a serious effort to become sufficiently conversant in one body of theory so they have their own hammer to wield on the many nails that networking presents. Or, foster a close collaboration with a specialist in one of these disciplines, and work together to have the kind of impact that neither of you could ever have alone. Patience is required, but the dividends are great.

In addition, networking is not only an application domain for the theoretical disciplines, but also for other systems fields. Perhaps the most compelling example is *distributed systems*. The components in a distributed system communicate over networks, and network protocols are themselves a special kind of distributed system. The two fields evolved in parallel, with far too little influence on each other. Yet, they explore similar issues, and it is fruitful to rethink many networking problems (such as distributed management of routing-protocol state) through the lens of distributed systems. The success of conferences like NSDI (Networked Systems Design and Implementation) is a good indication of the healthy synergy between distributed systems and networking. Other areas like operating systems and computer architecture are clearly relevant to thinking about improving the performance, reliability, and functionality of the end-host network stack, network interface cards, and router/switch design. And, software engineering (similar to programming languages) is crucial to helping us understand our design requirements for the Internet, and how to create solutions that are more modular and easier to reason about.

Another take on the "networking as a domain and not a discipline" is that, in the words of yet another colleague of mine, "networking is an *opportunistic* discipline." I think he meant it in a nice way, like we have our fingers on the pulse of the pressing practical problems—i.e., that we value "good taste" in problem selection. There's an element of truth in that. Here's a recipe to write a networking paper, or create a networking start-up company. First, identify an unmet (and perhaps previously unarticulated) need or capability, and then invent a feature or system that meets that need. Next, determine how your solution fits in the existing infrastructure, and build and/or evaluate your solution to demonstrate how it improves the world. Finally, pitch the problem and your solution to a program committee (or a venture capitalist), and either bask in glory or lick your wounds. Something like that.

To end this section on whether networking is a domain or a discipline, I'd like to mention a few quotations from colleagues in other fields, since I find them so illuminating. One colleague said, "Networking papers are strange. They have a lot of text." I think that's often true. We do spend a lot of real estate in our papers motivating the problem we've identified, and justifying our approach to solving the problem. As another colleague of mine said, "You *need* text to explain an ill-defined problem, and the approach you are taking to solve it. Only a very well-defined problem can be presented with less text." Also, we *need* to justify our choice of research problems, since we don't have community agreement on a "top ten" list of classic problems. Still, we may sometimes go to far—sometimes we so value the choice of problem and general approach to solving it, that we don't pay sufficiently careful attention to the specific theorems, analysis, or experiments that validate our solution. The best research introduces an important new problem *and* a rigorous and well-evaluated solution.

Finally, the last quotation is "So, these networking research people today aren't doing theory, and yet they aren't the people who brought us the Internet. What exactly *are* they doing?" Now, that's an interesting question. Admittedly every research project can't be an "out of the park" home run like the Internet was—and even that research project didn't enjoy tremendous success overnight. But it's a good question to ask: what *are* we doing?

4. TEACHING NETWORKING

Perhaps we should look to how we *teach* networking for a way to define our field and its key intellectual underpinnings. So, how do *practitioners* learn networking? Walk into any major bookstore, and you'll see row after row of thick books to prepare students for certification exams. Cisco certification, Juniper certification, Microsoft certification, and so on. Don't get me wrong. These books, and the accompanying courses, are important for training practitioners to configure individual pieces of equipment. But they do not point to a principled way to design network protocols or reason about the complex ways different components and protocols might interact. The other way that practitioners learn networking in "on the job training"—a polite term for "trial by fire." Far too much learning of networking happens this way, through "case studies" of Things Gone Horribly Wrong.

Perhaps colleges and universities do better? Well, yes and no. Most undergraduate networking courses teach "how the Internet works." I'm as guilty as anyone—that's what I know, and it's what the students want (and need) to know. And it is what the textbooks mostly cover. Graduate networking courses typically cover the "20 best papers"—which really means a handful of classic papers in networking coupled with the professor's favorite papers, with a healthy representation of the professor's own research. (Sigh, I'm guilty of that one, too.) Brighten Godfrey, a networking professor at UIUC, sent me this great quotation from John Day:

> There is a tendency in our field to believe that everything we currently use is a paragon of engineering, rather than a snapshot of our understanding at the time. We build great myths of spin about how what we have done is the only way to do it, to the point that our universities now teach the flaws to students (and professors and textbook authors) who don't know any better.

Now, I'm not quite as cynical as all that. I think we *do* have several great networking textbooks that capture our current understanding, and we are truly fortunate that several of the leading researchers in our field have devoted serious time to writing textbooks. I think the bigger point to take away from Day's comment is something about the relative immaturity of our *field*, that we don't have a clearer intellectual scaffolding for how to design and analyze protocols and networks. We can't teach something we don't (yet) know.

I'm excited about the resurgence of educational initiatives in SIGCOMM, as evidenced by the appointment of an Education Director (Olivier Bonaventure) and a workshop on networking education at SIGCOMM'11 in Toronto. I do believe that meaningful discussions of what we should teach, both at the undergraduate and graduate level, can go a long way toward helping us define the networking discipline.

5. WHY NETWORKING IS SO COOL

Now, after all this ranting, you may wonder why I am consider myself a networking researcher, and why I haven't switched to (say) computational biology or machine learning. Well, for me, many of the critiques of our field—by me and by others—have a positive side to them. In particular:

Networking is tangible: Networking, as a field, is very close to reality. We can measure and build real things. (Yes, we really *do* "love our artifacts," thank you very much!) And, through our work, we can effect truly far-reaching change in the real world, perhaps even during our own lifetimes. I really do believe that.

Networking is inherent: Networking is an important part of almost every practical problem. Hardly any real-world system, functions without some form of communication. The "smart grid," financial markets, and automobiles are three prominent examples. A networking researcher can essentially select any important problem domain and find interesting computer networking challenges there. And, networking researchers can easily shift from one problem domain to another throughout their careers, while still leveraging a common base of tools and techniques.

Inherently interdisciplinary: Networking *is* an application domain for many other disciplines. We have interesting, well-motivated problems that can yield to rigorous solution techniques. Beyond the theoretical and systems disciplines I mentioned earlier, networking also has a fascinating interplay with public policy, economics, and social science. How cool is that!

Widely-read papers: Many of the most cited papers in computer science are from our community—papers on congestion control, distributed hash tables, resource reservation, self-similar traffic, and multimedia protocols. Three of the top-ten cited authors in computer science are from our field (namely, Scott Shenker, Van Jacobson, and Sally Floyd). So, *somebody* is interested in reading this stuff. Now, perhaps we're all just part of a large cult where we read each other's papers, or perhaps our papers have longer bibliographies than in other fields, but I think there is more to it than that.

Young, relatively immature field: Our field is a work-in-progress. This is great if you like to make order out of chaos. Tremendous intellectual progress and practical impact are still desperately needed. *You* get to help decide what networking really is. That's a rare and wondrous thing.

Problem selection and taste: Defining the problem is as big a part of the challenge as solving it. Networking researcher thrive on identifying previously-unknown problems, or precisely formulating and solving an existing problem. That's hugely exciting.

Platforms for building your ideas: Because we so value our artifacts, we create a lot of them, allowing other researchers to build on top of them. We have programmable data planes, like Click, OpenFlow, and NetFPGA. We have open-source routing software, like Quagga, Xorp, and Bird. We have many invaluable testbeds, like Emulab, PlanetLab, and Orbit, just to name a few. And we have publicly-available measurement data from RouteViews, CAIDA, and Internet2, and can collect our own data by running a wealth of different measurement tools. This substantially lowers the barrier to doing influential systems research in networking.

6. INTRA-DISCIPLINARY QUESTIONS

So, networking is cool, but that doesn't address the question of what networking research really *is*. The short answer is that I don't really know. But I do think that our field is defined more by the *questions we ask* than the *techniques we use*—the heart goes before the brain. Clearly our field is interdisciplinary, and we apply a wide array of techniques to networking problems. And clearly the questions we ask change often over time, as new trends in applications or technologies alter the networking landscape. (Think how many papers we see on data-center networks and energy-efficient networks now, compared to a few years ago.) And researchers can have exciting careers solving networking problems in other domains, or applying techniques from other disciplines to solve key networking problems.

All of that is good, and arguably plenty to keep us busy as a field. Still, it begs the question of whether there are *any* research questions that are fundamentally *inside* the networking discipline—intradisciplinary research questions. Here, I believe there are overarching questions that we come to again and again—they just aren't the kind of sharply stated research problems that would please my AT&T colleague who wanted a "top ten" list of questions to answer for a killer SIGCOMM paper. These questions relate to the *definition and placement of function*—what the network should do, and how to divide the functionality between the components. How should we split functionality between the end host, the network elements, and the systems that manage them? Is the right split for traffic management to have end hosts do congestion control, network elements run routing protocols, and management systems optimize the tunable parameters in these protocols? Or something else? Similarly, how should we divide functionality across the many concurrent protocols and mechanisms that run on the many components in the network? These are all questions of *network architecture*, a term I'm hesitant to use since it always sounds vague and wishy-washy to me.

6.1 Today's "Divisions of Labor"

So, what are the divisions of labor that define today's Internet? These are the "motherhood and apple pie" of networking:

Between the end host and the network: End hosts (or, really, their points of attachment) have IP addresses and send packets over a network that offers a best-effort packet-delivery service.

Between the protocol layers: The classic "hourglass" of Internet protocols has the IP network layer smack in the middle, riding over many unforeseen link-layer technologies and supporting multiple transport protocols (like TCP and UDP) that support a wealth of unforeseen applications.

Between the data, control and management planes: Though not an original part of the Internet architecture, today's network infrastructure has its own division of labor. The data plane performs streaming algorithms on packets, the control plane responds to events (such as topology changes) by computing new paths, and the management plane applies policies to configure the network elements to influence how the control and data planes behave.

Between administrative domains: The Internet is, by definition, a "network of networks." The Internet is divided into independently administered networks that coordinate in a cooperative and competitive fashion to compute paths for delivering traffic. Today, this is done using policy-based path-vector routing.

The last two items on this list evolved over time as the Internet grew larger, faster, and more commercially-oriented, but they arguably play as big a role in the Internet of today as the first two more "classic" divisions of labor do.

6.2 Toward New "Divisions of Labor"

So, what about the future? I would argue that quite a bit of today's networking research is, in one way or another, looking forward to new divisions of labor that address needs and trends that did not exist in the early days of the Internet. These include:

The host vs. the network: The Internet is increasingly a platform for accessing online services—Web search, social networks, virtual worlds, and so on—replicated on many servers running in data centers. The service is more important, and more permanent, than the computers hosting specific instances of the service. Servers fail, and go down for maintenance or to save energy. Virtual machines move, by migrating from one physical server to another. Clients move, either from one location to another, or from one network interface to another (e.g., from 3G to WiFi). This is leading us to revisit everything from naming and addressing to the end-host network stack and the socket API. In addition, within a data center the same company has control over both the end host and the network, leading to a wealth of exciting research that moves key networking functionality to the servers, for better scalability and more flexibility.

The network vs. the management systems: During the past several years, networking researchers have explored the appropriate division of labor between the network elements and the systems that manage them. Should we have a smart management system coupled with a dumb network? This is the view of many research projects, including the successful OpenFlow initiative. Moving most of the "smarts" to the management system enables greater programmability and the benefit of controlling the network based on a network-wide view of the topology, traffic, and performance. On the other extreme, other research proposes a dumb management system and a smart network, where the network elements run distributed protocols that adapt automatically to changing conditions. A classic example is the resurgence in interest in load-sensitive routing, where the network performs its own traffic engineering by adaptively splitting traffic over multiple paths in response to measurements of link load or path performance. This allows faster adaptation to changing conditions, without the intervention of a separate management system.

Between administrative domains: Increasingly, we see a tension between the need for *decentralized* control of the Internet (where each AS can make routing decisions based on local policies) and the desire to see the Internet as single, *global* network (with desirable properties like stability, scalability, reliability, security, and manageability). Where is the right place for each of the stake-holders to "weigh in" with its local objectives, and how do we ensure the system as a whole behaves well? Today's Border Gateway Protocol (BGP) is one point in the design space, where the choice of path-vector protocol constrains some behavior but leaves plenty of opportunity for ASes to (collectively) shoot themselves in the foot. But, is this the right "division of labor"? Probably not. Much research has gone into analyzing the behavior of today's BGP, and looking forward to alternative ways to slice this cake.

Much of the research we do, and the papers we write, fall into categories like these. Though it may not be apparent to outsiders, we are all part of a larger discourse on these topics. I think this is part of why our main conferences are lively and well-attended. We do have a collective stake in each other's research, because it is part of a larger push to answer bigger, overarching questions about what functionality networks should offer, and how to best "slice and dice" that functionality across the various components.

7. CONCLUSION

So, did I recover from my "summer of angst"? Well, I came away with a rekindled belief that networking is an awesome field. We have real, important problems. We have tremendous opportunities for impact. And we have great fodder for interdisciplinary research. That's all good.

But, I do believe that our field is intellectually immature. We are (still) more of a "domain" than a "discipline," and are still searching for our intellectual center. Maybe that's okay, and our field is like the fictional character Peter Pan who proudly refuses to grow up. Our field is all about change, as the assumptions, requirements, and technologies underlying our field forever seem to shift under our feet. But, while we continue to embrace change, I hope that we can make the questions we ask more precise, and the way we answer them more rigorous, so we can put networking (both the field, and its artifacts) on a stronger foundation.

How do we do this? I think part of the answer lies in deep *interdisciplinary* work—becoming conversant in another discipline (whether a theoretical discipline or a systems discipline), or collaborating closely with a specialist in one of those fields, and applying that discipline to core problems in networking. I think another part of the answer lies in greater *patience* and *perseverance*. We have an understandable tendency to jump from one research problem to the next, as application, technology, and business trends shift. I fear we err too far on the side of valuing new problems over deeper answers to existing questions. We need to fight this urge, to encourage more thorough, complete, and deeper research that truly helps the field "grow up," without losing its child-like sense of wonder.

Acknowledgments

Thanks to Hari Balakrishnan, Mung Chiang, Nick Feamster, Pamela Zave, and the students in my research group for comments on earlier versions of the paper. Thanks also to the students in my graduate networking class and the CoNext'10 workshop attendees for valuable discussions.