Forty Data Communications Research Questions

Craig Partridge
Raytheon BBN Technologies
craig@bbn.com

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 It has NOT been peer reviewed.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

About ten years ago, Bob Lucky asked me for a list of open research questions in networking. I didn't have a ready list and reacted it would be good to have one. This essay is my (long-belated) reply.

Categories and Subject Descriptors

C.2 [**Computer-Communications Networks**]: General

General Terms

Algorithms, Measurement, Design, Economics, Experimentation, Security, Theory.

Keywords

Research, research agenda, data communications, data networking.

1. INTRODUCTION

"Are there any research questions left in networking?" Bob Lucky asked me that question in 2003 when I spoke to a National Research Council committee he chaired[40]. As I recall, I answered Bob's question with a handful of research questions that I knew were popular at the moment. I know I walked away from the discussion feeling that a more thoughtful list would be useful.

2. MAKING A LIST

There were two major motivations behind this list.

First, Dave Clark has periodically observed that other fields sometimes find it useful to create a list of open research problems. For example, much mathematical research in the 20th century was motivated by Hilbert's list of 23 questions[24]. While this list does not aspire to that level of influence, it was clear that creating a list would be a useful intellectual exercise.

Second, I felt that the tremendous success of the Internet had paradoxically limited the research community's vision. We are so often challenged to ensure the Internet's health (e.g. better security or traffic management) and future (e.g. future Internet research), that we often lose sight of the wider range of research challenges. That's a loss, as pursuing less pressing problems sometimes yields results that can help with immediate needs. So I wanted a list that offered an expansive view of the field.

2.1 Defining "Interesting Research Question"

Much of the work in this paper is simply the result of looking for interesting research questions. But that begs the question of what is an interesting research question? With some guidance from the work of Hilbert and Heilmeier[23] and Crowcroft[13], to be interesting a research question had to fulfill the following requirements:

- *Worth the attention of multiple (new) researchers*. The question had to be rich enough in scope that if multiple distinct researchers worked on it, the research community would likely find benefit in the multiplicity of research results. This requirement meant I excluded problems that already had several people working on it. However, some problems with only one research team pursuing them (no matter how good the team) are listed.

- *An answer should open up substantial follow on efforts (either in research or industry or both).* Newton famously said, in a pithy variant of Bernard of Chartes' epigram, "If I have seen a little further it is by standing on the shoulders of Giants." The goal here is similar – to find research questions of sufficient height that solving them (or showing they cannot be solved) gives others greater visibility into new problems.

- *Likely to reward attention.* I needed some reason to believe that if someone chose to pursue a question, they had a reasonable chance of success. There are important problems in our field on which, for good intellectual reasons, it is near impossible to make progress (i.e. all-optical regeneration[6] and several problems in secure systems). I felt it was disingenuous and might lead someone to waste valuable research energy to list such questions here. In a similar vein, Crowcroft has provided guidance for identifying "cold" research topics.

- *Some chance the result will have an impact*. Impact is often the first thing people consider in choosing a research question or research agenda. Heilmeier trenchantly asked, "Who cares?"

  But impact is a risky metric. First, researchers in all fields are notoriously bad at assessing the potential impact of their work. Second, impact may arrive many years after the result. To take an example from economics, the work of Arthur Pigou on environmental effects only became popular fifty years later[36]. Finally, there's a tendency to pound away at problem perceived as high impact, even if prior work strongly suggests that working on the problem is a waste of time.

  So while it is important to do work that matters, impact only mattered once I'd confirmed the problem was likely to reward attention.

2.2 Vetting Questions

Given a question that met the requirements, I then took two steps to validate the question.

- *Ensure the question was clearly articulated*. It was often hard to refine a problem into a clear question, but the effort to clarify was rewarding. Sometimes, once clarified, the question turned out to be uninteresting.[1]
- *Confirm the question was open and understudied.* I researched the question (conference proceedings, conversations with experts, Google) to see if the problem it expressed really was an open problem and wasn't getting much attention. This process killed several questions. For instance, a search led me to the PQCrypto conference and forced me to remove some questions related to security architectures in a world where quantum computers are prevalent.

3. NECESSARY CAVEATS

Producing this list has been a reminder of how little I know. In that spirit this section offers two warnings:

- While one of the standards for inclusion in this list are that the problem is unsolved, I make no promises – if you decide to make one of these ideas into your doctoral dissertation or the centerpiece of your research program for the next few years, please do a literature search to make sure someone else hasn't solved the problem.
- This list is in no way comprehensive or representative in the sense of reflecting a proper balance of different subareas of data communications. This list reflects one person's perspective and ignorance.

4. THE RESEARCH QUESTIONS

Originally this list was unstructured but multiple readers commented that reading a long unstructured list reduced its impact. Initially I tried sorting the list into categories (wireless, security, peer-to-peer) but using categories had two deficiencies: (1) it created pressure to create questions, even if the particular category had no good research problems at the moment (a problem that happens more often than we like to admit) and (2) readers tended to jump to the section on "their area" whereas a goal of this paper to encourage readers to think expansively.

So I ended up structuring the list around classes of problems. The notion is that research questions come in different styles. For instance, the goal may be to break through a known barrier, or we may have a result that turns our understanding of the world upside down, but it is a point solution and we need to understand the broader perspective it provides.

4.1 Getting past obstacles

A common research problem is that we can foresee a rich and beneficial future for society or for research if only we can solve a particularly difficult problem that blocks the path to that future. Examples of solved obstacles in other fields include Little's Law[39], the Taniyama-Shimura conjecture[66] and Harrison's chronometer[67].

One challenge in confronting obstacles is that, sometimes, they have proven to be remarkably resistant to being solved (and resistant to being proven unsolvable). At some point they stop rewarding attention and instead simply consume resources. I have left such thorny obstacles out of this list.

1. *How to handle parallelism inside routers and hubs*? Periodically, technology evolution forces us to re-examine the innards of connection devices.[2] Today we are (once again!) pressed by the issue of parallelism: we are achieving greater computational power using more processors in parallel (multicore) rather than faster single processors[57]. We may have to rethink the innards of our devices to adapt to this new form of parallelism. Three lines of thinking have emerged: running multiple independent stacks in parallel [70], running exactly the same software on each processor[34], and using co-processors. There's room for more thinking and a deeper examination of details (the devil of these problems is often in the details – big enticing visions can prove to have feet of clay).

   Discussion: Network engineers and researchers often struggle to maximize hardware to keep up with demands for better performance. Whenever we have a paradigm shift in how devices get faster, it takes time and thoughtful research to determine what approach is most effective. The right approach will drive how we build high-performance network devices for years to come.

2. *The Parallel-to-Serial (and back) problem*. An abandoned problem from the 1990s returns. The problem is how to connect a multi-processor (multicore) end system to a (serial) network. The problem is that the challenge of dividing arriving data across the processors (and or marshaling data from the processors to send) tends to scale poorly. More formally, the problem is given *n* processors, $n>1$, where each processor π is responsible for a variably-sized piece of data d_π from $d_1 \ldots d_n$ of a packet, *D*, we want the cost of assembling and disassembling *D* to scale sub-linearly with respect to *n*, but past history suggests assembly/disassembly scales exponentially, due to coordination required among the *n* processors to manage the receipt or delivery of *D* on the serial channel. When we looked at this problem in 1990s (e.g. in Thinking Machines), we thought the assembly/disassembly had to be done by a processor. Since then we've gotten better at special hardware and smart memories. So, for instance, on the transmission side we could (perhaps) imagine a memory that notified a processor when it had gotten data from all *n* contributors and did a scatter-gather assembly (with checksum) of the outbound packet. If we could find a way to variably split data on inbound side we might have a solution.

   Discussion: This problem is one of the Achilles heels of networking. We struggle to get data efficiently in and out of computers with parallel processors. A solution will smooth the path to parallelism and parallelism is clearly our future.

[1] On the importance of clear research questions see "Bad Career Move #2: Let Complexity be your guide" in [56]

[2] Consider the rapid change in router design between about 1988 and 1996. We went from single processor with single bus, to single processor attached to multiple busses (with intrabus transfers supported), to distributed processing using forwarding engines distributed around a parallel, switched backplane.

If there is no general solution, we will have to think more seriously about constraining our data packing in packets to meet parallelism's requirements and that's architecturally unpleasant as we would be enshrining a edge-system limitation inside our network.

3. *Networking processor cores*. The previous two questions observed that multicore processors are likely to be important in our future. Interestingly, the way that cores are being interconnected on a chip increasingly looks like computer networks and chips are suffering familiar networking problems such as congestion. What makes these networks of interest is that source behavior is sharply different– unlike Internet sources which may continue to transmit into congestion, these sources rapidly self throttle for organic reasons (e.g. they cannot page in the code they need).

   Discussion: There's a clear impact on how multicore chips are designed and the rate at which they effectively process. We may also find it has bearing on how data moves from exterior networks into cores. Initially I thought this problem was too simple – a matter of tweaking Nick Maxemchuk's work (e.g. [45]). But the authors of a paper at ACM HOTNETS convinced me differently[52].

4. *How to identify tussle spaces?* As defined by Clark, Wroclawski and Sollins[9], "tussles spaces" are portions of the networking milieu where adversarial stakeholders vie to achieve their respective interests. These tussles can spill over to affect other parts of the networks. More strikingly, many of these tussles are acted out in technical details of how network(s) are realized. In this light, [9] advised:

   "Anyone who designs a new enhancement for the Inter- net should analyze the tussles that it will trigger, and the tussles in the surrounding context, and consider how they can be managed to ensure that the enhancement succeeds."

   Fair advice, but currently almost impossible to achieve as we lack clear rules for identifying tussle spaces and for forecasting how tussles might spill into surrounding contexts.

   Discussion: The Internet's creators were lucky that the Internet's architecture was robust enough to survive its evolution into a central part of the world's economy. Tussle spaces are a recognition that we cannot trust in luck again, but are also a crippling challenge – a network architect can feel irresponsible if she does not think carefully about tussle issues. Yet, as a field, we have yet to give her effective tools to drive her thinking.

5. *What are the incentives for an implementation to faithfully follow the protocol specification?* Shneidman and Parkes [65] asked this intriguing question several years ago and while they made substantial headway it remains a potent challenge. The core idea is assume a rational network node and a protocol specification that is correct (there's no motivation to "fix" the protocol), but that there may be benefit to the node (perhaps at the expense of other nodes) in deviating from the specification. How might we devise protocols so the motivations align with correct operation? Or do we need points in the network that test for compliance?

   Discussion: This question is a constrained version of the tussle question (4) and perhaps more tractable.

6. *Create the cognitive radio*! The idea of a cognitive radio is now nearly 20 years old [47]. It is a radio that senses its environment and figures out how best to utilize the spectrum to meet its applications' needs[2]. The concept is wonderful and offers a clear path to better use of our RF spectrum. But, nearly 20 years later, we have not yet implemented one! One reason is that we have only recently created sufficiently programmable radio platforms (e.g. the ones mentioned in question 38). Another reason, however, is that creating a cognitive radio is *hard*. Network engineers do not intuitively structure their systems for cognitive control. Similarly, the AI community struggles with the notion of distributed, asynchronous cognitive control with imperfect information ("asynchronous" and "imperfect information" being most problematic)[20].

   Discussion: Cognitive control is an excellent way to manage and make sense of environments (such as networks or some wireless devices) with hundreds or thousands of configuration choices. As a result, cognitive systems seem likely to become increasingly important in networking. A cognitive radio is an excellent test case for this belief as it combines hard problems in cognition with potentially large (and measurable) benefits in network performance.

7. *Platform-independent link and media-access specification language?* The challenge is the following: when a software radio arrives in a new location (imagine encountering the first base station in a new country) we would like to tell the radio the protocol(s) it should use. Because this is a world of software radios, where we expect a profusion of protocols and rapid innovation, it is possible that the protocol in use is one the radio has never encountered before. So we would like a specification language that the radio can compile/interpret/reduce such that the radio can, in real-time, begin using the protocol(s). There's a wide range of choices about how one might solve this problem, ranging from a virtual machine model (something akin to Java for a radio) to a formal specification language that can be compiled by each radio to work on the radio's specific hardware. It is a difficult problem that requires a good understanding of what is possible in programming languages as well as a good understanding of RF physics, media access and the range of choices in how to implement a software radio.

   Discussion: This problem comes originally from the field of software radios but is now also relevant to optical networks (see question 33). It has received almost no attention – probably because the most prominent software radio system (JTRS) assumes that protocols are well known and registered in advance. Given the diverse research spaces (languages, RF physics, radio implementation) touched by this problem, even partial solutions are likely to yield important insights into the challenges of programming radios. For an example of the kind of impact that trying to program physics can yield, look at the Claytronics program at CMU and its programming languages, MELD[3] and LDP[17].

8. *Accurate wireless simulation*? The wireless field has a serious problem. Wireless researchers do not trust their analytical models and do not trust their simulators. We desperately need an open simulation platform, whose results have been field confirmed, that researchers can use and whose results will be accepted. Imagine a simulator that

accurately traces how several dozen concurrent signals propagate over a terrain (no unit circles!) and accurately estimates the combined signal each receiver experiences. There is confidence that such a simulator should be possible – we have been doing it with emulators (which run real radio software over an accurate RF emulator) for several years. Yet somehow we have not made the jump to simulation.

Discussion: The lack of a simulator has led to a poisonous situation in wireless research where surprising results are often dismissed because either (a) the handful of radios a researcher can afford is "too few for a useful result"; or (b) "the simulation results cannot be trusted." Finding the right simulation paradigm and showing it works would get the field past this painful dilemma.

9. *How much bandwidth can we utilize with ambitious wireless spectrum sharing and reuse?* We know that the wireless spectrum is woefully underutilized[42]. How can we exploit the unused and underused spectrum effectively? There are various suggestions for modest amounts of highly constrained reuse. The goal here is to envision a broad range of sharing – in the extreme, assume a radio that can tune over the entire RF spectrum and can change its MAC, power levels, directionality and coding. What could you achieve? Observe that one can go at this problem multiple ways. One could measure the spectrum in some locations and then compute, using different sharing approaches, how much capacity could be used (expressing the result as a bit rate would likely be most effective). Or one could implement a radio that is ambitious in its spectrum sharing and demonstrate sharing in the real world.

   Discussion: This problem is deceptive. It appears, on its face, to be a straightforward, if challenging, problem of measurement and implementation. At its heart, however, is a difficult question: what is the optimal strategy for a radio under a given set of RF conditions and traffic demands? Determining an optimal strategy (given many possible criteria for optimization including robustness of connectivity, bit rate, and fairness of sharing) and then determining how to implement and measure for it is a problem almost completely unexplored. The only results I've seen so far are a recent small study that highlights just how much we can learn[35] and an overview of possible strategies[77].

10. *Shared security for slices?* There is a network virtualization architecture that posits we can run multiple communications protocols in parallel in a network[70]. Each suite of protocols gets its own fraction, or *slice*, of the various pieces of network hardware such as routers and fiber optic links. This concept is enabling the creation of GENI and is also being pursued by several leading researchers. One of the challenges for slices is the security model. It is entirely possible that distributed applications may participate in multiple slices concurrently, which suggests that each application is only as secure as the least secure slice it participates in. A possible solution is to devise a security architecture that works across multiple slices. A cross-protocol, cross-network security architecture if you will. This idea is less quixotic than it sounds. Having watched security systems be devised for diverse protocols, my experience is that they share a common set of features.

    Discussion: In the short-term, this result would be tremendously valuable to GENI and others experimenting with sliced architectures. The long-term impact is likely to be learning how to map security concepts consistently across the disparate technologies and protocols.

11. *How should we compare network topologies?* This question has many forms. For instance, given two topologies of similar size (edges and nodes), how similar or different are they? Similarly, given two topologies of different size, how similar are they?

    Discussion: These kinds of questions arise in many situations, including sampling (is this part of a network we sampled representative of the whole network?), straightforward comparisons (if this algorithm works well on this network, will it work well on that network?) and cross-field studies (e.g. how similar is this comms network to that biological network?). Having a common metric, or two or three metrics, each imperfect but at least ones we understand, would be a valuable contribution. For a starting point, see[73].

4.2 Opening new doors

A feature of obstacles is that we can see (or think we can see) what the future will look like if the problem is solved. Many research problems, however, do not offer a clear vision of the future they enable until they are solved. Metaphorically we have to open the door (solve the problem) to truly understand what is on the other side. An example is the discovery of self-similarity of network traffic[38]. Until that work came out we did not know if we simply needed to tweak traffic models or whether there was something more serious that caused traffic predictions and real network traffic to diverge.

12. *Topology over time?* Consider the following abstraction of a computer network: it is a varying graph, a collection of nodes that are interconnected by some set of arcs and allow the arcs to change over time (e.g. some arcs may be removed, others may be added, as time goes on). Can we say anything useful about how the graph evolves? For instance, can we classify certain types of changes as retaining connectivity and others as disturbing connectivity, either in the entire graph or in sub-graphs?

    Discussion: One can make this problem relatively tractable (classification based on what we observe in practice) or very difficult (classification of three dimensional manifolds [e.g. time + 2 dimensions] is an incompletely solved problem in topology). While expressing the problem in terms of manifolds may make the problem sound esoteric, it is actually highly practical. Solutions could influence how routing protocols cluster portions of the network (e.g. create clusters that are maximally more robust to link outages) or give us greater insights into when "graceful restarts" of network devices are appropriate and feasible (cf. [49] but also imagine how one might engineer graceful restarts for route servers such as [7]).

13. *Distributed quantum computing*. Researchers are putting the finishing touches on quantum repeaters – devices that enable the transmission of quantum state over long distances[71][72]. This development raises the possibility that if and when quantum computers appear, we could have distributed

quantum systems. To my knowledge, no one has looked at the question of what kinds of problems might best be solved with a distributed quantum system. How might we assign the problems to the various quantum computers? What is the quantum equivalent of mapreduce[16]? Do we prefer hundreds of identical quantum computers or do we want collections of diverse quantum computers, each tuned for different types of quantum computations? How do we move information between machines and aggregate and refine results?

Discussion: Assuming quantum computers happen, research on this problem will enable us to create quantum computing clusters and generally do distributed quantum computing. Whether quantum computing will be close enough to standard computing that distributed quantum computing is desirable or useful is, of course, an open question and likely one of the answers generated by pursuing this research question.

14. Ar*e there any new addressing paradigms beyond the four we know (unicast, multicast, broadcast, anycast)?* This may seem like a quixotic question. There's an impulse to assert these four are the complete set but it is worth remembering that we started 40 years ago with just unicast and broadcast and found multicast[3] and anycast[53] along the way.

    Discussion: If we find new addressing paradigms, we will likely find a beneficial use for them. The initial reaction to both multicast and anycast was that they were unneeded optimizations and yet both have important roles in today's network. If there are no other paradigms, we will have more firmly bounded the scope of routing and addressing problems.

15. *Get rid of unicast addressing*? This problem has been a popular conversation over beer at a pub for years, but I've yet to see a rigorous approach. The central idea is that many of our networks are intrinsically multicast (wireless, free space optical, and coax plants such as classic Ethernet and some cable installations) and many other technologies could be multicast (fiber). In a world of unicast addressing, to paraphrase Van Jacobson, an intrinsically multicast medium means that when you transmit a packet on a network, 100 nodes receive the packet, and 99 immediately drop the packet because it isn't addressed to them. We know that's wasteful (cf. network coding for one way to try to take advantage[1]). What if you radically rethought networking so that this never happened? Eliminate unicast addresses and get rid of the notion of connections between two end-points. Everything is multicast (maybe you broadcast too, or maybe you don't). What does the network look like? Is routing simpler? Can one create a valuable network that serves a wide range of applications without implicitly re-creating unicast?

    Discussion: This work is definitely "clean slate" network architecture as it could enable a new style of networking and transform how we think of routing and addressing. Or if we find ourselves forced to implicitly re-create unicast, it would tell us that unicast is central to networking (that's where pub conversations usually land).

16. *Transient network addresses*. Another fun pub conversation. Suppose that network addresses are transient and perhaps, even, flat (have no topological information in them). An application can create a new network address anytime it wants and discard it anytime. The cost of new address creation is low enough you can imagine changing it every transaction (e.g. each web page download). While in use, the address is unique (no two nodes are using the same address at the same time). In a client-server world, this abstraction is useful for clients. Can we make it work for servers too? What routing infrastructure is required for transient addresses to work?

    Discussion: The reason this topic is a fun pub conversation is that it smashes any attempt to associate an address permanently with a name or an attachment point or a node[63]. That result has both intellectual impact (we keep stumbling over assumptions of permanence) and may have security and privacy impacts (the address tells you nothing about the opposite party so many techniques such as geographic mapping or address filtering stop working and more sophisticated techniques must be used to authenticate a party). What's more, there's reason to believe this problem is tractable: several researchers have recently produced results that suggest it might be possible to have flat addresses over parts of a network[33][59].

17. *Tracing across multiple types of networks. Can we trace a piece of information as it is communicated via postal mail, a newspaper, the Internet, and social contacts?* While we like to think all information flows over the 'Net, a lot of information follows a richer set of paths, including hallway conversations, exchanges of papers, and telephone conversations in addition to packets over the 'Net. We can trace conversations within the 'Net (even if encrypted)[12] and we can trace relationships in social networks (unencrypted). It would be a triumph for network science and perhaps also for time series analysis (question 34) if we could trace information across multiple networks, possibly inferring steps we cannot directly observe.

    Discussion: This problem is, of course, of deep practical interest to the military as it describes how insurgents pass information. The challenge is finding a unified way to represent the various networks that allows measurements from each network to inform a map.

18. *Resource auctions*. People are periodically interested in figuring out if ISPs could shift resources among each other in real-time (cf. the excitement when Enron tried to create a marketplace for bandwidth). While it is possible we could have network operations centers staffed by bandwidth arbitrage experts (the Enron model) it is easier to imagine that some amount of bandwidth/connectivity/dark fiber is shifted from one network to another automatically based on programmed guidance about availability and pricing. Imagine multiple private or public coordinated peering points[43] with big optical switches through which we can connect or disconnect networks in seconds (vs. the highly negotiated connectivity in such centers now). How might we structure auctions? Would it be useful?

[3] The precise origins of multicast addressing are unclear. In was not in the original Ethernet design[46] but was in the 1980 IEEE Ethernet specification. As best I can tell, the idea was initially developed by Mockapetris and Farber in their work on the distributed computer system (DCS).

Discussion: Apart from the pragmatic benefits of being able to dynamically interconnect, I suspect there are some difficult problems in network economics and routing in this question. How do we dynamically price the value of being more closely connected? How do we ensure our networks take full advantage of the dynamic connectivity (e.g. how can we be sure that our routing protocols won't simply continue routing all traffic over previously existing paths)?

19. *Where should the network "waist" be*? A feature of the network architectures of the 1970s and 1980s (which includes the architecture of today's Internet) is a "waist" in the network stack. The waist is the central, largely unchanging, API above and below which innovation is enabled. We understand, pretty well, how to handle a waist placed at layer 3 (IP in the Internet, CLNS in OSI, the PUP layer in PUPNET, etc.). But newer architectures are proposing to move the waist up (e.g. content networks[68] and HTTP[58]) or down (e.g. virtualization architectures[70]) and we don't really understand the consequences of moving the waist. What are the tradeoffs? What problems become easier or harder to solve with the waist in a different place? A question in the same vein that I've recently been asked is whether we can create a composable transport protocol, which adds and sheds functionality as it crosses network boundaries.

    Discussion: There are many network architecture questions that we still struggle to understand. I think the success of John Day's recent book[14] reflects the fact that John's willing to express some strong perspectives grounded in his long experience in the field. The question posed is one of the central architectural problems and has many consequences in how we architect networks in the future.

20. *Is there a formal theory for combining protocol elements*? Over the years the data communications community has shown tremendous skill at dividing protocols into distinct modules. Layering[15][77][8], of course, has been the primary tool but by no means the only one[27]. What we have been far less good at is putting the elements back together. There's been a steady if modest stream of work in "stack smashing" or "cross layering" that show substantial inefficiencies or even functional errors that result from layered architectures[41] (as well as one cautionary paper showing where cross layering can go wrong[32]). Can we create a theory for how protocols should be composed that either eliminates or, at least, exposes these inefficiencies?

    Discussion: This question is closely related to questions 19, but is more tactical (how best to implement protocols) rather than architectural. This question is equally important, however, as it guides us in how to create better implementations.

21. *Is there a theory of protocol decomposition*? This question may seem odd, given that the previous question stated that, as a field, we are good at decomposing protocols into modules. That decomposition, however, is for implementation and specification. The issue here is understand when a protocol has all the functions it must have and when it has too many functions or too few. Imagine if you will, asking how many degrees of freedom a protocol requires to realize congestion control. That's a well-formed question, but we have no method for answering it.

    Discussion: Of all the questions in this list, this question is, for me, the most difficult – I have no idea how someone might start to answer it and, as a result, hesitated to include it. The question, however, is clear, the research space is completely empty and the results would become an integral part of how we devised protocols in the future. How could I leave it out?

22. *Are there cooperative protocols above the physical layer?* There's a substantial body of work showing that wireless networks can perform better if nodes help each other[51] – e.g. a node off to the side echoes what it hears from a sender in order to improve reception at the receiver. For the most part, these are physical layer improvements (perhaps coordinated with the media access layer). Is there a role for similar cooperation at higher layers? At first this may seem impractical, as most higher layer communications is point-to-point. Yet many servers often work together to deliver a web page and many servers work together to deliver an email. The step from collaboration to cooperation would seem small.

    Discussion: Like several other questions, answering this question would extend our understanding of communications and expand our palette of techniques for realizing protocols.

23. *Ensuring we send the most important piece of information in the next message*. This problem is best illustrated by two examples: (1) suppose you have received four out-of-order data segments – which three should you acknowledge in the next selective acknowledgement (SACK) to maximize throughput? And (2) what ordering of information in an HTML document will allow a browser to display the information fastest? We think we have solutions to these individual problems (see [44]for throughput and the HTML specification intentionally puts style and framing attributes near the front of the document), but the general problem remains unstudied. The essential step would seem to be to recognize not all bits (or data) are equal and crafting some way to assess the benefits of sending one set of bits vs. another. Game theory and semantic information theory seem likely starting points.

    Discussion: This is another question in how to design protocols, but its focus on what is transmitted gives it a focus that may make it more tractable.

24. *A new paradigm for network management*. There's broad agreement that today's network management paradigm is broken. I say that as someone intimately responsible for that broken paradigm. Today's network management has at least two serious deficiencies: first, it is a method for raw data gathering – gathering without understanding; and second, it is focused on delivering data to management centers, when it is now recognized that management data is of great value to individual users. For those who would worry that delivering management data to users is commercially not viable – Dave Clark's *why* button[10] suggests that giving users at least some data will reduce demands on user service organizations.

    Discussion: I hesitated to include this question. That there is a rich, unexplored, research space and ample room for good ideas is clear. The challenge is that network management, as a field, is notorious for its failure to reward talent.

25. *Tearing down network management silos*? For several decades, we have divided network management into four or five distinct silos. The current set is Operations, Administration (and Accounting), Maintenance and Provisioning. Each silo tends to have its own tools, its own databases, and its own way of doing business. This leads to situations where multiple databases contain information about what software version a device is running. This situation is especially silly as the only place that truly knows what software version is being run is the device itself. Can we develop an integrated approach that is more efficient and less redundant?

    Discussion: All the comments about network management in question 24 also apply here, with one important distinction. It is clear that network operators and equipment makers realize there's an issue here and periodically begin to offer money for research ideas. Popular wisdom is that network costs scale with the cost of paying people to manage them[60] and so everyone sees a chance to reduce costs in a new network management structure. So there's a bit less risk.

26. *Can we develop a new model for security analysis?* Currently we analyze the security of systems (and networks and protocols) using two methods. One can be described as reactive: see what the opponent does and devise some way to counter. The other is a relatively formal process descended from the U.S. government's Rainbow series of reports. Both approaches have limitations. The reactive model is not designed to lead to secure systems. At best it allows us to patch the systems we have. The formal model supports the creation of secure systems but like most formal systems it is time consuming to employ and is designed to produce a binary result: secure or not secure. It is not a system designed to allow the discussion of alternatives or risk tradeoffs. What we would like is an approach to security analysis that enables us to straightforwardly evaluate tradeoffs in a well-defined (e.g. formal) context and to do so quickly and easily.

    Discussion: It is widely accepted that we need "better" approaches to security. At the same time, there's considerable skepticism that either current approach to security analysis will help us find such a new approach. Yet there's currently optimism (perhaps misplaced) that we could devise a new analytic model, one that might allow a reasoned reframing of the question of what it means to be "secure."

4.3 Understanding a solution space

Sometimes in data communications we find ourselves with some interesting, even surprising, results that create a whole new set of questions that we must answer if we are to fully understanding the implications of the results. An example from another field is high-temperature superconductivity: up until 1986 no one believed superconductivity was possible at temperatures over 30 degrees Kelvin, then Bednorz and Müller[5] demonstrated superconductivity at higher temperatures and the field had to completely rethink the topic.

27. *How can we optimize energy use in a single-hop wireless network (e.g. a base station and edge node network such as WIFI or the cellular phone system)?* There's a general agreement in the research community that the best approach to energy efficient wireless networking is to turn off the radio when it is not in use. There have been several papers looking at how to intelligently turn the radios on and off in sensor and multi-hop ad-hoc networks[73][61]. But almost no one has looked at the problem for the most common case, a single hop environment using a base station (an exception is [30]). The problem is intriguing, especially as its solution is probably asymmetric: the base station has a power source (it is plugged in) while the clients are typically running on battery power. So a scheme that reduces energy consumption more in clients is probably desirable. At the same time, the base station's possible ability to serve as a centralized controller makes the problem easier than the distributed schemes required for ad-hoc and sensor networks. Note that WiFi has power mechanisms that can serve as a starting point for experimentation.

    Discussion: The commercial and social impacts are likely to be large, as wireless is on a trajectory to be a major consumer of battery power in devices. More efficient wireless means longer battery life for PDAs and other consumer products. Intellectually, the impact of the result on other research would seem to depend on how distinct the solutions are from the excellent results that are being achieved for multi-hop ad-hoc networks (question 29).

28. *Protocols for continuously disrupted networks*. Today's networking protocols, even those such as Disruption Tolerant Networking (DTN) protocols, all assume that much of the network is stable, where by stable I mean that links are usually up or, at least, are operating on predictable schedules. What if we upended that assumption and replaced it with the assumption that most (all?) of the network was composed of links that were often down and whose periods of "good" operation are highly variable (in particular, can be quite short). How would we optimize data communications protocols for such an environment?

    Discussion: This question is intended to be the big and logical next step from two lines of research. One is research into disrupted edge networks (e.g. the DARPA SAPIENT program which found that an edge network that was down 10% of the time often caused the TCP/IP network to be down 50% or more and sought to reduce the penalty of downtimes). The other is research in networks for the other 3 billion (citizens of 3rd world countries with limited communications resources).

29. *What happens if we give radios multiple power levels (on, off, and one or more intermediate power modes)?* Jason Redi has developed a wonderful radio (now in its 3rd hardware generation) with three power levels (on, off and a low power "doorbell" mode that you can "ring" to tell a radio to wakeup) and shown it consumes 99% less energy with only slightly increasing delays for a range of traffic scenarios in multi-hop networks[61]. It is a wonderful result but it is also a point solution that begs for broader study. Is three modes the right number? Are there better algorithms for determining when to wake up a radio using the doorbell?

    Discussion: As we build low energy radios we need a roadmap that allows a designer to understand the energy tradeoffs of various decisions. Done right, this work will give us that roadmap and may open the door to even greater energy efficiency (99.99% anyone?).

30. *Packet headers for energy efficiency?* One feature of Redi's radio is that he was able to reduce energy costs by shortening the packet preamble. That's a reminder that, especially when sending small amounts of data, there's a tremendous overhead in preambles, MAC headers and trailers, and Internet packet headers. Years ago when bandwidth was short, we developed efficient methods for header compression on serial links[28]. Do the same techniques work when we seek to optimize energy use on inherently multicast wireless links? (The informal consensus appears to be "no"). If not, what is the right approach? Is there a complementary role for network coding?

    Discussion: Energy efficiency is a challenge at every layer in the protocol stack (cf. [62]) and I think this problem is likely to yield insights about energy efficient protocol design.

31. *What should be our standard traffic model for energy efficiency measurements?* A nasty problem in evaluating energy efficiency schemes is that, currently, we have no standard for traffic. Anyone is free to choose a traffic pattern that makes a proposed scheme look good. What should our standard traffic pattern(s) be?

    Discussion: Benchmarks all too often end up being gamed (manufacturers find ways to optimize to the benchmark). However, this early in the study of energy efficiency, I believe that having a common set of tests would be valuable. Defining a test, or suite of tests, that accurately captures the different ways networks are used (high load, low load, more or less bursty loads over various timescales, etc.) is likely to require both careful measurement and analytic research and, if successful, will enable the community to more effectively analyze the merits of various proposals (essential to moving forward).

32. *Designing a control channel for spectrum sharing*. Assume you have a collection of software-defined or software-determined radios in an area. How do they discover each other and decide how to divide up the spectrum among their applications? How do new radios entering the area learn of the current spectrum allocation and negotiate on behalf of their applications? One possibility is that the radios dynamically find each other, scanning the spectrum until they rendezvous – but the general version of this problem seems unsolvable. So most people I talk to assume that there will be some chunk of the spectrum, some frequency, defined as the control channel on which radios rendezvous. Furthermore, since this frequency is likely to be valuable (a frequency that is robust to fading etc.), the channel bandwidth is likely to be small. The question, then, is what protocol do we run on this channel and what does it do? For instance, is the channel simply where two or more radios find each other, and then they move to another frequency to exchange spectrum maps and agree on spectrum use? Or do we do everything on the control channel, so that newcomers and passive devices can just tune in and learn what to do? Two useful studies are [2], which assumes the control channel is locally assigned (I prefer to assume a national or regional assignment) and [64], which tries to think about how little information to exchange on a bandwidth constrained control channel.

    Discussion: The answer to this question would appear purely pragmatic – enabling better spectrum sharing. But I suspect the work also may yield insights into how to parsimoniously describe how the spectrum is being used. Also, while I note the rendezvous problem (find a common channel in a noisy dynamic spectrum where two radios' ranges partially overlap) appears insolvable, no one has (to my knowledge) proved it cannot be solved and I would be delighted to see a solution.

33. *Software-defined optical devices?* The optical communications community has been building progressively more configurable termination devices in recent years. The most sophisticated devices are arguably, nascent optical versions of the radio ASIC discussed in question 38. Interestingly, however, the optical communications community does not talk of programmability or real-time reuse of optical pass bands the way the radio community speaks of programming RF. I suspect (perhaps wrongly) there is a chance for cross-fertilizing ideas here and those ideas may lead to even more powerful ways to use optical fiber.

    Discussion: Fiber optic communication will remain a crucial part of our communications infrastructure for the foreseeable future. This research question seems likely to provide useful observations about how fiber optic communication could evolve.

34. *How much information can we extract from a time series?* Over the past several years there's been a trickle of papers showing that if you simply record when you see a packet and, sometimes, where the packet came from, you can extract an extraordinary amount of information from the resulting time series. For instance, you can determine the topology of a wireless network[12], you can learn what applications are present on the network[54], and how many sources[26]. There are suggestions that one can decompose an aggregate time series into the individual time series of the constituent conversations (the one hypothesis I know requires that the individual conversations' traffic be max-plus linear and mirabile dictu, TCP is max-plus linear[4][11]). How much more can we learn from simple times series?

    Discussion: There are several potential impacts of this work. First, it may make certain problems in network measurement easier because we will have to collect substantially less information from each packet to learn what we need to know. Second, there are implications for protecting encrypted traffic from traffic analysis. Third, there's been a hint in some work of underlying network "truths" – trial uses of principal component analysis on time series suggest only a handful of uncorrelated variables almost completely explain traffic timing[37]: what are those few variables and what do they tell us about our networks?

35. *How do we place information in a network so that users can access it efficiently?* The problem of locating information is the question one asks after creating a search engine. We have very successful search engines. Yet we are still struggling with information placement. Akamai has created one very useful solution. Peer-to-peer networks have created another. Jacobson's Content Centric Networks[68][29] is yet another, and Delay Tolerant Networks offer a fourth[18]. Cloud computing still needs a data placement model[69]. What is

interesting to me is that each of these approaches to information placement has spawned or is in the process of spawning a tremendous amount of research.[4] Yet much of this research is disjoint. That suggests to me that the research space here is still imperfectly explored and that there are likely other information placement approaches, and perhaps the possibility of a unified approach to data placement as yet unidentified that will yield new insights.

Discussion: This research question is, more than most in this list, a hunch based on my sense of the research space. My hunch is that either studies in additional types of information placement, or work on unification (finding out how to express the different schemes as points in a unified model), would be informative and offer us a rich set of follow on problems to study.

36. *Protocol verification*. Protocol verification, as a field, is deeply frustrating. Once in a while we see a paper that reminds us that verification gives us a unique perspective on a protocols – through the use of formal analysis, they allow us to see bugs and deficiencies and limitations that we did not see before[25][31]. Yet the effort required to verify a basic protocol such as TCP is huge and so time consuming that protocol designers simply accept they cannot make verification a standard part of creating a new protocol. How do we make progress in making verification easier to do? And does protocol verification have more relevancies as we look at exchanging/defining lower-layer protocols in software-defined radios? (See question 7).

    Discussion: The benefits are, I hope, obvious and the research space, while certainly well traveled, continues to offer periodic new insights. The big concern here is that talented people have been slogging at this problem for many years and progress is slow. There is, however, soon to be published work in semi-automatically identifying the most essential parts of software systems to verify (and focusing the verification on those parts) and perhaps this work can be transitioned to protocol work as well.

37. *What is the right abstraction for programming the cloud?* We have a few different paradigms for programming the cloud including mapreduce[16] and dryadlinq. They represent different mixes of expressiveness, power and safety. What are the merits of the different approaches? Are there other approaches we should approach? What answers change if we have homomorphic computing in our cloud[19] and wish to keep the type of computation we are doing private?

    Discussion: How we distribute computation across the net remains an important problem and one that we've yet to fully address. Mapreduce represented an important step forward and there are plenty more steps to take. From a communications perspective, it seems likely that the ways we choose to distribute computation may drive innovations or needs in the network.

5. Choosing another path

In mathematics, it is considered a useful result to reprove a known result, using a completely different approach from prior proofs (e.g. a different branch of mathematics). In computer science, we similarly consider different paths to similar results – mostly notably in computer languages, where we examine different ways to express the concept of programming a computer. In data networking, we seem more reluctant to consider multiple approaches (indeed, we often seem stuck on two communications models). But there are some places where following a different path could yield tremendous insight.

38. *Creating a richly configurable radio ASIC*. The idea of a software radio is well established in the field. Software radios contain processors such as DSPs or programmable logic such as FPGAs and can be programmed to implement any physical and media access layer over virtually any frequency. Examples of such radios include the SORA, WARP, JTRS and WNAN radios.[5] But as these radios have matured there's been an interesting *sotto voce* conversation among radio designers arguing that we don't need to use software, that we could build a highly configurable radio that could do anything a software radio could do. Intuitively this sounds plausible. One could imagine implementing a QAM engine that could be configured to do any of {16,64,128,256}-QAM. Or one could imagine implementing the components of QAM such as a Fourier transfer engine. A configurable random number generator plus a table of frequencies might implement spread spectrum. But to really prove this is feasible, someone has to actually do it.

    Discussion: The consequences of building this ASIC, if it proves possible, are surprisingly profound. First, it would bind what we know about RF physics and media access in one chip (suggesting a surprisingly contained research field). Second, there's reason to believe that such an ASIC would make the problems of both building a trusted spectrum agile radio and a cognitive radio easier[55].

39. *Mixing electrons and photons in optical transmission equipment*. For many years, the optical research community has sought to achieve all-optical networking. It has come close, developing densely packed photonic integrated circuits (PICs) but it is also clear that the inability to do optical regeneration means that optical equipment will continue to contain electronics in their data paths[6]. So how do we mix photons and electrons to give ourselves maximally useful optical networks?

    Discussion: Currently the intellectual leadership in this topic has passed to industry (e.g. Infinera Corporation). I think there are more challenges here than a product-driven organization can study and that good research will reveal insights that will enable us to better steer the future of optical communications.

40. *Efficient backplanes with redundancy for clusters?* If you've ever seen a computing cluster, you've seen just how horrific

[4] E.g. "Peer to Peer networks" yields 1.8M references in Google Scholar. Content delivery network (which is what Akamai offers) yields 1.4M references.

[5] Documentation for these radios consists of project websites. http://research.microsoft.com/en-us/projects/sora; http://www.darpa.mil/sto/solicitations/WNaN; http://warp.rice.edu; http://jtrsjpeo.mil

the wiring is among the devices[21]. Equally sobering is the discovery that the typical wiring plan, designed to provide redundancy in case of failures, is a Banyan network with all the serious throughput limitations that are inherent in Banyan networks. We know how to design better networks inside a computer (cf. work on switched backplanes for routers twenty years ago[76][75]) and we need to figure out how to create similarly flexible networks distributed over space.

Discussion: There are two contributions – advancing the state of distributed switching (a hard problem [48]) and creating good cluster interconnection architectures (preferably with equipment that allows one to simply plug new equipment in and have the cluster auto-reconfigure itself – also hard).

6. ACKNOWLEDGMENTS

I would like to thank Kilnam Chon, Jon Crowcroft, Michalis Faloutsos, Alden Jackson, Vikas Kawadia, John Wroclawski, several anonymous SIGCOMM reviewers and Grenville Armitage's research group at Swinburne University of Technology (who suffered hearing this list as a tech talk!) for their helpful comments and ideas.

7. REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, Network information flow. *IEEE Trans. Information Theory*, Vol 46 (2000), pp. 1204-1216

[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: a survey. *Comput. Netw*. 50, 13 (September 2006), 2127-2159. DOI=10.1016/j.comnet.2006.05.001 http://dx.doi.org/10.1016/j.comnet.2006.05.001

[3] Ashley-Rollman, M.P., Lee, P., Goldstein, S.C. Pillai, P. and Campbell, J.D. 2009. A Language for Large Ensembles of Independently Executing Nodes. *Proc. Intl. Conference on Logic Programming (ICLP '09)*.

[4] Baccelli, F. and Hong, D. 2000. TCP is max-plus linear and what it tells us on its throughput. In *Proc. ACM SIGCOMM 2000*. (Stockholm). 219-230. DOI= http://doi.acm.org/10.1145/347059.347548

[5] J. G. Bednorz and K. A. Mueller (1986). Possible high TC superconductivity in the Ba-La-Cu-O system. *Zeitschrift für Physik B* 64 (2): 189–193.

[6] Beeler, C. and Partridge, C. 2009. All-Optical Computing and All-Optical Networks are Dead. *Queue* 7, 3 (Apr. 2009), 10-10. DOI= http://doi.acm.org/10.1145/1530818.1530830

[7] Caesar, M., Caldwell, D., Feamster, N., Rexford, J., Shaikh, A., and van der Merwe, J. 2005. Design and implementation of a routing control platform. *Proc. 2nd Conf. on Networked Systems Design & Implementation* - Volume 2 (May 02 - 04, 2005). USENIX Association, Berkeley, CA, 15-28.

[8] Chiang, M., Low, S.H., Calderbank, A.R. and J. C. Doyle, J.C. 2007. Layering as optimization decomposition: A mathematical theory of network architectures. *Proc. IEEE*, 95, 1, (Jan 2007), 255-312.

[9] Clark, D. D., Wroclawski, J., Sollins, K. R., and Braden, R. 2002. Tussle in cyberspace: defining tomorrow's internet. *Proc. ACM SIGCOMM 2002*. (Pittsburgh, USA, August 19 - 23, 2002). 347-356. DOI= http://doi.acm.org/10.1145/633025.633059

[10] Clark, D. D., Partridge, C., Ramming, J. C., and Wroclawski, J. T. 2003. A knowledge plane for the internet. *Proc. SIGCOMM 2003*. (Karlsruhe, August 25 - 29, 2003). 3-10. DOI= http://doi.acm.org/10.1145/863955.863957

[11] Cohen, G., Dubois, D., Quadrat, J., and Viot, M. 1985. A linear-system theoretic view of discrete even processes and its use for performance evaluation in manufacturing. *IEEE Trans. Auto Control*, Vol. 30 (1985), pp. 210-220.

[12] Cousins, D., Partridge, C., Bongiovanni, K., Jackson, A.W., Krishnan, R., Saxena, T., and Strayer.W.T. 2003. Understanding Encrypted Networks Through Signal and Systems Analysis of Traffic Timing. *Proc. of IEEE Aerospace Conf.*

[13] Crowcroft, J. 2008. Cold topics in networking. *SIGCOMM Comput. Commun. Rev*. 38, 1 (Jan. 2008), 45-47. DOI= http://doi.acm.org/10.1145/1341431.1341440.

[14] Day, J. 2008. Patterns in network architecture. Prentice Hall (2008).

[15] Davidson, J. Hathaway, W., Postel, J., Mimno, N., Thomas, R., and Walden, D. 1997, The ARPAnet telnet protocol: its purpose, principles of implementation, and impact on host operating system design. *Proc. Fifth Data Communications Symposium*. 1977.

[16] Dean, J. and Ghemawat, S. 2008. MapReduce: simplified data processing on large clusters. *Commun. ACM* 51, 1 (Jan. 2008), 107-113. DOI= http://doi.acm.org/10.1145/1327452.1327492

[17] De Rosa, M., Goldstein, S.C., Lee, P., Pillai, P., and Campbell, J.D., 2009. A Tale of Two Planners: Modular Robotic Planning with LDP. *Proceedings IEEE Intl. Conf. on Intelligent Robots and Systems (IROS '09).*

[18] Fall, K. A delay-tolerant network architecture for challenged internets. *Proc. SIGCOMM 2003*. DOI=10.1145/863955.863960 http://doi.acm.org/10.1145/863955.863960

[19] Gentry, C. 2010. Computing arbitrary functions of encrypted data. *Commun. ACM* 53, 3 (Mar. 2010), 97-105. DOI= http://doi.acm.org/10.1145/1666420.1666444

[20] Haigh, K., Hussain, T.S., Partridge, C. and Troxel, G.D. 2008. Rethinking Networking Architectures for Cognitive Control. *Microsoft Research Cognitive Wireless Networking Summit 2008*.

[21] J. Hamilton. Data center networks are in my way. Stanford Clean Slate CTO Summit. 23 October 2009. http://www.mvdirona.com/jrh/work/

[22] Han, S., Jang, K., Park, K., and Moon, S. PacketShader: a GPU-Accelerated Software Router. *Proc. ACM SIGCOMM 2010* (New Dehli,,August 30-September 3, 2010).

[23] Heilmeier, G. Some Reflections on Innovation and Invention. Founders Award Lecture, National Academy of Engineering, Washington, D.C., Sept. 1992

[24] Hilbert, D. Mathematical problems. *Bulletin of the American Mathematical Society*, vol. 8, no. 10 (1902), pp. 437-479.

[25] Holzmann, G. 2004. SPIN Model Checker, The: Primer and Reference Manual. Addison Wesley, 2004.

[26] Hussain, A., Heidemann, J., and Papadopoulos, C. 2003. A framework for classifying denial of service attacks. *Proc SIGCOMM 2003* (Karlsruhe, August 25 - 29, 2003). 99-110. DOI= http://doi.acm.org/10.1145/863955.863968

[27] Hutchinson, N. C. and Peterson, L. L. 1991. The X-Kernel: An Architecture for Implementing Network Protocols. *IEEE Trans. Softw. Eng*. 17, 1 (Jan. 1991), 64-76. DOI= http://dx.doi.org/10.1109/32.6757

[28] Jacobson, V. 1990. RFC-1144: Header Compression for Low-Speed Serial Links. *Internet Requests for Comments*, No. 1144.

[29] Jacobson, V., Smetters, D. K., Briggs, N. H., Plass, M. F., Stewart, P., Thornton, J. D., and Braynard, R. L. 2009. VoCCN: voice-over content-centric networks. *Proc. ReArch 2009* (Rome, December 01 - 01, 2009). 1-6. DOI= http://doi.acm.org/10.1145/1658978.1658980

[30] Jun, E.-S., and Vaidya, N. 2002. A power saving MAC protocol for wireless networks. *Proc. IEEE INFOCOM 2002*, pp.

[31] Karsten, M., Keshav, S., Prasad, S., and Beg, M. 2007. An axiomatic basis for communication. *Proc. SIGCOMM 2007* (Kyoto, August 27 - 31, 2007). 217-228. DOI= http://doi.acm.org/10.1145/1282380.128240

[32] Kawadia, K. and Kumar, P.R. 2005. A cautionary perspective on cross layer design. *IEEE Wireless Communication Magazine*, 12, 1 (Feb 2005), 3-11.

[33] Kim, C., Caesar, M., and Rexford, J. 2008. Floodless in seattle: a scalable ethernet architecture for large enterprises. *Proc. SIGCOMM 2008* (Seattle, August 17 - 22, 2008). 3-14. DOI= http://doi.acm.org/10.1145/1402958.1402961.

[34] Kobayashi, K. 2008. Flexible arrays of inexpensive network (FAIN): toward global parallelism in the internet to satisfy future traffic growth. *Proc. CoNext 2008* (Madrid, December 09 - 12, 2008). 1-6. DOI= http://doi.acm.org/10.1145/1544012.1544075

[35] Kone, V., Yang, L., Yang, X., Zhao, B.Y., and Zhang, H. On the feasibility of effective opportunistic spectrum access. *Proc. IMC 2010.* 151-164. DOI=10.1145/1879141.1879160 http://doi.acm.org/10.1145/1879141.1879160

[36] Krugman, P. 2010. Building a green economy. *The New York Times Magazine*. 5 April 2010.

[37] Lakhina, A., Papagiannaki, K., Crovella, M., Diot, C., Kolaczyk, E. D., and Taft, N. 2004. Structural analysis of network traffic flows. *SIGMETRICS Perform. Eval. Rev*. 32, 1 (Jun. 2004), 61-72. DOI= http://doi.acm.org/10.1145/1012888.1005697

[38] W.E. Leland, M.S. Taqqu, W. Willinger, and D.V. Wilson. On the self-similar nature of Ethernet traffic. *Proc. SIGCOMM '*93. 183-193. DOI=10.1145/166237.166255 http://doi.acm.org/10.1145/166237.166255

[39] Little, J.D.C. A Proof of the Queueing Formula L = λ W.*Operations Research* 9, 383-387 (1961)

[40] Lucky, R. and Eisenberg, J., ed. 2006. *Renewing U.S. Telecommunications Research.* National Research Council (2006).

[41] Ludwig, R. Eliminating Inefficient Cross-Layer Interactions in Wireless Networking. Ph.D. thesis, RWTH Aachen, 2000.

[42] McHenry, M.A. 2005. *NSF Spectrum Occupancy Measurements - Project Summary* (on-line at http://www.sharedspectrum.com/inc/content/measurements/nsf/NSF_Project_Summary.pdf).

[43] Markoff, J. 2010. Scientists Strive to Map the Shape-Shifting Net. *The New York Times* (1 March 2010).

[44] Mathis, M., Mahdavi, J. Floyd, S., and Romanow, A. TCP Selective Acknowledgment Options. Internet Requests for Comments, RFC-2018. October 1996.

[45] Maxemchuk, N.F. Routing in the Manhattan Street Network. *IEEE Trans. on Commun*. May 1987, vol. COM-35, no. 5, pp. 503-512.

[46] Metcalfe, R. M. and Boggs, D. R. 1976. Ethernet: distributed packet switching for local computer networks. *Commun. ACM* 19, 7 (Jul. 1976), 395-404. DOI= http://doi.acm.org/10.1145/360248.360253

[47] Mitola, J. Software Radios – survey, critical evaluation and future directions. *National Telesystems Conference*, 1992.

[48] Mills, D.L, Boncelet, C.G., Elias, J.G., Schragger, P.A., and Jackson, A.W. Highball: a high speed, reserved-access, wide-area network. Electrical Engineering Department Report 90-9-3, University of Delaware, September 1990.

[49] Moy, J., Pillay-Esnault, P. and Lindem, A. Graceful OSPF restart. *Internet Requests for Comments*, No. 3623. Nov 2003.

[50] National Research Council. *Network Science.* The National Academies Press (2005).

[51] Nosratinia, A., Hunter , T.E., and A. Hedayat, A., 2004. Cooperative communication in wireless networks. *IEEE Communications Magazine,* 42, 10 (Oct 2004), 68-73.

[52] Nychis, G., Fallin, C., Moscibroda, T. and Mutlu, O. Next generation on-chip networks: What kind of congestion control do we need? *Proc. ACM HOTNETS IX* (2009).

[53] Partridge, C., Mendez, T., and Milliken, W. Host anycasting service. *Internet Requests for Comments*, No. 1546. November 1993.

[54] Partridge, C., Cousins, D., Jackson, A. W., Krishnan, R., Saxena, T., and Strayer, W. T. 2002. Using signal processing to analyze wireless data traffic. In *Proceedings of the 1st ACM Workshop on Wireless Security* (Atlanta, GA, USA, September 28 - 28, 2002). 67-76. DOI= http://doi.acm.org/10.1145/570681.570689

[55] Partridge, C. Realizing the Future of Wireless Data Communications. To appear in *Commun. ACM.*

[56] Patterson, D. 2002. How to have a bad career in research/academia. Feb 2002. http://www.cs.berkeley.edu/~pattrsn/talks/BadCareer.pdf.

[57] Patterson, D. The trouble with multicore. *IEEE Spectrum*, July 2010.

[58] Popa, L., Ghodsi, A. and Stoica, I. HTTP as the Narrow Waist of the Future Internet. *Proc. ACM HOTNETS 2010.* Nov 2010.

[59] Ray S., Guerin, R., and Sofia., R. A distributed hash table based address resolution scheme for large-scale Ethernet networks. In *Proc. IEEE Intl. Conf. on Communications 2007*.

[60] Raymond, E. The iron laws of network cost scaling. Blog posting at ibiblio.org.

[61] Redi J., Kolek, S., Manning, K., Partridge, C., Rosales-Hain, R., Ramanathan, R. and Castineyra, I., JAVeLEN – an ultra-low energy ad hoc wireless network. *Ad Hoc Networks*, Vol. 6, No. 1 (2008), pp. 108-126.

[62] Riga, N., Matta, I., Medina, A., Partridge, C., and Redi, J. 2007. An energy-conscious transport protocol for multi-hop wireless networks. *Proc. CoNext 2007* (New York, December 10 - 13, 2007). 1-12. DOI= http://doi.acm.org/10.1145/1364654.1364683.

[63] Saltzer, J. On the naming and binding of network destinations. *Internet Requests for Comments*, No. 1498, Aug 1983.

[64] Santivanez, C., Ramanathan, R., Partridge, C., Krishnan, R., Condell, M. and Polit, S. Opportunistic Spectrum Access: Challenges, Architecture, Protocols. *Proc. 2nd Annual International Wireless Internet Conference*, August 2006

[65] Shneidman, J. and Parkes, D. C. 2004. Specification faithfulness in networks with rational nodes. *Proc. PODC 2004* (St. John's, July 25 - 28, 2004). 88-97. DOI= http://doi.acm.org/10.1145/1011767.1011781

[66] Singh, S. *Fermat's Enigma*. Walker and Company (1997).

[67] Sobel, D. *Longitude*. Walker and Company (1995).

[68] Stanik, J. 2009. A Conversation with Van Jacobson. *Queue* 7, 1 (Jan. 2009), 8-16. DOI= http://doi.acm.org/10.1145/1508211.1508215

[69] B. Tiwana, M. Balakrishnan, M.K. Aguilera, H. Ballani, and Z. M. Mao. Location, location, location!: modeling data proximity in the cloud. *Proc. ACM HOTNETS 2010.* Nov 2010.

[70] Turner, J. S., Crowley, P., DeHart, J., Freestone, A., Heller, B., Kuhns, F., Kumar, S., Lockwood, J., Lu, J., Wilson, M., Wiseman, C., and Zar, D. 2007. Supercharging planetlab: a high performance, multi-application, overlay network platform. *Proc. SIGCOMM 2007* (Kyoto, August 27 - 31, 2007). 85-96. DOI= http://doi.acm.org/10.1145/1282380.1282391

[71] Van Meter, R., Ladd, T. D., Munro, W. J., and Nemoto, K. 2009. System design for a long-line quantum repeater. *IEEE/ACM Trans. Netw*. 17, 3 (Jun. 2009), 1002-1013. DOI= http://dx.doi.org/10.1109/TNET.2008.927260

[72] R. Van Meter, J. Touch and C. Horsman. Recursive Quantum Repeater Networks. To appear *Progress in Informatics*, Mar 2011.

[73] Yang, Q., Siganos, G., Faloutsos, M., and Lonardi, S. Evolution versus Intelligent Design: Comparing the Topology of Protein-Protein Interaction Networks to the Internet. *LSS Computational Systems Bioinformatics Conference (CSB'06)*. 299-310, Stanford, CA, August 2006.

[74] Ye, W., Heidemann, J., and Estrin, D. 2004. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw*. 12, 3 (Jun. 2004), 493-506. DOI=http://dx.doi.org/10.1109/TNET.2004.828953

[75] Yeh, Y-.S., Hluchyj, M.G., and Acampora, A. The knockout switch: a simple, modular architecture for high-performance packet switching. *IEEE Jour. Selected Areas in Communications*, Vol 5, No. 8, (October 1987), pp. 1274-1283.

[76] Zegura, E.W. Architectures for ATM switching systems. *IEEE Communications Magazine*, Vol 31, No. 2, (February 1993), pp. 28-37.

[77] Zhao, Q. and Sadler, B. A survey of dynamic spectrum access. *IEEE Signal Processing Magazine*. (May 2007).

[78] Zimmerman, H., 1980. OSI reference model – the ISO model of architecture for open systems interconnection. *IEEE Transactions on Communications,* 28, 4 (Apr. 1980), 425-432.