I/S: A JOURNAL OF LAW AND POLICY FOR THE INFORMATION SOCIETY

How Smart Do Networks Need to Be?

JOE WEINMAN[*]

Abstract: The current paradigm for distributed processing in support of services such as web search, social networking, microblogging, e-commerce, and entertainment generally assumes that these services are delivered across a best-efforts network whose functionality should be limited to packet delivery and with routing decisions made locally. While such an approach has benefits such as resilience, there are scenarios where such a network delivers suboptimal performance to end users as well as inefficient resource utilization. Other approaches, in which the network can provide or incorporate near-real-time information on congestion, enable intelligent control of routing, and make optimizing decisions, may provide superior performance without loss of resilience. However, such an approach has implications for both network architecture and regulatory policy.

[*] Joe Weinman is Senior Vice President, Telx, and the author of *Cloudonomics: The Business Value of Cloud Computing*. Prior to Telx, he held executive positions at HP, AT&T, and Bell Laboratories. He has been awarded 18 U.S. and international patents in areas ranging from pseudoternary line coding to distributed cloud storage and network services, and has a B.S. and M.S. in Computer Science from Cornell University and UW-Madison respectively. He is also the chair of the IEEE Intercloud Testbed executive committee.

The author would like to thank H. N. Nagaraja, co-author of the standard textbook *Order Statistics* (Herbert A. David and H.N. Nagaraja, *Order Statistics*, 3rd ed. (New York: John Wiley & Sons, 2004), and Professor of Biostatistics, College of Public Health, Ohio State University, for his incredible thoroughness and responsiveness in analytically assessing the surprisingly complex order statistics of the triangular distribution.

I. INTRODUCTION

Various designs for flying objects and machines such as hot air balloons, ballistic missiles and cannonballs, gliders, and box kites date back centuries, if not millennia. The hot air balloon offers little positional control, traveling merely downwind; the cannonball, once shot, has a predictable and inalterable trajectory. The Wright brothers are credited with the first controlled, powered flights, conducted beginning in 1903. Use of airplanes–for mail, passengers, and war–grew exponentially, and by 1921, a mere 18 years later, the world's first air traffic control system was introduced at London's Croydon airport. To put it another way, to optimize a balance of cost, reliability, and performance, a nontrivial amount of decision-making responsibility was taken away from the pilot–who could at best make only locally optimal decisions–and given to a central authority–which could make better decisions based on richer information.

While advances in physical transportation largely defined that age, today, information technology and telecommunications are critical drivers of the global economy, and thus advances in data transport are important. The latest technologies for distributed computing–such as mobility, social networking, the cloud, big data, and the like–have data networking as their core enabler. Three paradigms for air transport have analogues in data transport. The cannonball or hot air balloon is roughly analogous to a dedicated point-to-point private line: data is poured in at one end and arrives deterministically at the other. Air travel before 1921 is roughly analogous to today's Internet architecture. Routers (locally) decide the path that packets should take, the way that pilots planned their own routes through increasingly congested airspace. The global air traffic control system illustrates the possibilities inherent in a "smart network," such as one enabled by "Software Defined Networking" technologies and standards such as OpenFlow,[1] where routing decisions may be decoupled from the local element, and centralized in an intelligent controller. For both air transport and data transport, separating and logically centralizing network monitoring, decision-making, and control can have compelling systemic advantages.

While there are no doubt numerous denotations and connotations of "smart," one may generally consider a smart system to be one which acquires and processes data using algorithms and heuristics in such a

[1] Jad Naous, David Erickson, G. Adam Covington, Guido Appenzeller, and Nick McKeown, "Implementing an OpenFlow Switch on the NetFPGA platform" (lecture, Architecture for Networking and Communications Systems (ANCS), San Jose, CA, Nov. 6-7, 2008).

way as to adapt to changing conditions so as to continuously optimize performance.

Optimization–doing more with less–is an important technical and societal objective. Economically, digital-native businesses are a significant growth area with companies such as Facebook, Google, and Baidu generating many billions of dollars in revenue directly and facilitating commerce worth many billions more. However, minimizing resource consumption–whether of capital stock such as physical hardware or resources such as energy–during such value creation is a worthwhile objective as well. Efficient resource utilization coupled with net new revenue contributes variously to GDP, national competitiveness, standard of living, social welfare, and productivity.

These effects extend far beyond "digital businesses." Much of the rest of the global economy is dependent on information technology as well: airlines need flight scheduling, aircraft assignment, crew scheduling, and online reservations systems; manufacturers need shop floor control, forecasting, quality management, and supply chain management systems; retailers need inventory control, point of sale, and business analytics systems; and so forth, vertical by vertical.

Moreover, what might even be viewed as low-technology segments utilize IT: mines use seismic analysis to determine where to dig;[2] farms optimize irrigation;[3] fishermen determine where to sell their catch.[4] There are numerous social and political imperatives–and thus regulatory and policy concerns–for optimizing networked information technology: consumers, citizens, and governments use IT to engage populations, to acquire information and entertainment, and to rally for social causes.

Traditionally, the "information technology" segment has been notably divorced from the "communications technology" segment, in terms of expertise, educational programs, corporate structure, and regulatory policy. For example, information technology expertise might involve virtualization, programming languages, microprocessor architecture, and web page design; but communications expertise might involve configuring routers, protocol design, and optical

[2] Linda Tischler, "He Struck Gold on the Net (Really)," *Fast Company*, June 2002, http://www.fastcompany.com/ 44917/he-struck-gold-net-really.

[3] Daisuke Wakabayashi, "Japanese Farms Look to the 'Cloud'," *Wall Street Journal*, Jan. 18, 2011.

[4] Robert Jensen, "The Digital Provide: Information (Technology), Market Performance, and Welfare in the South Indian Fisheries Sector," *Quarterly Journal of Economics* 122, no. 3, (2007): 879-924.

interconnects. Organizationally the two are typically separate. From a regulatory perspective, common carriers are prevented from doing much with the packets traversing their IP networks. They are able to assess aggregate traffic for network capacity planning and design; implement security protocols only to the extent necessary to defend against cyber-attacks; and generally are unable to perform deep packet inspection. Information services are considered a substantially different arena, with largely different regulations. Network technologies, and higher-level services such as the cloud, have thus become of interest to lawyers, regulators, and policymakers.[5]

Back when technology architecture comprised dumb terminals accessing mainframes over dedicated private lines, the regulatory regime may have had some issues, but generally appeared to function. Today, however, rather than separate silos of information technology and communications, the world is moving towards a unified, distributed processing fabric. This processing fabric is made up of a variety of service nodes linked to each other and to users as well as endpoints such as connected devices belonging to the "Internet of Things" linked across a mesh network. Under certain circumstances, and for certain types of applications, better performance can be achieved at lower cost by recognizing this convergence. Elsewhere,[6] we delve into the order statistics of the triangular distribution[7] that quantify these effects; in this paper we will merely cite those results and address the effects qualitatively.

II. A SIMPLIFIED DISTRIBUTED PROCESSING ARCHITECTURE

Assume that the world comprises multiple "users" accessing multiple services across a mesh packet network. By "user," we also mean machine-to-machine endpoints. As these users access services–say, email or web searches–various routes will be utilized to access either the single point where the service is offered, or perhaps multiple service nodes where the same service is offered. We can think of this as people searching for entertainment in the physical world. In

[5] Cf. Deborah Salons, "*Cloudonomics:* The Business Value of Cloud Computing," *Federal Communications Law Journal* 65, no. 3, (2013): 297-311.

[6] Joe Weinman, "Better Together: Quantifying the Value of Smart Networks" (working paper 2013), http:// www.joeweinman.com/Resources/SmartNetwork.pdf.

[7] H.N. Nagaraja, "Moments of order statistics and L-moments for the symmetric triangular distribution," *Statistics and Probability Letters* 83, no. 10 (2013): 2357-2363.

the case of touring the Statue of Liberty, there is only one viable service node. However, in the case of seeing the latest Hollywood blockbuster, there are many possible movie theaters, each showing the same movie. Or, in the case of caffeinated beverage acquisition, there may be many different coffee shops, each offering essentially the same latté. As services are requested and delivered, various routes will be taken, leading in aggregate to certain routes that are congested at different times. Assuming the users and service nodes are interconnected through a mesh network, whether there are one or more nodes offering the same service, there will be multiple paths available to access the service. For example, even though there is only one Statue of Liberty, there are a number of ways to get to it. These paths will have a fixed network latency component based on propagation characteristics of the medium and physical distances traversed, and a stochastically[8] varying portion based ultimately on the patterns of requests being made by various users over time. By analogy, the time it takes to drive from New York to Los Angeles is partly a function of the route selected and its series of roads of various lengths and speed limits, and partly a function of traffic encountered and the luck of traffic light timing. Rather than converging to some sort of steady state equilibrium, various factors conspire to create variability, which may be highly erratic. This includes spikes at news sites due to natural phenomena such as earthquakes and hurricanes or news items such as scandals and deaths. It includes floods at sites due to the "Slashdot" effect, where mention of a particular site at a different popular site causes many web surfers to suddenly visit that site. Then there are cyber-attacks such as distributed denial of service attacks, which artificially cause traffic to increase massively at target sites. One recent flood was estimated to be generating over 300 Gigabits per second.[9]

Now consider a particular user who is desirous of accessing a service delivered at multiple different points. A time-constrained user would like to get his latté as quickly as possible, necessitating finding the shop that is the nearest, as well as the one with the shortest wait. By "nearest," we don't necessarily mean the closest shop as the crow flies or in terms of street driving miles, but in terms of time to get there. In an ideal situation, the coffee shop with the shortest wait

[8] I.e., randomly.

[9] Peter Bright, "Spamhaus DDoS grows to Internet-threatening size," Mar. 27, 2013, http://arstechnica.com/ security/2013/03/spamhaus-ddos-grows-to-internet-threatening-size.

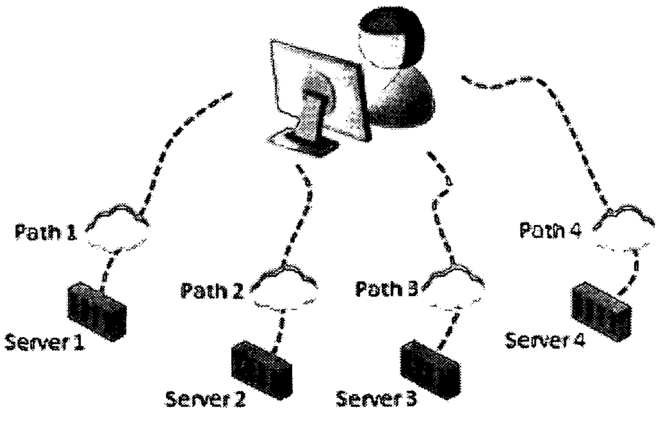
would also be the closest; but a moment's thought shows that the best option may not be the closest shop or the one with the shortest wait, but one where neither is true. For example, a shop with a five minute wait located five minutes away will provide a caffeine fix faster than one with a one minute wait located ten minutes away and also faster than one located one minute away with a ten minute wait.

Similarly, a computing request is not necessarily served most quickly by either the service node with the shortest wait or by the path with the fastest response time (based on physical propagation delay[10] and congestion). Rather, the fastest service for that user can be realized by the combination of path plus server with the lowest *total.*

III. A SIMPLE MODEL

In a simple model of a distributed processing architecture with a single user and multiple service nodes, we can assume that each node has exactly one path to it.

Figure 1. A simple model of a distributed processing architecture



[10] Propagation delay is the time it takes for a signal or message to get from point A to point B, simply based on the physical characteristics of photons in optical media, such as fiber or electrons in electrical media such as copper.

We can then evaluate four different strategies for selecting among the four pairs, each comprising a node and a path.

- **RANDOM**–In this approach, the server, and thus the path reaching the server, are selected at random. In the context of a coffee shop, this would be equivalent to throwing a dart at the telephone directory listings for coffee shops.

- **SERVER-BASED**–Here the server with the fastest response time is selected. In the coffee shop setting, this would entail selecting the location with the shortest wait, based on line size, orders in process, the baristas, and other coffee preparation resource capabilities and quantities.

- **PATH-BASED**–Here the path with the fastest response time is selected. In the real world, it would entail selecting a shop based on the best combination of driving distance, speed limits, traffic lights, congestion, and turns.

- **JOINT OPTIMIZATION**–In this approach, the fastest combination of server and path are selected. In the real-world context we have been discussing, this would be the best combination of coffee shop conditions and driving characteristics to get there.

If there is only one service node and only one path to that service node, which approach we use will not make any difference. If however, there are several, there are likely to be performance differences between the algorithms. We can then ask, how big *are* the differences, and how do these differences change as the number of choices of service nodes, and thus paths to those service nodes, increases.

IV. PROBLEM ABSTRACTION AND ORDER STATISTICS

Such a problem–and the heuristics and algorithms for solving it–is not tied solely to information technology and data networking: as we have seen it can arise with physical services, such as coffee shops, and physical networks, such as streets and highways or air transport networks. In fact, it is not even restricted to the time domain. It will be apparent that a selection of least-cost vacation destination could be made at random, based on the lowest-price resort, based on the lowest price airline ticket, or based on the lowest total price including air and hotel.

As an initial abstraction, consider results from rolling dice. Suppose that we roll three dice at once. We might get a "6," "4," and "5" one time, a "3," "2," and "1" the next time, a "2," "5," and "1" the next time, and so forth. If we *sort* the triples from each of these trials, we can restate the results as "4," "5," "6," from the first roll, a "1," "2," and "3" the second time, and a "1," "2," and "5" the third time. The first time, the minimum was a "4," the next two times, the minimum was a "1." A simple question one might ask is "what is the expected value of the minimum roll, if we roll a particular number of dice?"

The general problem–that of determining the expected value of the minimum sample drawn from a set of samples selected from a random variable with a given distribution–falls into the general category of *order statistics*. Order statistics tells us the expected value of the smallest sample, the expected value of the next smallest, and so forth up to the expected value of the largest sample.

For example, suppose that we took samples of a random variable that had a uniform distribution on the interval between zero and one hundred. Any given single sample might fall anywhere in that range. The first sample might be 37. The second might be 88. The third might be 99.5. Any *particular* sample might fall anywhere between zero and one hundred, but the *expected value* of the sample would be exactly fifty. Suppose we took two samples. We might have one sample that was eighty, and another one that was .0000012345. In this case the second sample would be the minimum for that trial. In another trial, we might have one sample be .9, and the other one be .900000000000000000000000001. In general, it turns out that when we draw two samples of a random variable with this distribution, the expected value of the minimum is thirty-three and a third.

Now suppose that both path and service node response times are uniformly distributed between zero and one hundred milliseconds.[11] If we select any particular node at random, it might have a response time anywhere in that range, but we would *expect* its average response time to be fifty milliseconds. In other words, the "expected value" of its response time is fifty milliseconds. If we select any particular path

[11] Both service node response times and path latencies are assumed to be independent, identically distributed continuous random variables in this simplified base case.

at random, it too might have a response time anywhere in that range, but we would *expect* its response time to also be fifty milliseconds. In this case, a combination of service node and path selected at random would have an expected total response time of one hundred milliseconds.

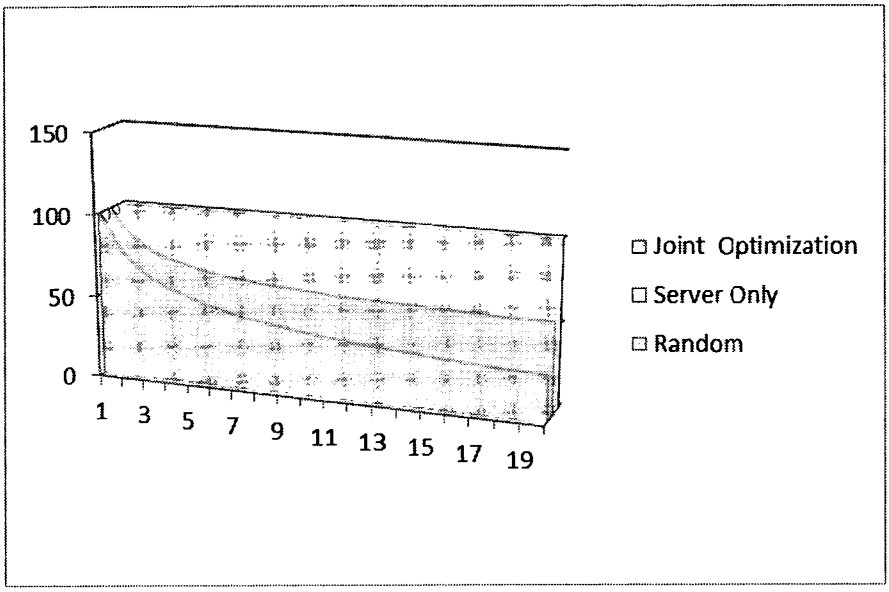
Suppose that we select the best possible service node. If we only have one option, it would have an expected response time of fifty milliseconds. However, as the number of nodes increases, we would expect to find the expected value of the minimum to decrease. There is a formula for this,[12] but to keep things simple, let us assume that we have plenty of options. In fact, let's assume that we have an infinite number of choices. In this case, we would expect the expected value of the best choice to be zero milliseconds. However, by picking this particular service node, we end up picking the path to that node at random which means that the expected value of the path latency is fifty milliseconds. Therefore, even with an infinite number of choices, the expected value of the server-based algorithm is fifty milliseconds. Using the mirror image argument, we see that picking the best path without regard to the server also gives us an expected value of fifty milliseconds.

However, suppose we had an infinite number of combinations of path and service node to choose from. Rather than a selection just based on the service node, with the path therefore being selected at random, or just based on the path, with the service node thereby being selected at random, suppose that we picked the lowest total. The expected value of the minimum combination would clearly be zero milliseconds. While there would be plenty of really bad combinations, say a hundred milliseconds and a hundred milliseconds, and some average combinations, such as (0,100), (100,0), or (50,50), there would be quite a few (an infinite number, in fact) of combinations that came arbitrarily close to (0,0).

With a finite number of combinations, we will not do quite that well, but this logic tells us that a server-based or path-based approach is always better than a random one and, importantly, that a joint optimization approach is the best approach of all. Formal analysis, using the theory of order statistics, as well as simulation results, shown below, bears this out.

[12] $E\left(S_{(1)}^{n}\right)=\frac{1}{n+1}$

FIGURE 2. Simulation Results Comparing Algorithms[13]



As the number of choices increases, the random algorithm does not generate improved results, but rather generates a total result that is the sum of the expected value of the path response time together with the expected value of the service node response time.

As the number of choices increases, the server-based algorithm has some rapid gains, but exhibits diminishing returns to scale and can never do better than the expected value of the path response time.

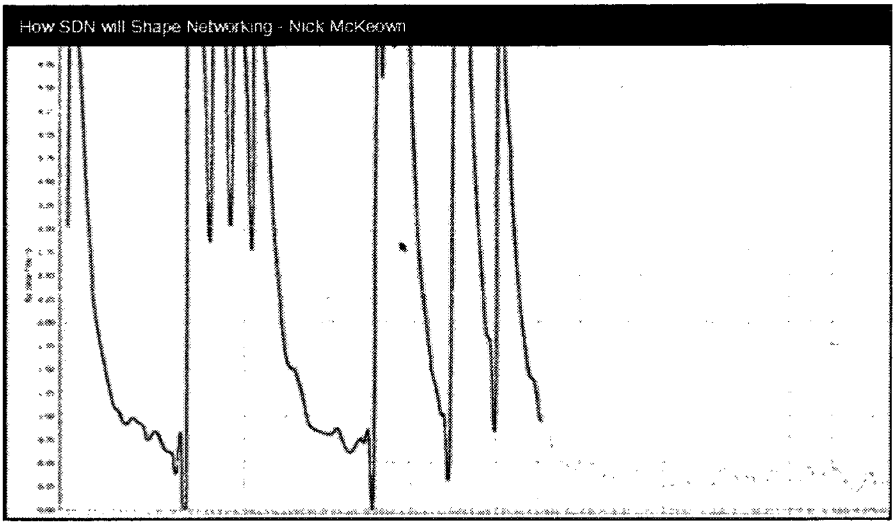
However, as the number of choices increases, the joint optimization algorithm, while also exhibiting diminishing returns, gets better and better, and as expected, reaches zero in the limit as the number of options increases.

An experiment conducted by Nick McKeown, Nikhil Handigol, and their colleagues run on the GENI (Global Environment for Network Innovations) network test bed supports the purely mathematical conclusions and the Monte Carlo simulation results.[14] As can be seen from the chart below, the server-based algorithm shown (on the left) does not perform anywhere near as well as one that incorporates network path optimization (on the right).

[13] Joe Weinman, *Complex Models*, http://complexmodels.com/SmartNetwork.htm.

[14] Nick McKeown, "How SDN will Shape Networking - Nick McKeown," http://www.youtube.com/watch?v=c9-K5O_qYgA.

FIGURE 3. Server-based selection vs. Network-based selection[15]



V. Implications

This simple and straightforward analysis has potentially profound implications on network architecture, distributed processing services such as cloud computing, and regulatory policy.

First, it suggests that users need to have multiple choices to maximize performance. The more choices that users have, the better response time they will experience for interactive applications. These choices may include everything from multiple providers, to mesh network architectures, to dispersed data center footprints rather than consolidated ones. Free markets ensuring competitive choice can help maximize choice, as well as standards and services that facilitate interoperability, such as for the emerging Intercloud.[16] The benefits and architectural requirements for such markets can also be

[15] *Ibid.*

[16] Institute of Electrical and Electronics Engineers, "IEEE Intercloud Testbed in conjunction with IEEE P2302," http://cloudcomputing.ieee.org/intercloud.

quantified using the theory of order statistics and Monte Carlo simulation.[17]

Second, it suggests that some sort of supervisory function that can direct users to services by accounting for both network congestion and service node congestion can be useful. Moreover, it argues in favor of "smart networks," i.e., ones that rather than merely routing packets based on a combination of best efforts and limited information, can understand the state of network congestion, and specify routes. The supervisory function can be embedded in "the network," or can be performed by a third party, but the requirement that the network layer provide status and be able to route specific packets or flows remains. This means that network service providers must be legally able to provide certain types of data to third parties, or utilize the data themselves. This requires achieving a delicate balance between privacy and network optimization. Even without deep packet inspection, mere raw data concerning patterns of flows between two IP addresses may be sensitive.

There are clearly benefits to today's Internet architecture based on "stupid networks,"[18] such as growth, accelerated innovation by providing a platform for services,[19] and a high degree of resilience[20] due to no single point of failure, including no single point of control. However, compared to a typical basic Internet architecture, a variety of proposals, services, and initiatives offer greater intelligence and control, e.g., Multiprotocol Label Switching Virtual Routing and Forwarding, the Intelligent Route Service Control Point,[21] Software-

[17] Joe Weinman, "Quantifying the Value of a Cloud Computing Market," Telx (blog), October 24, 2013, http:// www.telx.com/blog/quantifying-the-value-of-a-cloud-computing-market.

[18] David Isenberg, "The Rise of the Stupid Network," *Computer Telephony* (Aug. 1997): 16-26.

[19] Barbara van Schewick, *Internet Architecture and Innovation* (Cambridge, MA: MIT Press, 2010).

[20] Albert-Lászl6 Barabási, *Linked: How Everything is Connected to Everything Else and What It Means for Business, Science, and Everyday Life* (New York: Perseus Books, 2002).

[21] J. Van der Merwe, A. Cepleanu, K. D'Souza, B. Freeman, A. Greenberg, D. Knight, R. McMillan, et al., "Dynamic Connectivity Management with an Intelligent Route Service Control Point" (lecture, Special Interest Group on Data Communication (SIGCOMM), Pisa, Italy, Sept. 11-15, 2006).

Defined Networks and OpenFlow,[22] Network Functions Virtualization,[23] the OpenStack Quantum/Neutron[24] effort and Network as a Service, Cisco's Network Positioning System,[25] Internap's Managed Internet Route Optimizer,[26] Multipath Interdomain ROuting,[27] Application-Layer Traffic Optimization,[28] and so forth.

VI. PLACING JOINT OPTIMIZATION IN CONTEXT

To be fair, one must be wary of drawing too many conclusions from this thought experiment (and its supporting statistical analysis, simulation, and GENI test bed experiment).

Generally, path latencies are heavily dependent on network propagation delays, which in turn are heavily dependent on network route distances. To the extent that these distances vary greatly across available service nodes, and are correlated with, say, other cost

[22] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review* 38, no. 2 (2008): 69-74.

[23] Margaret Chiosi, Don Clarke, Peter Willis, Andy Reid, James Feger, et al., "Network Functions Virtualisation" (lecture, Software-Defined Networking (SDN) and OpenFlow World Congress, Darmstadt, Germany, Oct. 22-24, 2012), http://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[24] Sean Michael Kerner, "Open Source OpenStack Quantum Networking Renamed to Neutron?," *InternetNews.com*, July 10, 2013, http://www.internetnews.com/blog/skerner/open-source-openstack-quantum-networking-renamed-to-neutron.html.

[25] Cisco Systems, Inc., "Cisco Network Positioning System,“ http://www.cisco.com/en/US/solutions/collateral/ns341/ns991/ns1174/Network_Positioning_System.pdf.

[26] Internap Network Services Corporation, "MIRO: Internap's Internet Latency Solution," http://www.internap.com/business-internet-connectivity-services/performance-ip/route-optimization-miro/ (accessed Oct. 24, 2013).

[27] Wen Xu and Jennifer Rexford, "MIRO: Multi-path Interdomain ROuting" (lecture, Special Interest Group on Data Communication (SIGCOMM), Pisa, Italy, Sept.11–15, 2006).

[28] V.K. Gurbani, V. Hilt, L. Rimac, M. Tomsu, and E. Marocco, "A survey of research on the application-layer traffic optimization problem and the need for layer cooperation," *IEEE Communications* 47, no. 8 (2009).

metrics such as hop counts or round trip times, existing routing algorithms such as Open Shortest Path First or Border Gateway Protocol will tend to select the best path. Also, every user is unlikely to be at *exactly* the same baseline distance from every service node. Suppose that there are service nodes in Boston, Miami, San Diego, and Seattle. While people in Kansas, or perhaps even large swaths of the Midwest are likely to be equidistant from multiple nodes, those living in, say, Providence will not need a very sophisticated algorithm to determine that the Boston service node is likely to be best. On the other hand, if the Boston node suffers a failure, perhaps the other three locations would become equally viable options.

An assumption has been made that path latencies are independent from service node latencies. However, in some cases, say, Internet floods or DDoS attacks, a highly congested service node will have a highly congested path or paths leading to it. In this case, selection based on server load would be identical to selection based on joint optimization.

We have assumed that armed with the right information, routing decisions can be made on a timely basis. This in turn requires that areas of network congestion remain stable long enough for decisions to be made, that such calculations are computationally tractable, and that decisions can then be implemented quickly enough to take advantage of network state.

We also have assumed that the state of the network shifts frequently enough that an algorithm is worthwhile. If, for example, the state of the network only changes once a year, such an analysis could be done offline by someone with a calculator, or at least a spreadsheet application, and implemented in the network in a more static fashion.

We have also made the assumption that such changes in network routing do not jeopardize existing sessions: one would not want to spend an hour on an e-commerce site filling up a shopping cart, only to have the session re-routed to a replica site with no knowledge of this activity. Research has shown that algorithms that successfully transition routes while preserving "micro-flows" appear to be feasible to implement at scale.[29]

We also have implicitly assumed that we are not giving up more than we are getting from the current architecture: resilience, scale, etc. For example, a single central supervisory node implementing this

[29] Richard Wang, Dana Butnariu, and Jennifer Rexford, "OpenFlow-Based Server Load Balancing Gone Wild," http://static.usenix.org/event/hotice11/tech/full_papers/Wang_Richard.pdf.

algorithm would be prone to failure, if not outright cyber-attack. It would appear, however, that in the event of such a failure or attack, a network could be designed to fail gracefully, returning to the *status quo*. Moreover, while intelligent network architectures, such as Software-Defined Networks and OpenFlow are designed to be *logically* centralized, they can be *physically* implemented in a distributed fashion, enhancing scalability, performance, and resilience.

VII. QUESTIONS FOR FURTHER RESEARCH

If smart networks can, at least some times, offer clear quantitative benefits over best-efforts networks with localized decision-making, a number of technical and policy questions arise.

- For what other applications (use cases) do smart networks have provably superior performance? We have quantified "anycast" services here, where a given service is available from multiple nodes and a user can be provided identical functionality from any of those nodes.

- What are the regulatory implications of achieving such quantifiable benefits? These clearly include privacy and security constraints on deep packet inspection, but there are other areas to consider as well. If quality service depends on intelligent access to multiple nodes, should there be a concept of universal service that extends beyond mere service availability to multiple choices and performance optimization among those choices?

- What are the issues surrounding monetization and the market ecosystem? Network operators have long been interested in at least a portion of the revenue flows associated with "over the top" ecosystem participants, as opposed to offering mere "plumbing." If smarter networks enable premium-priced services, who should benefit?

- If welfare is enhanced by better performing applications, but requires substantial investment in network intelligence and optimization algorithm deployments, how to best match investment and reward?

- If smart distributed architectures require collaboration among network providers as well as between network providers and (cloud or application) service providers, how to ensure cooperation without collusion?

- Does network architecture with logically centralized control offer a tempting cyber-attack target? How resilient can such architecture be made? What is the role of government in mandating such resilience, and should such architecture be the focus of national cybersecurity policy?

- How well do current algorithms perform and how much better would results be from joint optimization approaches?

- What are the tradeoffs–such as resilience and decision cycle times–between a single centralized architecture and a distributed, federated approach?

VIII. SUMMARY

Today we live in an increasingly digital, collaborative, and connected world, which Don Tapscott calls the "Age of Networked Intelligence."[30] At least some applications can be quantitatively and rigorously shown to perform better when information concerning the state of the network is combined with information concerning the state of computing resources and used for real-time decision making in a distributed computing system. However, achieving these benefits

[30] Don Tapscott, *The Digital Economy: Promise and Peril In The Age of Networked Intelligence* (New York: McGraw-Hill, 1997).

may require rethinking not just technology and architecture but also regulatory policy.