2013

The NEBULA Future Internet Architecture

Tom Anderson
University of Washington

Ken Birman
Cornell University

Robert Broberg
Cisco Systems

Matthew Caesar
University of Illinois at Urbana-Champaign

Douglas Comer
Purdue University

See next page for additional authors

Repository Citation

Anderson, Tom; Birman, Ken; Broberg, Robert; Caesar, Matthew; Comer, Douglas; Cotton, Chase; Freed, Michael J.; Haeberlen, Andreas; Ives, Zachary G.; Krishnamurthy, Arvind; Lehr, William; Loo, Boon Thau; Mazieres, David; Nicolosi, Antonio; Smith, Jonathan M.; Stoica, Ion; van Renesse, Robbert; Walfish, Michael; Weatherspoon, Hakim; and Yoo, Christopher S., "The NEBULA Future Internet Architecture" (2013). *Faculty Scholarship at Penn Law*. 2034.
https://scholarship.law.upenn.edu/faculty_scholarship/2034

Authors

Tom Anderson, Ken Birman, Robert Broberg, Matthew Caesar, Douglas Comer, Chase Cotton, Michael J. Freed, Andreas Haeberlen, Zachary G. Ives, Arvind Krishnamurthy, William Lehr, Boon Thau Loo, David Mazieres, Antonio Nicolosi, Jonathan M. Smith, Ion Stoica, Robbert van Renesse, Michael Walfish, Hakim Weatherspoon, and Christopher S. Yoo

The NEBULA Future Internet Architecture*

Tom Anderson[1], Ken Birman[2], Robert Broberg[3], Matthew Caesar[4],
Douglas Comer[5], Chase Cotton[6], Michael J. Freedman[7], Andreas Haeberlen[8],
Zachary G. Ives[8], Arvind Krishnamurthy[1], William Lehr[9], Boon Thau Loo[8],
David Mazières[10], Antonio Nicolosi[11], Jonathan M. Smith[8], Ion Stoica[12],
Robbert van Renesse[2], Michael Walfish[13],
Hakim Weatherspoon[2], and Christopher S. Yoo[8]

[1] University of Washington
[2] Cornell University
[3] Cisco Systems
[4] University of Illinois
[5] Purdue University
[6] University of Delaware
[7] Princeton University
[8] University of Pennsylvania
[9] Massachusetts Institute of Technology
[10] Stanford University
[11] Stevens Institute of Technology
[12] University of California, Berkeley
[13] University of Texas, Austin

1 Introduction

The NEBULA Future Internet Architecture (FIA) project is focused on a future network that enables the vision of cloud computing [8,12] to be realized. With computation and storage moving to data centers, networking to these data centers must be several orders of magnitude more resilient for some applications to trust cloud computing and enable their move to the cloud.

An example application we envision is to use cloud computing as the basis for a personal health monitoring and advisory service. Sensors, data repositories, and interactive components could input parameters to the cloud – such as food consumed and exercise regimen followed. The challenge is in extending such a service to more advanced inputs and outputs, including real-time data communications to and from medical devices, such as continuous glucose monitors and insulin pumps. This application requires both high reliability and data privacy, or, seen from a network security perspective, all of the "CIA" security properties of Confidentiality, Integrity and Availability.

The NEBULA approach is organized into three architectural thrust areas: a reliable routing system and data center interconnect (NCore), a data plane that enables policy enforcement (NEBULA Data Plane, NDP), and a novel approach to control plane architecture (NEBULA Virtual and Extensible Networking Techniques, NVENT) that allows users to control the network configuration from the edge.

* Invited Paper.

A. Galis and A. Gavras (Eds.): FIA 2013, LNCS 7858, pp. 16–26, 2013.

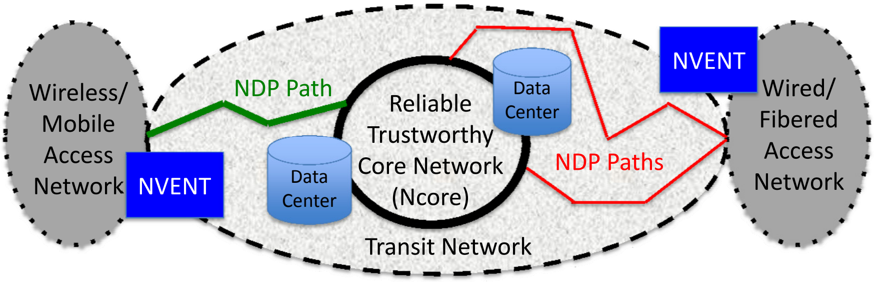

Fig. 1. The NEBULA Future Internet Architecture model

The NEBULA FIA project is characterized by three attributes. First, the architecture is *comprehensive*, in that it addresses a large set of complex and difficult problems. Second, the approach is *completely new*, and therefore in many aspects could not be "extended" or "composed from" any existing work: invention was required before integration could begin. Third, the comprehensive nature of the architecture demanded a *large team with a diversity of skill sets and approaches to sub-problems.*

2 NEBULA as a Network Architecture

A network architecture defines the structure of the network and its components. In NEBULA, we started from the basic position that cloud computing, where computation is performed on large network accessible data centers, would transform networking. Concerns about security properties such as confidentiality, integrity and availability would inhibit the use of cloud computing for new applications unless a new network architecture is designed.

To illustrate what the key challenges are, we discuss our example application – closed-loop control of blood glucose levels for an insulin-dependent diabetic – in a bit more detail. We can presume availability of some devices that already exist, and that the patient is equipped with some of these: a continuous glucose monitor, camera, exercise monitor and insulin pump. The cloud application would determine the current glucose level, monitor what was being eaten, monitor the exercise activity level, and make an insulin infusion recommendation. This application would have strict confidentiality requirements (it is *very* personal healthcare data), integrity requirements (incorrect dosages can be harmful) and availability requirements (a lack of network availability might, at the limit, cause physical harm).

NEBULA addresses these challenges with the three architectural thrust areas named in the introduction. Some basic decisions included the use of packet-switching and a network structure with hosts interconnected by a graph of links and store-and-forward routers. Paths are composed of sequences of links and routers, and will have performance dictated by the capacities of the components

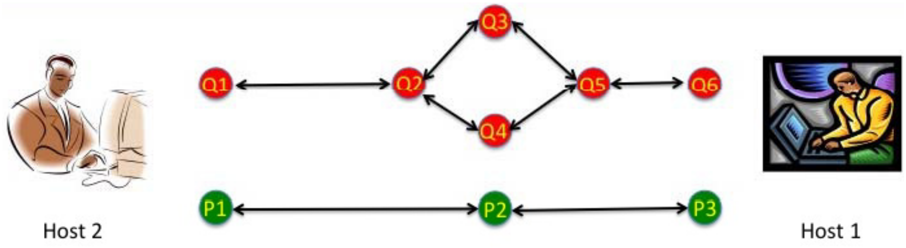

Fig. 2. Host 1 and Host 2 are connected by two physical paths

as well as their current workloads. These are basic properties NEBULA shares with the conventional Internet.

2.1 Today's Internet

However, the Internet makes some additional design decisions that influence its architecture. The first of these is that the routing algorithm attempts to find the *single* best path between two endpoints. The second of these is that the routers use a *best-effort* policy, allowing a packet to reach its destination if possible. Third, routers make *dynamic decisions* about the best path, allowing for unreliable links and routers under the assumption that the network graph is reasonably connected. Finally, the conventional Internet's *evolution occurs at the end-points* (hosts), rather than in the network.

For instance, consider the simple example network in Figure 2. There are nine routers (the red and green balls marked with Ps and Qs), and two interconnected hosts. These could represent either two edge nodes or – more relevant to the NEBULA vision – an edge host and a cloud data center. The red routers have a richer (but more complex) transit network, and the red and green access networks provide redundant access networking. In this network, the Internet would be unable to exploit the redundancy in the paths, and would be unable to enforce any policy (e.g., one related to health care data security and privacy). Diagnosis and policy enforcement would have to be performed at the endpoints.

2.2 NEBULA

NEBULA's support of cloud computing [8,2,12] and highly reliable applications [5] forces it to be different in several respects. First is the realization that data center networks [7] and their interconnection are both different in structure than a conventional Internet and require routers that are far more reliable than today's core routers [8]. Our NEBULA Core (NCore) of ultra-reliable Core routers interconnecting data centers are equipped with new fault-tolerant control software [2]. We exploit path diversity for resilience, and have originated new ways to detect/resist failures [24,18] and malicious attacks [11], including resilient interdomain protocols. In addition, we are examining implications of (and for) resilience solutions on policy and economics.

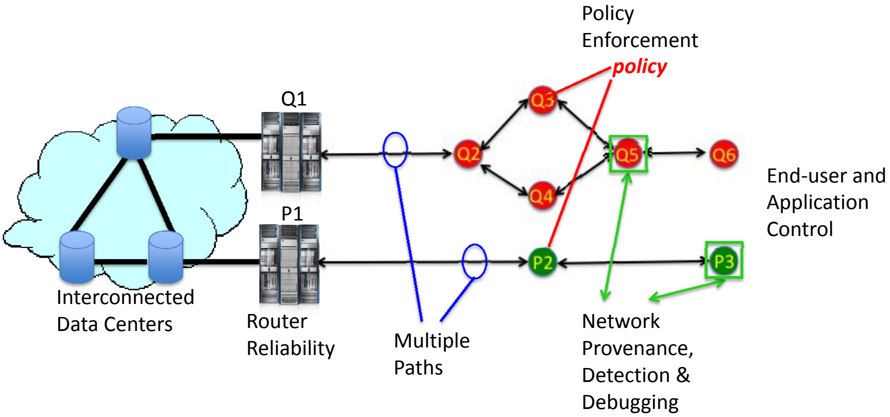

Fig. 3. A NEBULA perspective on the network from Figure 2

Consider Figure 3, a redrawn version of Figure 2 with NEBULA components added; Q3 and Q4 exhibit different security policies. The NEBULA network architecture is resilient: If any single router fails, the application is still left with a usable path it can exploit. Specifically, if Q1, P1, Q6, or P3 fail, access is preserved via P1, Q1, P3, and Q6, respectively. For policy enforcement, if Q3 or Q4's policy is incompatible with the application or the cloud service, either Q4 or Q3 can be used, or the green path can be exploited if it is policy-compliant. Using network provenance [42], failure detection [18,41] and network debugging [20], problems can be rapidly diagnosed and repaired, even while the services continue operating.

3 NEBULA Future Internet Architecture Integration

The first phase of NEBULA required invention of new approaches and elements in areas such as router architecture, data center networking, interdomain networking, network control, and network security. Further, it required work by scholars of regulatory policy and economics seeking to understand the implications of the cloud and NEBULA's changes to networking (including reliable routers and routes chosen with constraints set by flexible policies that reflect desires of parties such as users, cloud providers, internet service providers and web servers).

Ultra-reliable Future Internet Router Example (FIRE): This is the router component [2,10] of NCore. To ease sharing and implementation (see Section 4), an open source implementation is being used, and the team at the University of Delaware has been experimenting with RouteBricks [13] on two high-performance PCs (supplied by Cisco Systems) equipped with 10 Gbps cards.

Ultra-reliable data center or ISP network: This is the data center interconnect to the NCore routers; it is used to access data and compute cycles in the data centers. The interconnect must be ultra-reliable – for example, accesses

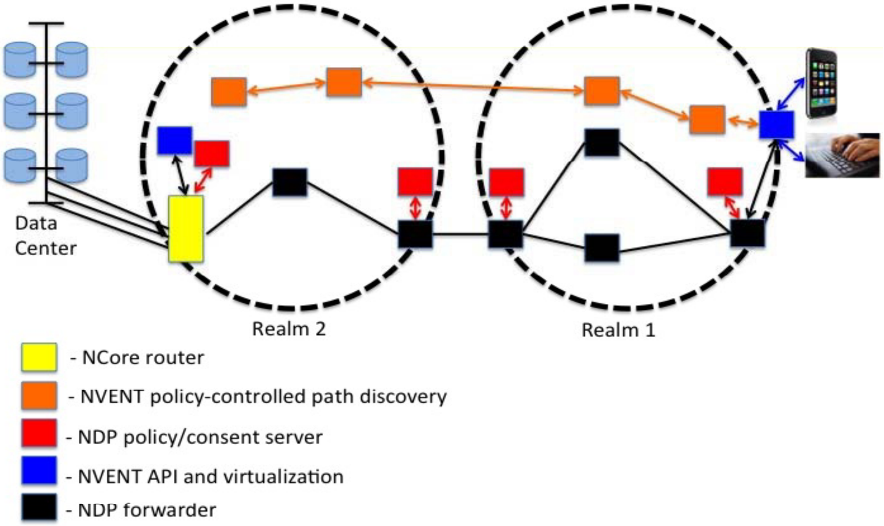
Data
Center
Realm 2
Realm 1
🟨 - NCore router
🟧 - NVENT policy-controlled path discovery
🟥 - NDP policy/consent server
🟦 - NVENT API and virtualization
⬛ - NDP forwarder

Fig. 4. Illustration of NEBULA Elements as an Integrated Architecture

to the stored glucose records and healthcare data must not fail, and there should be some end-to-end performance isolation [23]. It also makes sense to manage data flows to the NCore routers with a redundant [30] and load-balanced interconnect [33,34,7].

Ultra-reliable Interdomain service: This is primarily focused on the Nebula Data Plane (NDP) and several alternatives we have created [29,31] for its implementation. It is the primary locus for policy issues such as security, privacy, fault-tolerance, and more complex and specialized policies. This is a particular challenge, as the policies must be relevant in spite of a federated structure for NEBULA that we expect to emerge as a result of an emergent marketplace of offered network services.

Policy Control: This is primarily focused on NEBULA Virtual and Extensible Networking Technology (NVENT). It must define an API for service requests [32] – in particular, requests for a specific policy. Approaches [38] developed and matured during Phase I are being melded into an NVENT that provides a service abstraction consistent with our model of network support for the cloud. In particular, the Declarative Networking approach [28] will be used to form networks of policy compliant paths when needed by applications. Since much of the path information is cacheable, discovery can be asynchronous, and path lookup can be very quick. There are also extensive efforts in diagnosis [42,21,1], verification [37,25,35,36,3,41,19], detection [18] and debugging [20].

Economic and Policy Implications of NEBULA: This work [9,40,26,27] is primarily focused on the implications of the NEBULA project on economic and regulatory properties. The Phase I work examined economic implications of dramatically increased router reliability and some of the policy impacts of cloud computing. As we move towards an integrated architecture with policy enforcement, policy and regulatory issues abound. For example, TorIP [29] prevents visibility into packet destinations or content, and the economic and policy implications of this must be understood. Another example is the complex tussle between fine-grained policy enforcement and the desire by some for network neutrality.

A variety of interesting research questions have become apparent. We expect to resolve these as the integration efforts proceed. We briefly discuss two here.

First, the control plane must query policy servers for an ICING-based NDP [31], and the query model of declarative networking must be supported to create policy-compliant routes for the NEBULA control plane. ICING [31] is one alternative for NDP, and ICING enforces policy by having ISPs check the work of other ISPs, while TorIP [29] enforces policy by preventing ISPs from gathering the information needed to do effective policy discrimination. Transit as a Service (TaaS), if used as NDP, uses a third approach to policy enforcement, randomized test packets that cannot be distinguished from regular traffic.

Second, the application interface to specify policy has not been finalized, and the relationship between the policy-enforcing forwarding plane and the NCore routers remains somewhat in flux. We expect to investigate this via the integration of Serval [32] and Declarative Networking [28] that will comprise NVENT.

4 NEBULA Configuration and Operation

If we are to have highly trustworthy networking for the cloud, we must have highly trustworthy interdomain paths, and these in turn require highly reliable *intra*domain paths and services. To realize the latter, we are pursuing enhancements of RouteBricks (to build FIRE), the use of Quagga, and the addition of fault-tolerant protocols for intradomain networking.

Consider two scenarios where the NEBULA architecture's resilience and policy mechanisms would be applied. In Scenario 1, the U.S. Department of Defense wishes to acquire a high-assurance path over the Future Internet, and this path must traverse only trusted networks. Further, since adversaries might choose to attack the path, the network must not only be highly available (i.e., no single point of failure), it must also be DoS-resistant and tolerate Byzantine faults. In Scenario 2, outpatient medical monitoring is performed by software running at a data center; to support this scenario, both high assurance and predictable bandwidth are required.

A variety of goals can be abstracted from these scenarios – for instance, organizations need to be able to contract for end-to-end paths with explicit promises from each organization along the path to convey the traffic, no third party must be able to disrupt the traffic, and so forth. In our view, the model of the Internet being provided by multiple organizations will persist.

For all of this to work, that is, for a user or organization to specify a policy for what it wants from the network, and for a Future Internet Service Provider to specify its policy of what it is willing to provide, the need arises for protocols for the following components: (1) a “name service,” by which users locate the services they need; (2) "path advertisements," through which constituent networks gather and convey what each is willing to provide; (3) "assured path provisioning," whereby network resources are sought and bound for use in a given communication; (4) "verifiable data forwarding," whereby traffic is sent on network routes that verifiably comply with the policies of the assured path; and (5) "fault diagnostics," which detects and/or identifies faulty network components.

4.1 Policy Configuration

There is clearly a very large number of possible policies, and these policies come from multiple sources – for example, end-users, their organizations and the ISPs bearing their traffic in the middle of the network. Some applications need path control, some need resource reservations, some need failover guarantees, etc. This requires work at the API level, in NVENT (using Serval [32] as the basis for this API) to determine how a client can ask for certain properties. A second issue is at the protocol level in NVENT, where mechanisms for an ISP advertising its willingness to offer certain properties must be determined.

A policy server will have zero or more policies. The default policy is to drop traffic, sometimes called "deny by default". Policies are assumed to be dynamic (changeable) but we assume they are changed infrequently, and thus are cacheable. In our initial architecture, we expect that users and prototype applications will want policies that are easy to state; for instance, a policy indicating HIPAA compliance could simply be stated as `HIPAA=yes`. A policy server's policies can be queried by clients and consent servers; a path is constructed from consenting servers.

In choosing paths, the policy logic must not only know which paths are permitted, it must also know which paths are available. This kind of knowledge requires detection support from the network [18].

Some of our work is in the direction of removing the ability for ISPs to discriminate against certain types of traffic (e.g., TorIP), although that might also be accomplished by other means, e.g., regulation or testing. ICING assumes the ability of ISPs to veto paths, which means either delegation of path-granting authority or extra RTTs. Depending on some of those choices, the policy options are somewhat different.

4.2 Path Setup

A user or application specifies policy requirements for the path they wish to use – for instance `NEBULA_PATH=HIPAA:AES256:T_BOUND_5`. The application specifies a destination or service (see also Section 4.4).

For an ICING-based [31] NDP, when this specification is received, the system checks a cache for a cached compliant path to the destination or service. If such a path is available, try to obtain consent to use the path, perhaps with cached proofs of consent if obtaining consent is expensive. If nothing is cached, or there is no consent for a cached path, the system would iterate requests for consent to consent servers. The end result is that NEBULA will either establish and cache a path, or will fail with an error. For TorIP [29]/TaaS, the assumption is that advertisements will be long-lived; there is no path-veto, so advertisements (in contrast to ICING) are simple and composable, albeit less expressive.

4.3 Forwarding

NEBULA users (either senders, receivers, or service providers) can require that specific network resources be bound to a network path (cf. e.g. Scenario 2 above). To verify compliance, packets can carry secure "markings" of consent, as well as a secure reference to the resources allocated to that connection. This marking strategy might be implemented via Icing's cryptographic "proofs of consent" and "proofs of provenance", or via the cryptographic sealing implied by Onion Routing in TorIP. Below we outline the key steps for the case of verifiable data forwarding in Icing.

Senders mark their packets using the cryptographic tokens included in the proofs of consent they obtained when the connection is established. When processing an incoming packet, an NDP edge router checks whether from the packet's marks it can evince that the packet is "permitted" (it carries proper proof of consent) and "travelled right" (it carries proper proof of provenance). The last check requires that previous routers had updated the "marks" on the packet whenever an ISP ("realm") boundary was crossed. Thus, before forwarding the packet to the next node on the path, an NDP edge router "blesses" the packet (or peels off an onion layer in the case of TorIP).

4.4 Naming

In Serval [32], Service IDs are resolved via a Service Abstraction Layer (SAL). Both TorIP [29] and ICING [31], alternatives for NDP have appropriate solutions for mapping human-readable names into network-usable information. In TorIP a *name server* resolves a name (e.g., `google.com`) to a set of (ISP,ID) pairs. The ID identifies a mailbox where a client can rendezvous with a server or service. A client finds a path connecting ISP-advertised triples. In ICING, DNS is augmented by policy enforcement, by forcing paths to have consenting elements. For example, DNS client resolution of `www.foo.com` requires consenting paths to servers for ".", ".com", "foo.com", etc.

Proofs of Consent (PoCs) are cacheable by clients, so in the common case, only resolving the most specific name would require interaction with consent servers.

5 Conclusions

NEBULA is a Future Internet Architecture intended to provide secure and resilient networking to support present and future applications of cloud computing. At the time of writing, the work on NEBULA is still ongoing, but we have made significant progress in a number of essential areas. NEBULA is a comprehensive architecture, and a number of novel technologies have already been developed for it; however, due to space constraints, we can only give a brief overview of the key building blocks here. For details, we refer the interested reader to the papers we cite below.

Acknowledgments. This work is supported by the U.S. National Science Foundation.

References

1. Aditya, P., Zhao, M., Lin, Y., Haeberlen, A., Druschel, P., Maggs, B., Wishon, B.: Reliable client accounting for hybrid content-distribution networks. In: Proc. NSDI (April 2012)
2. Agapi, A., Birman, K., Broberg, R., Cotton, C., Kielmann, T., Millnert, M., Payne, R., Surton, R., van Renesse, R.: Routers for the Cloud: Can the Internet achieve 5-nines availability? IEEE Internet Computing 15(5), 72–77 (2011)
3. Arye, M., Nordström, E., Kiefer, R., Rexford, J., Freedman, M.J.: A provably-correct protocol for seamless communication with mobile, multi-homed hosts. Technical Report 1203.4042v1, arXiv (March 2012)
4. Birman, K.P., Huang, Q., Freedman, D.: Overcoming the "D" in CAP: Using Isis2 to build locally responsive cloud services. IEEE Internet Computing 12, 50–58 (2012)
5. Birman, K.P.: Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services. Springer (2012)
6. Birman, K.P., Ganesh, L., van Renesse, R.: Running smart grid control software on cloud computing architectures. In: Proc. Workshop on Computational Needs for the Next Generation Electric Grid (April 2011)
7. Bodík, P., Menache, I., Chowdhury, M., Mani, P., Maltz, D.A., Stoica, I.: Surviving failures in bandwidth-constrained datacenters. In: Proc. SIGCOMM (2012)
8. Broberg, R., Agapi, A., Birman, K., Comer, D., Cotton, C., Kielmann, T., Lehr, W., van Renesse, R., Surton, R., Smith, J.M.: Clouds, cable and connectivity: Future Internets and router requirements. In: Proc. Cable Connection Spring Technical Conference (June 2011)
9. Clark, D., Lehr, W., Bauer, S.: Interconnection in the internet: the policy challenge. In: Proc. 39th Research Conference on Communication, Information and Internet Policy (September 2013)
10. Comer, D., Javed, S.: Applying open resilient cluster management (orcm) to a multi-chassis core router. In: Proc. ISCA International Conference on Computers and Their Applications (March 2012)
11. Comer, D., Suingh, P., Vasudevan, S.: Towards a practical and effective BGP defense system. In: Proc. ICICS (January 2012)
12. Comer, D.: A future Internet architecture that supports Cloud Computing. In: Proc. 6th International Conference on Future Internet Technologies (June 2011)

13. Dobrescu, M., Egi, N., Argyraki, K., Chun, B.G., Fall, K., Iannaccone, G., Knies, A., Manesh, M., Ratnasamy, S.: RouteBricks: exploiting parallelism to scale software routers. In: Proc. SOSP (2009)
14. Foster, N., Freedman, M.J., Harrison, R., Monsanto, C., Reitblatt, M., Rexford, J., Story, A., Walker, D.: Language abstractions for software-defined networks. In: Proc. Workshop on Lang. for Distrib. Algorithms (2012)
15. Foster, N., Harrison, R., Freedman, M.J., Monsanto, C., Rexford, J., Story, A., Walker, D.: Frenetic: A network programming language. In: Proc. ICFP (2011)
16. Freedman, D., Marian, T., Lee, J., Birman, K., Weatherspoon, H., Xu, C.: Instrumentation for exact packet timings in networks. In: Proc. Instrumentation and Measurement Technology Conference (May 2011)
17. Ghodsi, A., Sekar, V., Zaharia, M., Stoica, I.: Multi-resource fair queueing for packet processing. In: Proc. SIGCOMM (2012)
18. Gupta, T., Leners, J.B., Aguilera, M.K., Walfish, M.: Exposing network failures to end-host applications for improved availability. In: Proc. NSDI (April 2013)
19. Gurney, A.J.T., Haeberlen, A., Zhou, W., Sherr, M., Loo, B.T.: Having your cake and eating it too: Routing security with privacy protections. In: Proc. HotNets (November 2011)
20. Handigol, N., Heller, B., Jeyakumar, V., Maziéres, D., McKeown, N.: Where is the debugger for my Software-Defined Network? In: Proc. HotSDN (2012)
21. Hong, C.Y., Caesar, M., Duffield, N., Wang, J.: Tiresias: Online anomaly detection for hierarchical operational network data. In: Proc. ICDCS (2012)
22. Hong, C.Y., Caesar, M., Godfrey, P.B.: Finishing flows quickly with preemptive scheduling. In: Proc. SIGCOMM (2012)
23. Jeyakumar, V., Alizadeh, M., Mazières, D., Prabhakar, B., Kim, C.: EyeQ: Practical network performance isolation for the multi-tenant Cloud. In: Proc. HotCloud (2012)
24. Khurshid, A., Kiyak, F., Caesar, M.: Improving robustness of DNS to software vulnerabilities. In: Proc. ACSAC (2011)
25. Khurshid, A., Zhou, W., Caesar, M., Godfrey, P.B.: VeriFlow: Verifying network-wide invariants in real time. In: Proc. HotSDN (2012)
26. Lehr, W.: Measuring the Internet: The data challenge. OECD Digital Economy Papers, No. 194. OECD Publishing (2012)
27. Lehr, W., Clark, D., Bauer, S.: Measuring Internet performance when broadband is the new PSTN. Paper prepared for the “End of PSTN” Workshop at the University of Pennsylvania (May 2012)
28. Liu, C., Ren, L., Loo, B.T., Mao, Y., Basu, P.: Cologne: A declarative distributed constraint optimization platform. Proc. VLDB Endowm. 5(8), 752–763 (2012)
29. Liu, V., Han, S., Krishnamurthy, A., Anderson, T.: Tor instead of IP. In: Proc. HotNets (2011)
30. Liu, V., Halperin, D., Krishnamurthy, A., Anderson, T.: F10: A fault-tolerant engineered network. In: Proc. NSDI (April 2013)
31. Naous, J., Walfish, M., Nicolosi, A., Mazières, D., Miller, M., Seehra, A.: Verifying and enforcing network paths with ICING. In: Proc. CoNEXT (2011)
32. Nordström, E., Shue, D., Gopalan, P., Kiefer, R., Arye, M., Ko, S.Y., Rexford, J., Freedman, M.J.: Serval: An end-host stack for service-centric networking. In: Proc. NSDI (2012)
33. Popa, L., Krishnamurthy, A., Ratnasamy, S., Stoica, I.: FairCloud: Sharing the network in cloud computing. In: Proc. HotNets (2011)
34. Popa, L., Kumar, G., Chowdhury, M., Krishnamurthy, A., Ratnasamy, S., Stoica, I.: FairCloud: sharing the network in cloud computing. In: Proc. SIGCOMM (2012)

35. Setty, S., McPherson, R., Blumberg, A.J., Walfish, M.: Making argument systems for outsourced computation practical (sometimes). In: Proc. NDSS (February 2012)
36. Setty, S., Vu, V., Panpalia, N., Braun, B., Blumberg, A.J., Walfish, M.: Taking proof-based verified computation a few steps closer to practicality. In: Proc. USENIX Security (2012)
37. Wang, A., Jia, L., Zhou, W., Ren, Y., Loo, B.T., Rexford, J., Nigam, V., Scedrov, A., Talcott, C.: FSR: Formal analysis and implementation toolkit for safe interdomain routing. IEEE/ACM Transactions on Networking (ToN) 20(6), 1814–1827 (2012)
38. Wang, A., Talcott, C., Gurney, A.J., Loo, B.T., Scedrov, A.: Brief announcement: A calculus of policy-based routing systems. In: Proc. PODC (July 2012)
39. Williams, D., Jamjoom, H., Weatherspoon, H.: The Xen-Blanket: Virtualize once, run everywhere. In: Proc. EuroSys (2012)
40. Yoo, C.S.: Cloud computing: Architecturand and policy implications. Review of Industrial Economics 38(4), 405–421 (2011)
41. Zhao, M., Zhou, W., Gurney, A.J.T., Haeberlen, A., Sherr, M., Loo, B.T.: Private and verifiable interdomain routing decisions. In: Proc. SIGCOMM (August 2012)
42. Zhou, W., Fei, Q., Narayan, A., Haeberlen, A., Loo, B.T., Sherr, M.: Secure network provenance. In: Proc. SOSP (October 2011)