CLOUDS, CABLE AND CONNECTIVITY: FUTURE INTERNETS AND ROUTER REQUIREMENTS

Robert M. Broberg (Cisco), Andrei Agapi (Cisco), Ken Birman (Cornell), Douglas Comer (Purdue), Chase Cotton (University of Delaware), Thilo Kielmann (Vrije Universiteit), Bill Lehr (MIT), Robbert VanRenesse (Cornell), Robert Surton (Cornell), Jonathan M. Smith (University of Pennsylvania)

Abstract

The "computing utility" vision of cloud computing posits a future Internet that offers a universal infrastructure capable of providing on-demand access to computing, storage, and communication services. Clouds will support a diverse range of user/usage contexts, ranging from the delivery of advanced television and content to support for corporate enterprise networks, from controlling smart grids to remote control of an insulin pump.

This paper discusses the research program for a future Internet that is being undertaken by the Nebula project, with support from the National Science Foundation, and in collaboration with Cisco in its on-going efforts to rethink the software architecture for large multi-processor router platforms. The Nebula architecture is comprised of three elements: NCore for tying together cloud data centers and core routing infrastructure, NVent for implementing a flexible and extensible control plane, and NDP for fine-grained, policy-based end-to-end control of network flows. Herein, we concentrate our discussion on NCore. We also briefly highlight some of the non-technical policy and business challenges posed by migrating to this new, more capable and robust architecture.

1. INTRODUCTION

Cloud computing means many things to many people. We adopt a computing utility vision [Fano65] of the cloud as a universal infrastructure capable of providing on-demand access to computing, storage, and communication services over the Internet to support a wide array of user-needs and applications. These may range in diversity from entertainment television delivery, to support for public safety emergency calling, or even the ability to remotely control a diabetic's insulin pump. This vision is analogous to the collective relationship between the data centers constituting an electricity grid of power generating facilities, the long-haul transmission grid manifest as a core Internet routing infrastructure, and the local distribution facilities that provide access networks. Supplying electric power to businesses and consumers, this ensemble is equally responsible for maintaining flexibility towards all different respective requirements regarding performance, security, and end-user control.

This computing utility vision posits the existence of virtualization software, capable of supporting the illusion of dedicated capacity while sharing computing and storage resources located across multiple providers. To end-users, this vision implies a shift of intelligence, data, and services into the network "cloud." An increase in energy and administrative costs, the growth of data-intensive applications, and a proliferation of new usage contexts (including thin clients, mobile computing, and machine-to-machine applications) are all contributing to an inevitable adaptation to some form of network-centric, cloud-based resource sharing. In light of a surge in video and other rich media traffic precipitating the search for new content delivery strategies, and their critical role in providing last-mile broadband, we believe that traditional broadcast/cable companies should be lead players in guiding the design and migration to a more capable, flexible, and secure Internet.

The Nebula research project (see http://nebula.cis.upenn.edu) supported by the United States National Science Foundation (NSF) and complemented by Cisco's on-going research effort (see http://r3.cis.upenn.edu), is exploring an Internet architecture to foster this cloud computing vision. The Nebula architecture will embody three components: a high-speed core that interconnects data centers and enables direct transfer among them, a set of wired and wireless access networks that provide connectivity to individuals and enterprises, and a transit layer that allows an individual to connect to the nearest data center over a path with guaranteed properties, such as security. Early goals include continuous availability for routing components such as BGP, even when the processor on which BGP is running fails; peers will be completely insulated from failure/restart events, avoiding route flaps, black holes and other transients seen when BGP fail-over occurs on today's core network routers. As our effort moves forward, we believe we can do even more. The team is exploring novel options for securing routes and protecting against attacks, for creating new kinds of router-hosted services, and broadly, for transforming the modern router into a better partner for the evolving cloud. The development of this new architecture is consciously motivated to address real-world deployment issues, such as compatibility with regulatory policies, and scalable deployment within today's evolving industry value chain.

This paper is organized into the following sections. Section 2 presents an overview of the Nebula architecture. In Section 3 we examine some of the technical and non-technical challenges posed by this vision of cloud computing. Section 4 concludes.

2.0 A SECURE ARCHITECTURE FOR CLOUD COMPUTING

Traditional Internet services [Comer06] are built on a best effort packet delivery service. What is ultimately desired is the ability to deliver the best features of today’s Internet architecture, combined with new architectural features to support applications with *beyond best-effort* real-time and policy requirements. For stored or dynamic content, the HTTP/TCP/IP model seems adequate, but for an expanding range of service offerings, the conventional architecture could be much improved. Consider, for example, the highly reliable video delivery services offered by Cable providers. This traditional “Cable TV” service is still the basis for many service subscriptions, and continues to evolve in fidelity as end-user equipment evolves (e.g., HDTV). An ideal Internet solution would allow for this traffic to be treated in isolation throughout the network, and for the resources necessary for subscriber fidelity to be acquired and maintained as needed. Further, the reliability would be such that the illusion of classic coaxial service could be maintained, while at the same time capitalizing on the operational advantages of a universal IP infrastructure. Conversely, multiple internal IP infrastructures are often used today. This split acutely illustrates the challenge for the future: a universal infrastructure that affords extensibility for new services, policy enforcement and ultra-high availability. To meet the demands of scalable performance growth, as well as the variety of security and trust requirements implicit in managing healthcare applications, smart grids, delivery of entertainment television or bulk data delivery, new architectural elements are clearly needed.

2.1 The Nebula Project

Nebula is a project that was founded in 2010, supported by both Cisco and NSF funding which was awarded to a number of institutions. It is an outgrowth of an earlier research effort, Router Reliability Research (R3), an investigation initiated by some of the authors into new software architectures for large-scale multiprocessing core routing systems, which will be discussed later in this paper.

Nebula is intended to encourage the *cloud-based future Internet*. In doing so, it addresses many challenges for which solutions would be useful even in today's world. Nebula is based on three architectural elements:

- Nebula Core (NCore), that provides a highly-connected graph of ultra-reliable routers to interconnect data centers;
- Nebula Virtualizable and Extensible Network Techniques (NVENT), that provides an extensible future control plane with more transparency and control for applications; and
- Nebula Data Plane (NDP), which provides robust fine-grained policy enforcement and subsumes many roles that are now filled by a zoo of middleboxes.

Figure 1 is a conceptual illustration of the Nebula architecture with these three elements in place:

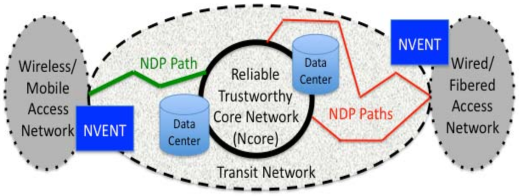

Figure 1: The Nebula Network Architecture: NCore, NVENT and NDP

We will give a very brief description of the three components here; more information is available on the Nebula project web site, including a white paper and project overview slides.

The NDP [Naous10, Popa09] is a new packet format incorporating cryptographic tokens that demonstrate each "realm" (roughly equivalent to an autonomous system, but realm borders are defined by public keys). Each realm must consent to carry the traffic under some specified policy, e.g., HIPAA-compliance for health care data mentioned in the introduction. Two things required for a forwarder to forward the packet are a proof of consent (that the packet would be passed) and a proof of path (that the path was taken). These cryptographic tokens require 42 bytes per realm at present – based on our analysis, about 200 bytes will be required per packet. We are trying at this stage to resist premature optimization, as the major goal is robust policy enforcement. Some preliminary experiments with an FPGA-based prototype have shown roughly 4 Gbps performance, so high performance is achievable.

The purpose of the NVENT control plane is to furnish an application programming interface (API) that can specify attributes for [Birman03], and to acquire [Loo05], paths through the network. For example, Figure 1 illustrates that an NVENT system for an attached access network is providing two paths using NDP – this might satisfy a reliability requirement (e.g., graceful degradation of service), an increase in capacity (through network striping [Traw95]) or some other desired property. This form of reliability requirement might exist for a medical application utilized by the potential insulin pump mentioned in Section 1, where a continuous glucose monitor might sample every five minutes or so. In such a situation, the data rate would remain fairly low, but the reliability must be extremely high. NVENT nodes also fill the role of policy and consent servers for NDP.

The NCore architecture uses striped connections between a data center and core router, and then again stripes amongst a set of core routers. Figure 2 illustrates:

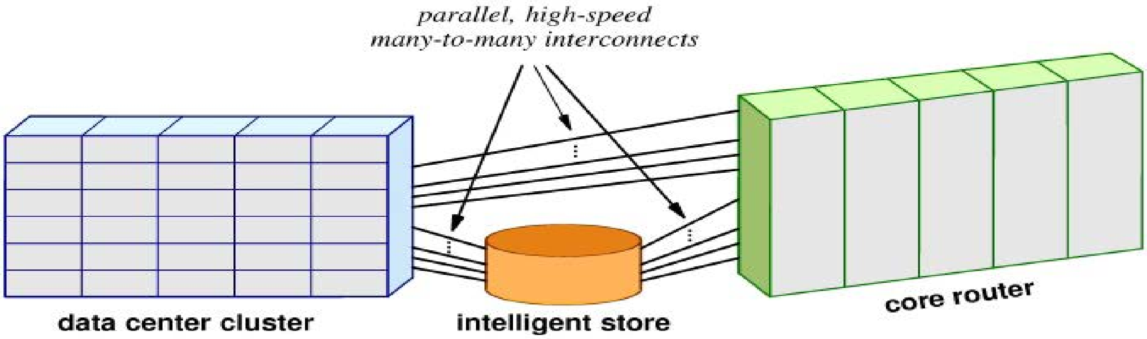

Figure 2: NCore has a rich connection graph for high reliability and performance

In our view, the rich graph connectivity depicted offers many benefits, amongst which are the capacity to resist denial of service (DoS) attacks, resistance to failures and the ability to load-balance or physically isolate nodes for reliability. The use of network striping techniques allows aggregation of links for higher capacities; this might permit, for example, the migration of virtual machines (VMs) for load-balancing, latency or to “follow the sun” (*i.e.,* open stock exchanges).

A key part of NCore is the use of multi-chassis core routers. These can be viewed as large-scale multiprocessors or cluster computers, and can be transformed into ultra-reliable routing systems with the type of software architecture enhancements discussed in the next section.

2.2 New Software Architecture for Cloud Routers

The Cisco research effort is seeking an innovative response to the heightened demand for reliable content distribution over the Internet, while simultaneously encompassing legacy systems (through emergency services such as 911/telephony, with five nines expectation). The purpose is to maintain significant existing investment to date, constituted currently by delivery over highly custom embedded machines, in conjunction with a migration onto a newly created environment. Relying on the isolation of applications from the fault tolerant infrastructure, the result is a fully distributed fault tolerant system, which has also been designed to provide a platform for the smooth integration of future system developments.

The complexity and performance demands of the modern Internet have made core routers into large-scale parallel processing devices. Each line card can be realistically viewed as a high performance processor that is primarily tasked with managing multiple high-speed I/O streams (*i.e.*, the packets). Much of the line card’s hardware functionality is devoted to offloading and accelerating packet processing activities such as flow identification and forwarding, and may include features such as access control, rate policing and queue management schemes. Topologically, the line card manages a link layer interface to another line card, a WAN connection, or a host interface. Line cards are interconnected through an internal *switching fabric*. The fabric is a specialized router-internal network that optimizes communication amongst the line cards for high throughput and minimal collisions. Control processors for the router aggregate and share adjacency and policy information across the switching elements, as determined by routing protocols (e.g. BGP) and user-applied policies.

A modern core router can be configured with hundreds of line cards, distributed across multiple chassis interconnected by fiber optic links. While often (naively) thought of as a processor with some line cards attached to its I/O bus, large router configurations are necessary to minimize hop counts, consolidate management and minimize cost, energy and real estate footprint. The analogies to scalable cluster computing are very strong.

An important issue with scale is the likelihood of failed components, which for a given constant component reliability increases with scale. Since the incentives to scale configurations are compelling, failures become more of an issue and motivate the application of fault tolerant computing techniques to modern routers.

Particular goals include an overall “always-on” model that allows for multiple concurrent software versions, live upgrades [Hicks05] and robust failover for processes. While line cards connecting customer equipment with computer host interfaces cannot recover all state, fault tolerant protocols to interconnect core routers (such as a fault tolerant BGP based on a fault tolerant TCP protocol) can overcome many intermittent failures. The live upgrade / versioning issue can be addressed with virtualization technologies similar to those which enable cloud computing. Fault-tolerant storage of state in the router (*e.g.,* information contained in tables) can be made robust with new data structures such as distributed hash tables across multiple independent compute elements. More generally, replication to achieve redundancy is a very powerful strategy, and can be used to provide the software equivalent of a "hot spare" capability to the router control software. Each of these techniques is being pursued with the overall goal of downtime, for one or more routers in a "routing complex."

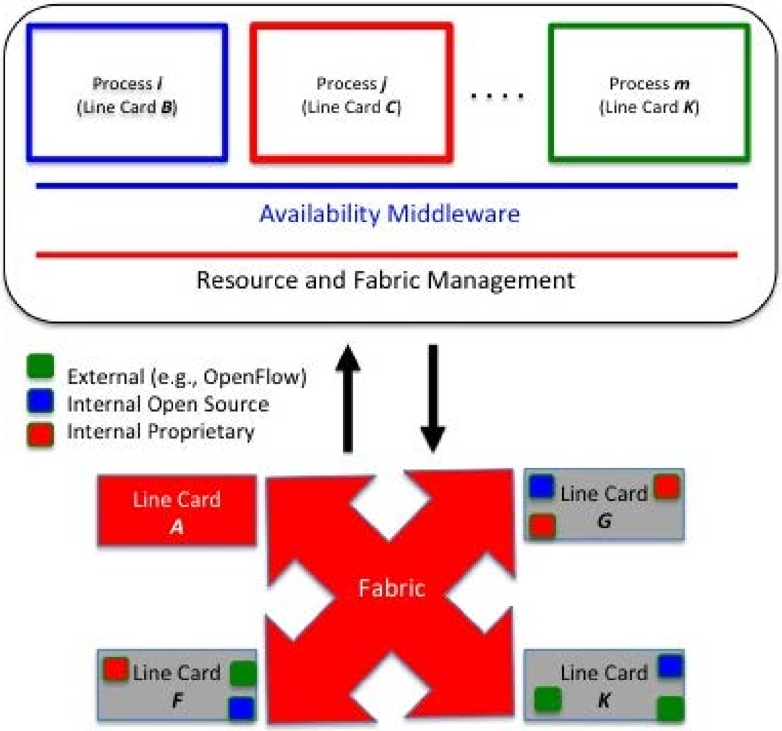
Process *i* (Line Card *B*)
Process *j* (Line Card *C*)
....
Process *m* (Line Card *K*)
Availability Middleware
Resource and Fabric Management
🟩 External (e.g., OpenFlow)
🟦 Internal Open Source
🟥 Internal Proprietary
Line Card *A*
Fabric
Line Card *G*
Line Card *F*
Line Card *K*

Figure 3: An advanced software architecture leverages core router hardware

Figure 3 illustrates a logical division of software layers (in the top box), with multiple applications instantiated as processes operating on line cards, and its further instantiation shown below on a set of line cards. The colors indicate the type of software, with a green box indicating external software (which could be loaded while the router is in operation), a blue box indicating Nebula/R3 open source, and a red box indicating vendor proprietary software. The virtualization [Birman03] in this instance protects not only resources but isolates software constrained by different intellectual property regimes. It should be clear that specialized functionality (e.g., per-customer services

[Alexander98]) is only one business model direction enabled by this approach.

3. IMPLEMENTATION CHALLENGES FOR NEBULA/CLOUD VISION

In the following sub-sections we consider some of the technical and non-technical challenges to developing and migrating to new architecture.

3.1.0 Technical Implementation Challenges: Fault Tolerant State Store (FTSS)

Some of the crucial steps in implementing this Nebula/Cisco vision for a cloud-enabled network involve actualizing fault-tolerant protocols and fault-tolerant state storage in the router. As with many other aspects of our system, we accomplish this by using the current Internet fabric and improving its resilience, while preserving backward interoperability.

Intra- and inter-domain routing protocols represent typical classes of latency-sensitive, critical, "online" applications. Protocols such as BGP compose the foundation for a functional Internet, which therefore demand quick failover, failure resilience, and speed. Yet BGP, as it is currently deployed, exhibits serious resilience and availability shortcomings. For example, complete recovery from BGP process crashes on routers is basically now done by remote synchronization of full Internet tables from potentially distant peers, which is a sub-optimal feature. However, we believe deficiencies of this nature to be inherent to the traditional resilience models executed in practice, perhaps most significantly 1+1 redundancy, rather than to the protocol itself.

Incrementally improving certain BGP characteristics, such as resilience, stability and Mean Time To Recovery (MTTR), while preserving the current protocol and requiring only minimal modifications to its existing codebase, is a highly desirable scenario to both Internet carriers and equipment vendors. To reiterate the relevance of the cloud model, we are able to achieve our goal largely by building on a paradigm that supports the separation of data from processing. In addition, we realize fault tolerance through a focus on safeguarding the application state data.
The major architectural element that allows for this approach is called FTSS (Fault Tolerant State Store), a distributed, resilient, high-performance, in-memory data store running across router components. It is designed to rapidly store, replicate, and retrieve arbitrarily structured application state. Our FTSS prototype is essentially a performance-optimized 1-hop distributed hash table (DHT); being malleable, it sustains failures, additions and replacements of underlying storage elements, while also providing automatic load balancing. The purpose of this format is to prevent overall unavailability for any subset of stored application data, in the event of multiple failures across any K storage elements. The store itself is specifically optimized for write intensive operations, both latency- and throughput-wise. It is tailored to scale well, takes advantage of underlying testbed size and characteristics, while adapting its communication and routing optimizations.

An example of the use value for such a fast store is the need to checkpoint online applications, expected to be highly responsive while operating at high throughputs and low latencies, and allowing for fast recovery as processes fail. We have successfully used FTSS to create a resilient BGP, with no modifications to the protocol itself, and minimal adjustments to off-the-shelf BGP implementations for protocol machinery. This method permits the resulting system to be deployed or migrated on computer clusters, Internet routers (*e.g. as prototyped on* Cisco CRS), or a combination thereof.

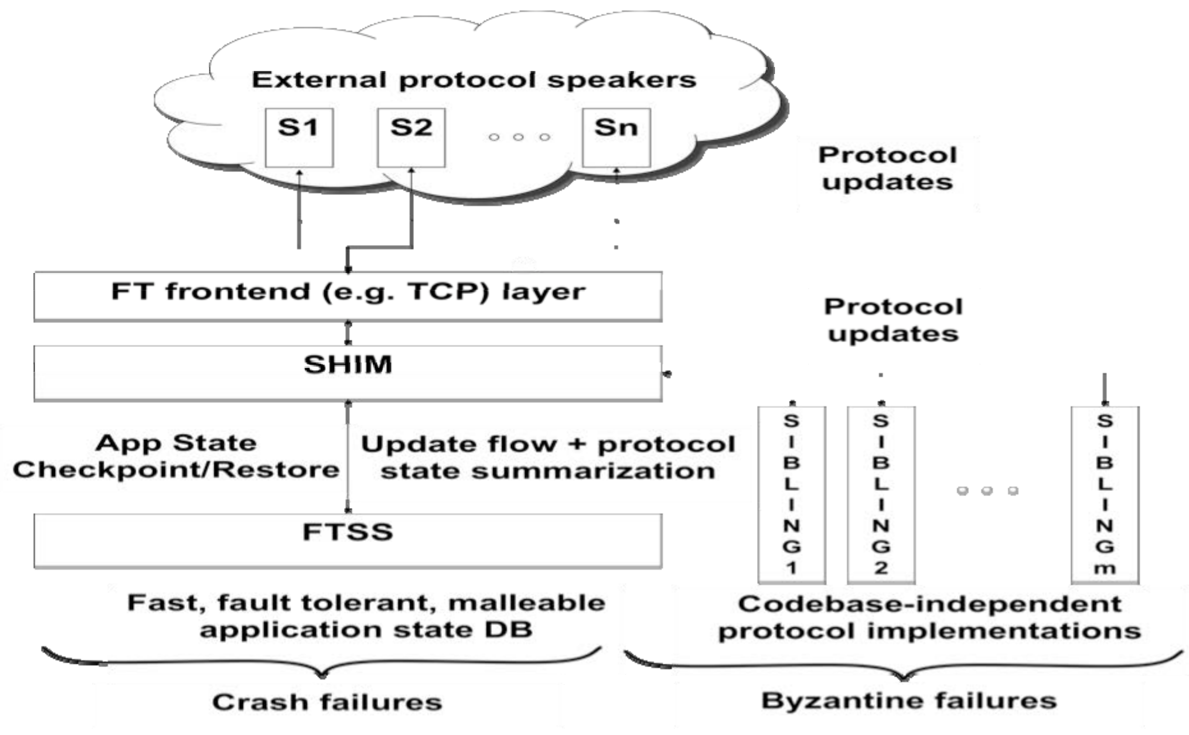
External protocol speakers
S1
S2
Sn
Protocol updates
FT frontend (e.g. TCP) layer
Protocol updates
SHIM
App State Checkpoint/Restore
Update flow + protocol state summarization
FTSS
SIBLING1
SIBLING2
SIBLINGm
Fast, fault tolerant, malleable application state DB
Codebase-independent protocol implementations
Crash failures
Byzantine failures

Figure 4: Resilience-enabling online processes such as BGP

In Figure 4, we outline FTSS use for fast, K-redundant state replication. Byzantine failures (*e.g.* implementation bugs) are handled by separated modules, such as through feeding independent implementations of the same process state machine in parallel. The state stored inside FTSS, received from a protocol-specific module called the *shim*, should therefore be as “raw” as possible so as to avoid any contamination from processing in the state machine. The shim delays update TCP ACKs to the remote side until replication has occurred; therefore, in any failure scenario, updates have either been persisted or will be retransmitted. In this fashion, correct K-redundancy against failures can, in principle, be achieved with only one running copy of each of the involved process types. While more than one process copy can still clearly be deployed (*e.g.* to maintain “hot spare” processes), there is never the need to maintain up to K copies of each process type, and the latter scheme displays much more efficient resource usage.

3.1.1 Technical Implementation Challenges: TCP with Session Recovery (TCP-R)

__To return to the diagram in Figure 4, we will now discuss the "fault-tolerant front end" (i.e. TCP) layer situated between the shim and external protocol speakers, i.e. BGP, which is designed to address further vulnerabilities inherent in current BGP deployment concerning router availability and recovery. We call this layer TCP-R, for "TCP with session recovery," and it is structured to target the following issues.

BGP servers communicate with their remote peers via TCP sessions, and if these sessions happen to be disrupted (perhaps in the case of a fail-over within our fault-tolerant BGP implementation), those remote BGP servers may react in unexpected ways that can harm availability, such as by routing around a failed router. Slow resynchronization can subsequently occur when the sessions are re-established, causing routing instability delays of up to several minutes for core Internet routers. These routers operate under very high loads, and may well have tens or hundreds of remote BGP peers. During these periods, problems such as route flaps, "black holes" or routing "loops" can arise.

As a preventative measure, BGP servers often implement a "graceful restart" to handle such events. The value of a restarted BGP server's initial state is empty, but it restarts on a hardware router that has maintained active hardware routing tables. Proceeding under the assumption that those tables are mostly correct, the BGP itself recovers, but the routing tables are left in place and neighboring routers may continue to send traffic. The intended goal is for BGP, which is functioning based on increasingly stale routing state, to resume active control quickly enough for this period of inconsistency to be brief. However, this method has not proven to effectively eliminate the problems visible in ungraceful restarts.

The solution we have developed to approach this involves masking a BGP fail/restart from all neighboring routers. We basically graft a new TCP connection onto an old one, in such a way that this event is made invisible to the remote endpoints holding the old TCP connection. Technically, it is possible to compare this to the behavior of a standard network address translating (NAT) box, which effectively grafts a TCP endpoint that believes itself to be connected to, i.e. server X on port P, while in reality those values are different. TCP-R achieves similar results through a TCP session's internal sequence numbering, which is used to identify bytes within TCP's sliding window. If BGP fails over, the new server restarts with the same state prior to its crash, which has been stored in FTSS. It restarts in a state prepared to finish any interrupted send of a BGP update, and ready to read the next byte in sequence of input from a peer's update. Because its per-flow state is only a few bytes of session-related data, TCP-R can handle tens of thousands of concurrent flows. The system that comprises FTSS, FT-BGP, and TCP-R, allows for router failures to be completely concealed from remote peers, and maintain the appearance of always-on, non-stop routing.

As for our technical requirements, no changes were made to the O/S kernel or the TCP stack used, other than recompiling the kernel with a standard Linux packet-filter package. Regarding speed, we are in the process of measuring numbers for the time delay required for BGP's migration from node to node, but best-scenario results currently suggest they are in the tens of milliseconds.

3.1.2 Technical Implementation Challenges: System IS-IS (SIS-IS)

Another issue we have been compelled to address, relevant to constructing a large distributed system, is that of automatically organizing and configuring a system that will comprise many processes and process types spread across many execution elements. Furthermore, these processes must be able to quickly and reliably find each other across the system.

System IS-IS (SIS-IS) is a lightweight system used to register processes within a distributed system, based on the use of Link-State routing protocols. Link-State protocols such as IS-IS [Oran90] and OSPF [Moy98], with their ability to reliably synchronize global system knowledge, have become the foundation of many carrier and enterprise routing networks. SIS-IS, prototyped using Linux and Quagga [QUAGGA], exploits these strengths so as to allow a large group of distributed processes to easily identify each other, both by type and location, across a set of processing elements. In addition to building arbitrary communication meshes between both sibling and other cooperating processes, knowledge of the global process state also allows any individual process to enter new processes into the overall system. The goal of these process additions is to enable both the desired scale across the whole system and the desired physical distribution across available processing elements, so as to meet overall system reliability requirements. This reliability is achieved by instantiating a specific process activity (*e.g.* routing, statistics collection, management, etc.) into a set of identical, cooperating sibling processes, all executing on different hardware components. The results from these sibling process groups are then compared prior to evaluating, and consequently selecting, them based on their validity, with the intent of removing any results that contain errors due to software or hardware faults.

3.2 Non-Technical Challenges

Success in resolving the technical issues will not be sufficient to ensure realization of the Nebula/R3 vision. From a value-chain perspective, we expect that the Internet services will be supported over facilities owned and controlled by multiple complementary and competing cloud service providers. Ensuring the security and reliability of end-to-end services while fostering open and vigorous facilities-based competition poses significant commercial and regulatory challenges.

The problem of ensuring interconnection across multiple networks is hardly new and underlies a history of extensive telecommunications regulation, but the technical, business, and policy challenges become immensely more complex in a world of cloud computing. First, the rise of cloud computing does not imply the decline of edge-based computing any more than the rise of electronic communications implied the end of paper-based communications. Second, the range of resources that need to be transparently shared and integrated is greatly expanded (transport, storage, computing cycles, and power). Third, the range of capabilities to be supported is much more ambitious (increased need for diverse QoS and security to support both sharing of video entertainment, health records, and public safety communications on much faster time-scales and across more diverse physical infrastructures ranging from fiber to ad hoc wireless). By focusing on a few prototypical challenges, we expect to be able to better highlight the challenges. One of those core challenges is the need to interconnect core routers across disparate ASes (where from an economic perspective, what we mean by AS is a centrally-managed cluster of networking resources, where the central management refers to the economic management of those resources – property rights to manage CAPEX/OPEX decision-making, including contracting for wholesale/retail services). The saliency of these issues was highlighted in a recent talk by Vint Cerf wherein he noted that the networking community with respect to interconnecting cloud resources is confronting a situation that is comparable in challenge and import to that which prevailed at the creation of the Internet [Cerf11]. The interconnection issue is attracting current attention relative to the discussions over Network Neutrality (network management) regulation [Stelter10] and as a consequence of the interconnection flap between Comcast and Level 3.

Clearly as we migrate more socially and economically diverse and important applications onto cloud resources, the challenges of ensuring appropriate reliability expand. The Nebula/R3 vision anticipates enabling an ultra-reliable Internet routing core via a fully distributed, high performance, fault-tolerant software platform that provides virtualized access to distributed data centers. Since different applications have different requirements and abilities to pay for security/reliability, it will be challenging to design fair and efficient resource allocation and cost recovery mechanisms.

The core switching fabric of modern telecommunications networks and electricity grids are designed to meet the requirements of "5 9's" reliability – implying availability asymptotically approaching 100%. This is viewed as a requirement for critical basic infrastructure. The Nebula/R3 goal is to achieve a similar level of highly reliable core routing functionality. A better understanding of how the incremental costs of enhanced system reliability might be shared across competing service providers is needed.

4.0 CONCLUSIONS AND NEXT STEPS

The cloud computing model provides a new form [Armbrust09] for networked computation and is a rich source of new applications. The challenges posed by the cloud computing model include high performance, high availability, flexible configuration and policy enforcement. The Nebula project is attempting to address these challenges in a comprehensive and coherent way, including thinking about regulatory policy and economic implications of new technologies, as well as the regulatory hurdles to their adoption.

The cable industry has been characterized by rapid introductions of new services. The rethinking of core router software architecture we have described here enables rapid deployment of these new services and capabilities, allowing for concurrent execution of multiple software versions, possible run-time updating of software systems and an "always-on" availability mode. More generally, the cloud computing model is well-suited to what can be perceived as in-network computing and data, lessening the burden on set-top box and cable modem technologies, thus reducing technology transitions at customer premises.

We expect the next steps to be deployment of software bundles made up of open source and proprietary software that result in an ultra-reliable router. Over the long-term, as our router software model evolves, we expect a software marketplace to emerge, with vendor communities competing to deliver novel products to service providers and their customers.

5.0 REFERENCES

[Alexander98] D. S. Alexander, W. A. Arbaugh, M. W. Hicks, P. Kakkar, A. D. Keromytis, J. Moore, C. A. Gunter, S. M. Nettles, and J. M. Smith, ''The SwitchWare Active Network Architecture,'' *IEEE Network Magazine, special issue on Active and Programmable Networks* **12**(3), pp. 29-36 (May/June 1998).

[Armbrust09] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica and Matei Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", *Technical Report No. UCB/EECS-2009-28*, Electrical Engineering and Computer Sciences, University of California at Berkeley, February 10, 2009.

[Birman03] Ken Birman, "The League of SuperNets", *IEEE Internet Computing*, **7**(5) 2003, pp. 92-96.

[Cerf74] V. Cerf and R. Kahn, "A Protocol for Packet Network Intercommunication", *IEEE Transactions on Communications*, Vol. **COM-22**, No. 5, pp 637-648, May 1974.

[Cerf11] V. Cerf. "Re-thinking the Internet," talk presented at Stanford, February 2, 2011, available at: http://www.youtube.com/watch?v=VjGuQ1GJkYc.

[Comer06] Douglas Comer, "Internetworking with TCP/IP: Volume 1: Principles, Protocols and Architecture, 5th Edition", Prentice-Hall 2006.

[Fano65] R. M. Fano, "The MAC System: The Computer Utility Approach," *IEEE Spectrum*, vol. 2, pp. 56-64 (Jan. 1965).

[Hicks05] M. Hicks and S. Nettles, "Dynamic software updating", in *ACM Trans. Program. Lang. Syst*. 27, 6 (Nov. 2005), pp. 1049-1096.

[Jacobson88] V. Jacobson, "Congestion Avoidance and Control", in *Proc. SIGCOMM 1988*, Stanford, CA., pp. 314-329.

[Loo05] Boon Thau Loo, Joseph M. Hellerstein, Ion Stoica, and Raghu Ramakrishnan, "Declarative Routing: Extensible Routing with Declarative Queries", ACM SIGCOMM Conference on

Data Communication, Philadelphia, PA, Aug 2005.

[Moy98] J. Moy. “OSPF Version 2,” *RFC 2328*, April 1998.

[Naous10] Jad Naous, Arun Seehra, Michael Walfish, David Mazières, Antonio Nicolosi and Scott Shenker, “Defining and enforcing transit policies in a future Internet,” *Technical Report TR-10-07*, Department of Computer Science, The University of Texas at Austin, February 2010.

[Oran90] D. Oran. “OSI IS-IS Intra-domain Routing Protocol,” *RFC 1142*, February 1990.

[Popa09] Lucian Popa, Ion Stoica, and Sylvia Ratnasamy. “Rule-based Forwarding (RBF): improving the Internet’s flexibility and security”, in *Proc. ACM Workshop on Hot Topics in Networks (HotNets),* October 2009.

[QUAGGA] “Quagga Routing Suite,” available at: http://www.quagga.net

[Stelter10] Brian Stelter. “F.C.C Faces Challenges to Net Rules,” *The New York Times*, December 22, 2010.

[Traw95] C. Brendan S. Traw and Jonathan M. Smith. ‘‘Striping within the Network Subsystem,’’ *IEEE Network*, pp. 22-32 (July/August 1995).