Democratizing the Network Edge

Larry Peterson
Open Network Foundation
llp@cs.princeton.edu

Tom Anderson
University of Washington
tea@cs.washington.edu

Sachin Katti
Stanford University
skatti@cs.stanford.edu

Nick McKeown
Stanford University
nickm@stanford.edu

Guru Parulkar
Open Networking Foundation
guru@opennetworking.org

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

Mahadev Satyanarayanan
Carnagie Mellon University
satya@cs.cmu.edu

Oguz Sunay
Open Networking Foundation
oguz@opennetworking.org

Amin Vahdat
Google, Inc.
vahdat@google.com

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 It has NOT been peer reviewed.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

ABSTRACT

With datacenters established as part of the global computing infrastructure, industry is now in the midst of a transition towards the edge. Previous research initiatives laid the groundwork for this transition, but that is no guarantee the emerging edge will continue to be open to researchers. This paper argues that there is a tremendous opportunity to innovate at the edge, but having impact requires understanding the nature of the current industry momentum, and making a concerted effort to align with that momentum. We believe there are three keys to doing this: (1) focus on the intersection of the cloud and access networks, (2) contribute to the relevant open source projects, and (3) address the challenge of operationalizing the results. The paper puts forward a concrete proposal for all three, and discusses the opportunity to influence how the Internet evolves at the edge and enable new and transformative edge applications.

CCS CONCEPTS

• **Networks → Network architectures**.

KEYWORDS

Radio Access Networks, Passive Optical Networks, Edge Cloud

1 ACCESS-EDGE CLOUD

Two industry trends with significant momentum are on a collision course. One is the cloud, which in pursuit of low-latency/high-bandwidth applications is moving out of the datacenter and towards the edge. The promise and potential of applications ranging from *Internet-of-Things (IoT)* to *Immersive UIs* to *Autonomous Vehicles* has triggered a gold rush to build edge platforms and services [1, 2]. The other is the access network that connects homes, businesses, and mobile devices to the Internet. Network operators (Telcos and CableCos) are transitioning from closed and proprietary hardware to disaggregated and virtualized software running on white-box servers, switches, and access devices [3].

The confluence of cloud and access technologies raises the possibility of convergence. For the cloud, access networks provide low-latency connectivity to end users and their devices, with 5G in particular providing native support for the mobility of those devices. For the access network, cloud technology enables network operators to enjoy the CAPEX savings that come from replacing purpose-built appliances with commodity hardware, as well as accelerating the pace of innovation through the softwartization of the access network.

The confluence of cloud and access technologies is also rich with opportunities to innovate. This is in part because there is a large set of research problems that need to be addressed to realize the *access-edge cloud* (Section 3 highlights some of them), and in part because of the availability of open platforms on which innovations can be both evaluated and deployed (Section 4 introduces some of them). We are at an inflection point. The question the research community should be asking is: *Where are the opportunities to shape the future, and how can we maximize our ability to have impact?*

Industry trends are creating this opportunity, and market forces will surely play an important role in picking winners and losers, but our approach is to focus on *democratizing the network edge for sustained innovation.* Doing so involves keeping both the edge cloud and access networks in scope, and working to lower the barrier for anyone (not just global carriers and cloud providers) to deploy and operate access-edge clouds. It also involves taking advantage of and contributing to open source software, and addressing the challenges of deploying and operationalizing complete systems.

2 DEMOCRATIZING

Before outlining the research opportunities or introducing an experimental approach to tackling them, we address the 800-pound gorilla in the room: that the two industry trends outlined in the Introduction correspond to two enormous industries—cloud providers and network operators—both of which are eager to define how the Internet evolves at the edge, with or without participation from the research community.

On the one hand, cloud providers believe that by saturating metro areas with edge clusters and abstracting away the access network, they can build an edge presence with low enough latency and high enough bandwidth to serve the next generation of edge applications. In this scenario, the access network remains a dumb bit-pipe, allowing cloud providers to excel at what they do best: run scalable cloud services on commodity hardware.

On the other hand, network operators believe that by building the next generation access network using cloud technology, they will be able to co-locate edge applications in the access network. This scenario allows operators to leverage their built-in advantages: an existing and widely distributed physical footprint, existing operational support, and native support for both mobility and guaranteed service.

We acknowledge both of these possibilities (and the research opportunities outlined in the next section encompass the full range, without regard to market winners and losers), but we advocate working towards a third outcome: the democratization of the access-edge cloud, making it widely accessible and not strictly the domain of incumbent cloud providers or network operators. There are three reasons why we are optimistic about this possibility.

First, hardware and software for access networks (and 5G in particular) is becoming commoditized. This is a key enabler that we discuss in more detail in Section 4. Big tower equipment will likely remain challenging since 5G brings a lot of complexity with the higher frequencies, but for small cells, we expect widespread availability in the next 2-3 years, with research platforms available today.

Second, there is demand. Enterprises in the automotive, factory, and warehouse space increasingly want to deploy private 5G networks for a variety of physical automation use cases (e.g., a garage where a remote valet parks your car or a factory floor making use of automation robots). The common theme is high bandwidth, low latency connectivity from the robot to intelligence sitting nearby in an edge cloud. This drives lower robot costs (you don't need to place heavy compute on each one) and enables robot swarms and coordination more scalably.

Third, spectrum is becoming available. 5G is opening up for use in an unlicensed or lightly licensed model, with CBRS in the US being a prime example. As another example, the German automotive industry recently lobbied the German regulator to make 5G spectrum available for free in the 3.5GHz band. Other European countries are likely to follow suit, meaning 5G should soon have around 100-200 MHz of spectrum available for private use.

3 OPPORTUNITY

The case for putting cloud services at the edge of the network is as old as the cloud itself [4, 5], and follows from the observation that human interaction times require low-latency connectivity to sufficient computational resources. Battery lifetime and physical limitations dictate that some computing needs to happen off-device (i.e., in the "cloud") and the speed-of-light dictates that for certain applications the nearest datacenter is too far away. The time is now right to address this challenge.

3.1 Architectural Framework

The access-edge is a fundamentally new component in the global Internet/Cloud, providing a rich opportunity for architectural work. To see this, consider that low latency will be achieved by moving functionality to the edge, closer to end-users and their autonomous devices, but this will imply that *the cloud itself needs to become mobile*, not just the user's broadband connection (see Figure 1). In other words, a key challenge will be to *simultaneously* support:

- **Low Latency:** Moving functionality to the edge, closer to devices.
- **Mobility:** Accessing that edge functionality while continuing to be mobile.

Earlier generations of wireless technology assumed functionality was located in the network core (e.g., in datacenters) and so only had to worry about making the broadband connection mobile; the functions the user was accessing remained fixed. When functionality is also at the edge—this functionality includes access services (RAN, PON, WiFi), converged packet core services (EPC, BNG), and value-add edge services—moving from one edge site to another implies that the corresponding functions may have to (logically) move as well.

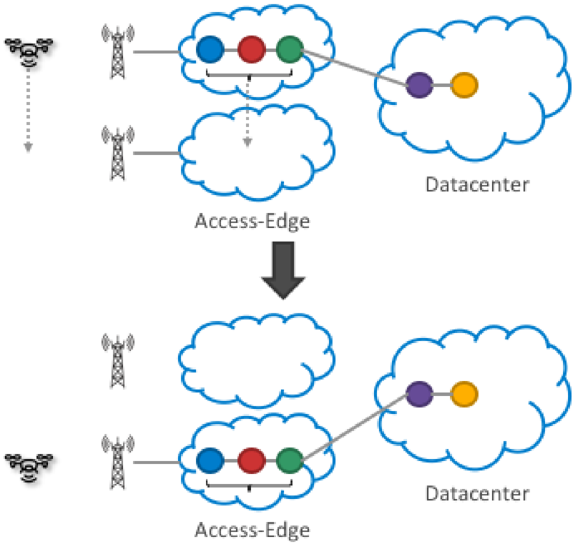

Figure 1: Mobile Cloud, simultaneously supporting edge functionality and mobility.

Simultaneously supporting mobility and edge functions is just one dimension of the architectural challenge. There are many other factors—several of which are outlined in the following subsection—all of which need to be taken into account when defining new frameworks, programming models, and interfaces in support of edge computing.

3.2 Mechanism Design and Optimization

In addition to architectural work, there are also a wealth of more narrowly defined research problems focused on designing, implementing, analyzing, and optimizing the mechanisms that realize such an architecture. Examples include:

- **Multi-Access:** The access-edge will need to support multiple access technologies (e.g., WiFi, 5G, fiber), and allow users to seamlessly move between them. Research is needed to break down existing technology silos, and design converged solutions to common problems (e.g., security, mobility, QoS).

Heterogeneity: Since the access-edge will be about low-latency and high-bandwidth connectivity, much edge functionality will be implemented by programming the forwarding pipeline in white-box switches, and more generally, will use other domain-specific processors (e.g., GPUs, TPUs). Research is needed to tailor edge services to take advantage of heterogeneous resources, as well as how to construct end-to-end applications from such a collection of building blocks.

Virtualization: The access-edge will virtualize the underlying hardware using a range of techniques, from VMs to containers to lambdas, interconnected by a range of L2, L3, and L4/7 virtual networks, some of which will be managed by SDN control applications. Research is needed to reconcile the assumptions made about by cloud native services and access-oriented Virtualized Network Functions (VNFs) about how to virtualize compute, storage, and networking resources.

Multi-Tenancy: The access-edge will be multi-tenant, with potentially different stakeholders (operators, service providers, application developers, enterprises) responsible for managing different components. It will not be feasible to run the entire access-edge in a single trust domain, as different components will operate with different levels of autonomy. Research is needed to minimize the overhead isolation imposed on tenants.

Customization: Monetizing the access-edge will require the ability to offer differentiated and customized services to different classes of subscribers/applications. Sometimes called network slicing, this involves support for performance isolation at the granularity of service chains—the sequence of functional elements running on behalf of some subscriber. Research is needed to enforce performance isolation in support of service guarantees.

Near-Real Time: The access-edge will be a highly dynamic environment, with functionality constantly adapting in response to mobility, workload, and application requirements. Supporting such an environment requires tight control loops, with control software running at the edge. Research is needed to analyze control loops, define analytic-based controllers, and design dynamically adaptable mechanisms.

Data Reduction: The access-edge will connect an increasing number of devices (not just humans and their handsets), all of which are capable of generating data. Supporting data reduction will be critical, which implies the need for substantial compute capacity (likely including domain-specific processors) to be available in the access-edge. Research is needed to refactor applications into their edge-reduction/backend-analysis subcomponents.

Distributed Services: Services will become inherently distributed, with some aspects running at the access-edge, some aspects running in the datacenter, and some running on premises or end device (e.g., on-vehicle). Supporting such an environment requires a multi-cloud solution that is decoupled from any single infrastructure-based platform, with research needed to develop heuristics for function placement.

Scalability: The access-edge will potentially span thousands or even tens of thousands of edge sites. Scaling up the ability to remotely orchestrate that many edge sites (even at just the infrastructure level) will be a qualitatively different challenge than managing a single datacenter. Research is need to scale both the edge platform and widely deployed edge services.

3.3 Edge-Native Applications

As to specific applications that best take advantage of an edge deployment, or correspondingly, how to best factor existing applications to their edge/centralized components, there is no definitive answer. Popularly exposed examples include Autonomous Vehicles, Internet-of-Things, and Immersive User Interfaces, but limiting ourselves serves no purpose.

The real objective should be to create brand new *edge-native* applications that are critically dependent on the low latency and high bandwidth that can only be provided from the edge, rather than limit ourselves to thinking only about *edge-accelerated* applications that already exist in today's cloud, but would be marginally better with edge computing. The ultimate goal is to leverage edge computing—low-latency processing, storage, and sensing—to augment cognition.

An illustrative example is *Wearable Cognitive Assistance*. The idea is to generalize what navigation software does for us: it uses one sensor (GPS), gives us step-by-step guidance on a complex task (getting around an unknown city), catches our errors promptly, and helps us recover. Can we generalize this metaphor? Could a person wearing a device (e.g., Google Glass, Microsoft Hololens) be guided step-by-step on a complex task, perhaps for the first time? The system would effectively act as "an angel on your shoulder." All the sensors on the device (e.g., video, audio, accelerometer, gyroscope) are streamed over wireless (possibly after some device preprocessing) to a nearby edge-cloud that performs the heavy lifting. This is a human-in-the-loop metaphor, with the "look and feel of augmented reality" but implemented by AI algorithms (e.g., computer vision, natural language recognition.) More information about prototypes of this and similar promising edge-native applications is available at http://gabriel.cs.cmu.edu.

3.4 Where is the Edge?

We conclude this section with a brief discussion of the first question that is typically asked: *Where is the edge?* Whether the edge is on-premise, on-vehicle, in the cell tower, in the Central Office, or distributed across a metro area (or all of the above) is partly a technical issue (i.e., how close to the end-user is it cost-effective to place computing resources), but also partly driven by market factors (i.e., who owns the edge and how is it monetized). With multiple incumbent players—e.g., network operators, cloud providers, cell tower providers—and countless startups jockeying for position, it's impossible to predict how the dust will settle. The dimensions outlined above are intended to be agnostic as to how the market shakes out. Any given research agenda is free to pick winners and losers, and prioritize the problem space accordingly, but all possible combinations of deployment scenarios should be on the table.

It is noteworthy that this paper emphasizes the access network as an integral part of the equation, rather than abstracting it away.

As discussed in Section 2, the access network has historically been the purview of the Telcos and the vendors that sell them proprietary boxes, but the softwarization and virtualization of the access network opens the door to another possibility: that the access-edge becomes democratized, making it possible for anyone (from smart cities to underserved regions to manufacturing plants) to establish an access-edge cloud and connect it to the public Internet as easily as it is today to deploy an IP router. Doing so not only brings the access-edge into new (edgier) environments, buty also has the potential to open the access network to developers that instinctively go where there are opportunities to innovate.

4 PLATFORM

Previous calls-to-action were born out of frustration with the Internet's ossification [6–8], but today's opportunity comes about because the networking industry is already in the midst of a transformation. This transition is being fueled by the increasing availability of open source software and commodity hardware. With organizations like the Open Compute Project (https://www.opencompute.org), the Open Networking Foundation (https://www.opennetworking.org), xRAN and the O-RAN Alliance (http://www.xran.org), the P4 Consortium (https://p4.org), and the Cloud Native Computing Foundation (https://www.cncf.io) leading the way, it is less about working around entrenched incumbents and more about *being productively engaged in these open source efforts.*

These open/disaggregated components are an important source of research problems and provide a clear tech transfer path, but it is also important to leverage them in a strategic way that maximizes impact. Working in safe-but-isolated environments that are not on the path to production deployments runs the risk of being overtaken by events. In contrast, embracing the components and platforms being advanced by the open source community in a coordinated way provides an opportunity for lasting impact, and just as importantly, establishes the foundation for sustained innovation and impact.

To this end, we propose CORD [9] as an Access-Edge Cloud experimental platform.[1] CORD integrates the relevant open source components (disaggregated radio and optical access networks, white-box switches with a programmable forwarding plane, micro-services infrastructure, end-to-end service mesh), packaged as a stand-alone POD that is easy to configure, build, deploy, and operate. Various configurations of CORD are being deployed in production networks by major carriers around the world, making it an ideal experimental platform with a compelling tech transfer story.

A hardware Bill-of-Materials and software download-and-install instructions are available online at https://guide.opencord.org. The rest of this section gives an overview of the approach we advocate, which has three parts: (1) to draw on a curated set of open source projects that cut across the traditional cloud/network boundary, where the individual projects included in CORD represent our current best judgement about the high-impact opportunities in influence the future of the converged access-edge cloud; (2) to package these components with a continuous integration and lifecycle management toolkit that supports continual innovation and evolution; and (3) to deploy and operate the resulting system in realistic environments carrying live traffic.

4.1 Disaggregated Components

Figure 2 gives a high-level overview of the CORD platform, which includes a complete open source software stack running on a cluster of commodity hardware. The following summarizes each layer in Figure 2, including specific examples that might be configured into a given configuration of CORD:

Commodity Hardware: CORD runs on a cluster of commodity servers, white-box switches, and white-box access devices. We typically recommend OCP-certified hardware, but this is not a hard requirement. The servers boot with Ubuntu Linux. The switches boot with Open Networking Linux (ONL), and optionally, a P4 Runtime agent. The switches are arranged in a leaf-spine fabric, although a minimal 4u configuration with a single switch, a single access device, and two servers is the most common developer/researcher configuration.

Controller/Orchestrator: On top of this hardware, CORD orchestrates the servers with some combination of a container manager (e.g., Kubernetes) and a VM manager (e.g., OpenStack). CORD controls both the switching fabric and the access devices with an SDN Controller (e.g., ONOS).

Disaggregated Components: The next layer consists of a collection of disaggregated components. These include a set of SDN Control Apps that specify how to program flow rules into the underlying switches and access devices (in blue) and a set of Virtualized Network Functions (VNFs) and other horizontally scalable cloud services running in server-hosted containers or VMs (in red). Example SDN Control Apps include vOLT (virtual Optical Line Terminal), vBBU (virtual Broadband Base Unit), and vRouter (virtual Router). Example VNFs include vEPC (virtual Extended Packet Core), vBNG (virtual Broadband Gateway), and vCDN (virtual Content Distribution Network).

Service Layer: The topmost layer (e.g., XOS) integrates the disaggregated components into a self-contained system. It defines a set of abstractions that govern how all the individual components interconnect to form a service mesh, how individual subscribers acquire isolated service chains across the service mesh, and how operators monitor, provision, and configure CORD.

Note that Figure 2 is an overview, and so does not do justice to the level of disaggregation in CORD. In addition to the control/data-plane disaggregation implied by SDN, CORD also makes heavy use of micro-services. To focus on two examples, vEPC (virtual Evolved Packet Core) is implemented as a mesh of five micro-services (one of which was subsequently rewritten in P4 and moved into the switching fabric), and vOLT (virtual Optical Line Terminal) is implemented a single ONOS control application plus a collection of ten other micro-services that abstract away differences in how the PON hardware is controlled. Some of these micro-services are purpose-built, but many are based on general-purpose Docker images, such as Kafka, Prometheus, and Elk-Stack. All of them are open source!

---

[1] CORD is an acronym for "**C**entral **O**ffice **R**e-architected as a **D**atacenter." The Central Office (CO) is the traditional edge of the Telco Network—for example, AT&T operates over 4500 COs across North America—but CORD is designed to be a general access-edge platform. It is not CO-specific, and it is designed to be an autonomous platform that has no dependencies on a global operator's network.

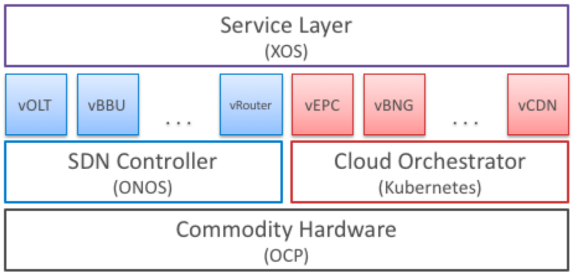

Figure 2: CORD software stack, including example components: vOLT, vBBU, vRouter, vEPC, vBNG, vCDN.

4.2 Radio Access Network

Of the access components available in CORD, the RAN is particularly noteworthy due to 5G's explicit goal of pivoting the cellular network from being purely connectivity-based to being service-based [10]. We discussed this larger architectural agenda in the previous section, but more narrowly, CORD opens the door for new services because of the flexibility 5G builds into the over-the-air interface. The eNodeB has historically been a closed/proprietary system, but efforts by the xRAN Forum—which is merging with Cloud-RAN to form the operator-led Open RAN (ORAN) Alliance—is driving towards open/programmatic control of the RAN. The cornerstone of this effort is to split the RAN stack between remote and centralized elements, a so-called *Split RAN.*

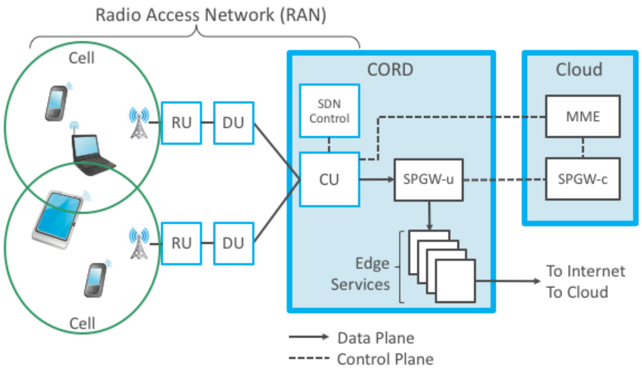

Figure 3: Split RAN, with Central Unit (CU) and EPC user plane (SPGW-u) running in an Access-Edge and the EPC control plane (MME,SPGW-c) running in a datacenter.

Figure 3 depicts a (simplified) realization of the proposed Split RAN, with the CU (Central Unit) implementing much of the L2/L3 layers of the legacy eNodeB, the DU (Distributed Unit) implementing the Media Access layer, and the RU (Radio Unit) implementing the Physical layer. The most important aspect of the split architecture is that the CU can be managed as an SDN-capable device, with control over how the radio spectrum is "sliced" on a per-subscriber basis codified by an SDN control application. The full implementation is still a work-in-progress for ORAN, but a version based on its xRAN predecessor runs in CORD today.

4.3 Operational Platform

A block diagram of disaggregated components does not a platform make. Individual researchers may be interested in only a single component of CORD, but the biggest obstacle to transitioning a new idea into practice is to operationalize it. We call this the *disruptor's dilemma:*

- Disaggregation catalyzes innovation. This is the value proposition of open networking.
- Integration facilitates adoption. This is a key requirement for any operational deployment.

Network operators recognize this problem, and their supply chain has historically included large integration teams. Unfortunately, the result of such integration efforts is often a point-solution that is both difficult to evolve and not particularly open to ongoing innovation. Fortunately, tools that foster continuous integration and deployment are now available. CORD adapts and extends them for this disaggregated environment. Stated succinctly:

> *CORD's goal is to sustain the innovate/deploy/operate cycle. This requires democratizing the operationalization of disaggregated components, which both facilitates adoption and enables continued research and innovation.*

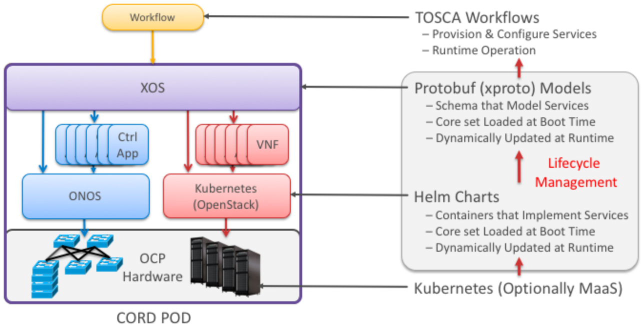

Figure 4: Operationalizing a CORD POD.

Figure 4 gives a high-level view of how a given instance of CORD (a so-called POD) is configured, deployed, and operated. From the bottom up, Kubernetes is first installed on the servers. On this foundation, a set of Helm Charts are executed to bring up a base set of micro-services, including OpenStack, ONOS, and XOS. Once XOS is running, a set of model definitions (expressed as an extension of ProtoBufs) are loaded, and the XOS tool chain uses these to both initialize the underlying components and to generate the NorthBound Interfaces (NBI) used to operate CORD. Finally, this NBI is used to provision and control a particular service workload. This is typically done using TOSCA workflows, although other interfaces are also available (e.g., REST, GUI, gRPC). Once operational, lifecycle management is handled by runtime updates to the XOS models and the Helm charts.

4.4 Deployment Scenarios

CORD can be deployed in multiple ways. A single POD can be used to experiment with both new edge services and individual platform components. A POD connected to a commodity cloud (and optionally, to on-premises/on-vehicle resources) permits experiments with distributed service meshes and function placement. A small set of PODS can be used to experiment with mobility across edge sites.

All of these configurations can be run in the lab or in a live environment (e.g., on a campus/enterprise or in a manufacturing plant using unlicensed bandwidth). We strongly advocate for live deployments carrying real traffic, and to this end, each POD is designed to be autonomous, requiring neither the resources nor the wherewithal of a global carrier to deploy and operate.

4.5 Continued Evolution

A long-standing dilemma in building experimental platforms is that many people want to work on (and change) the platform itself. The number of ways CORD and its subsystems can be improved are too many to count; the current version is just a starting point.

Fortunately, since CORD is a built-from-parts platform, and all of those parts are open source, there is an established path for platform innovation. Even CORD's "Platform Architecture" is programmable, as it is defined by a set of declarative models. The real challenge is to embrace the open source model, and the cost of shepherding innovations from proof-of-concept to software release.

It takes effort to see promising results through to adoption, but our collective experience strongly points to the value in pushing preliminary results through such a gauntlet. The goal is widespread use, either by production systems or by enriching the shared platform. It is often through the process of making cutting edge ideas practical-and-real that one discovers the next cutting edge problem that needs to be addressed [11].

5 CALL-TO-ACTION

The access-edge cloud—where the edge cloud and access networks intersect—is a fertile ground for innovation and a great opportunity to be out in front creating the future Internet. This paper lays out a plan for democratizing the access-edge: *opening it up for sustained innovation from all stakeholders*. This includes researchers and entrepreneurs, but even incumbent network operators benefit from an ecosystem that encourages the widest possible participation. The plan involves three mutually-supportive actions:

Focus on the Access-Edge: The access network has long been a critical part of the Internet, but it will become increasingly important as the cloud expands from the datacenter to points-of-presence closer to users. It is unique in its native support for mobility and resource guarantees, and as the place where access and the cloud intersect. The access-edge is not the only edge site (there will be many edges and there are many stakeholders), but we advocate democratizing the access-edge because it is such a critical link in the end-to-end chain.

Participate in Open Source: Working with open source software and commodity hardware removes a barrier to innovation, but working in a "research sandbox" that exists in parallel to what's happening in industry is not likely to have impact. We advocate aligning with and contributing to a curated set of open source projects, selected to have the greatest impact on what operators ultimately deploy.

Operationalize the System: Building prototypes and proofs-of-concept are a necessary step to validating ideas, but it is not sufficient for having impact. Integrating components into a fully operational system is a requirement for adoption, and having people use your results (whether they be end-users in production environments or other developers in experimental deployments) is essential for sustained innovation. We advocate gaining experience through live deployments. This is true for both the underlying platform and the edge-native applications that run on the platform.

We have proposed CORD as a technical approach to bootstrapping all three actions. CORD explicitly designed to run at the access-edge (i.e., it is not just a micro-datacenter), it is built from a curated set of open source and commodity components that are being embraced by network operators, and it includes open-and-accessible lifecycle management tools that lower the barrier for anyone to deploy, operate, and evolve an access-edge cloud.

In addition to the above recommendations, it is important to not lose sight of the big picture, which is to *enable new and transformative edge-native applications*. Edge computing is key to augmenting cognition. It provides the low-latency processing, storage and sensing infrastructure that is essential for this demanding class of applications. Although actual deployments of edge computing are minimal today, there is intense industry interest and it is believed that we are on the cusp of major industry investments [12].

ACKNOWLEDGMENTS

Many people provided comments, suggestions, and encouragement, including Michal Wawrzoniak, Akihiro Nakao, Abhimanyu Gosain, Raj Yavatkar, and Mark Berman. We are also indebted to the open source community that is contributing to CORD, and to the network operators working with the ONF on trial deployments.

REFERENCES

[1] Gartner Maverick Research. The Edge Will Eat the Cloud. September 2017. https://blogs.gartner.com/thomas_bittman/2017/03/06/the-edge-will-eat-the-cloud/
[2] P. Levine, Andreessen Horowitz. Return to the Edge and the End of Cloud Computing. August 2017. https://www.youtube.com/watch?v=-QRXQTSZxdQ
[3] SDxCentral. IHS Markit: 85% of Operators Plan to Deploy Smart Central Offices. January 2018. https://www.sdxcentral.com/articles/news/ihs-markit-85-of-operators-plan-to-create-smart-central-offices/2018/01/
[4] M. Satyanarayanan, *et. al.* The Case for VM-based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, October 2009.
[5] M. Satyanarayanan. The Emergence of Edge Computing. *IEEE Computer, Volume 50, Number 1*, January 2017.
[6] L. Peterson, *et. al.* PlanetLab: A Blueprint for Introducing Disruptive Technology into the Internet. *HotNets-I*, October 2002. (Appears in *ACM SIGCOMM Computer Communications Review 33(1):59-64*, January 2003.)
[7] T. Anderson, *et. al.* Overcoming Ossification Through Virtualization. *HotNets-III*, November 2004. (Appears in *IEEE Computer 38(4):34-41*, April 2005.)
[8] N. McKeown, *et. al.* OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communications Review 38(2)*, April 2008.
[9] L. Peterson, *et. al.* Central Office Re-architected as a Datacenter. *IEEE Communications*, October 2016.
[10] Qualcomm. Making 5G NR a Reality. September 2016. https://www.qualcomm.com/media/documents/files/making-5g-nr-a-reality.pdf
[11] L. Peterson and V. Pai. Experience-Driven Experimental Systems Research. *Communications of the ACM, Vol. 50(11):33-44*, November 2007.
[12] Editorial. Take it to the edge. *Nature Electronics, 2*, January 2019. https://doi.org/10.1038/s41928-019-0203-8.