The Emerging Landscape of Edge-Computing.

Shadi A. Noghabi
Microsoft Research

Landon Cox
Microsoft Research

Sharad Agarwal
Microsoft Research

Ganesh Ananthanarayanan
Microsoft Research

ABSTRACT

Edge computing is a trending notion introduced a decade ago as a new computing paradigm for interactive mobile applications. The initial vision of the edge was a multi-tenant resource that will be used opportunistically for low-latency mobile applications. Despite that vision, we see in practice a different set of applications, driven by large-scale enterprises, that have emerged and are driving real-world edge deployments today. In these applications, the edge is the primary place of storage and computation and if network conditions allow, the cloud is opportunistically used alongside. We show how these enterprise deployments are driving innovation in edge computing. Enterprise-driven scenarios have a different motivation for using the edge. Instead of latency, the *primary factors* are limited bandwidth and unreliability of the network link to the cloud. The enterprise deployment layout is also unique: on-premise, single-tenant edges with shared, redundant outbound links. These previously unexplored characteristics of enterprise-driven edge scenarios open up a number of unique and exciting future research challenges for our community.

1 Introduction

The original case for edge computing arose from the observation that interactive applications for lightweight devices can benefit from accessing more powerful machines over a low-latency, high-bandwidth network. Early advocates of edge computing envisioned a world of mobile devices that augmented their limited local resources by opportunistically launching short-lived, low-latency jobs on nearby edge servers [62]. For example, a cognitive-assistance application on a wearable computer could overlay realtime guidance on a heads-up display by streaming video to an inference model running on a nearby tensor-processing unit (TPU). This vision has been described as *cyber foraging* [60], and supporting interactive applications on mobile devices has been the focus of most academic and industry research on edge computing. In the coming years, the edge will likely provide critical infrastructure for emerging wearable and robotic systems. However, the cyber-foraging metaphor, i.e., that of a lightweight computer searching for nearby resources as it moves through an environment, does not accurately capture how most practitioners are using edge computing today.

Many of today's edge deployments are best described as *edge-sites* for long-running applications, such as industrial sensing and video analytics. These sites are single tenant, and they rarely (if ever) host transient jobs for mobile devices. For example, a restaurant might use the edge to monitor customer arrivals and schedule meal preparation [11], and an oil rig might process live video to identify safety hazards [17, 36]. Somewhat surprisingly, these data-processing applications are not bound by the strict latency requirements of cyber-foraging applications such as cognitive assistance (which must process video frames in 10ms or less). Restaurants must respond to arriving customers on the order of tens of seconds, and an oil rig must detect hazards within hundreds of milliseconds or seconds. With such relatively high latency tolerance, and the high availability, scalability, and low-cost computation offered by the cloud, why do these applications run on the edge rather than offloading to the cloud?

We observe that the dominant reasons for adopting edge computing are the need to tolerate cloud outages and the scarcity of network bandwidth. First, because today's edge applications are mission- and safety-critical, any downtime is unacceptable. A restaurant must serve food and an oil rig must prevent injuries, even when they are disconnected from the cloud. For these kinds of applications, an edge-site is the primary place of storage and computation, and cloud resources can be used opportunistically as edge-cloud network conditions allow. Second, insufficient bandwidth (even transiently) can be as bad as disconnection. An offshore oil rig analyzing dozens of 8K video feeds may have hundreds of ms to prevent an accident, but it would be infeasible (and expensive) for this application to transfer video frames to the cloud within that time (much less analyze the video) [17,36].

To be clear, we are by no means the first to point out that edge computing can mask cloud disconnections and make bandwidth-intensive applications economically viable [61, 63]. Rather in this paper, we wish to highlight how edge-sites are used by critical applications to survive transient network disruptions and how their structure differs from the cyber-foraging model. Viewing edge computing through the lens of today's use cases presents a distinct set of challenges and opportunities compared to mobile cyber foraging.

First, developers need better abstractions and services for

Table 1: **Real world deployments of edge compute**

Industry		Company	Use case	Edge location	Source
Business	Restaurants	Chick-fil-A	– forecast food preparation (e.g., more food needs to be fried)	In store	[11]
	Retail	Walmart, Coca Cola (vending machines)	– monitoring (e.g., fridge temperature ensuring produce quality) – tracking customers & improving sales (e.g., customized coupons)	In store	[3, 26]
	Gas station	Shell	– detect safety hazards (e.g., a person smoking a cigarette) across their 44,000 gas stations	In gas stations	[21]
Smart Cities	Cities	City of Bellevue	– traffic administration (e.g., intelligent control of traffic light) – safety at intersections (e.g., alerting drivers to prevent accidents)	Intersections & City clusters	[29, 50]
	Construction	PCL, ATF Services	– increase safety, efficiency, and productivity (e.g., detecting a temperature spike or gas leak in a unit) – increase security of construction cites (e.g., protecting equipment over night)	Construction site	[8, 19]
Transportation	Aviation	Airbus, Bombardier	– analyze in-flight experience of customers – monitor aircraft operations and maintenance	On Plane	[1, 16]
	Railway	CAF	– monitor train tracks, freight cars, and wheels for problems that lead to derailment	On the train	[9, 24]
	Road Control	Alaska DOT	– monitor quality of roads and detect roads with need of maintenance (e.g., finding spots that need snow plowing to prevent icing)	On Trucks	[14]
Industrial Plants	Oil Refinary	Schneider Electric, ExxonMobil	– predictive maintenance (of the pumps and equipment) – workplace safety	Oil rig or pump	[17, 27, 36]
	Manufacturing	GE, CPG, DAIHEN, Airbus	– improve manufacturing yields (e.g., automation or detecting defected products) – monitor equipment & predict need for maintenance	In factory	[40–42, 58]
	Manufacturing	BMW	– manage fleet of robots aiding in production pipeline	In factory	[15]
	Agriculture	Buhler	– control quality of produce at harvest, storage, and processing using imagery (e.g., for grains, processing 20,000 kernels/s).	In field	[18]
	Agriculture	DroneWorks, FarmBeats	– observe and monitor agricultural fields using sensors and drone imagery (e.g., detect areas that need water or pesticides)	In field	[10, 66]

building adaptive applications that can utilize the cloud when network conditions permit it but remain operational when conditions do not permit. Existing edge applications often consist of an ensemble of custom and off-the-shelf containers, with ad-hoc mechanisms for monitoring edge-to-cloud network conditions and primitive mechanisms to adapt their behavior. Second, edge resource managers must use an edge-site's multiple network interfaces. Many edge-sites support several network technologies, such as satellite, cellular, and broadband, and each technology offers different availability, cost, and performance characteristics. There is currently no standard mechanism for directing applications' messages to the appropriate interface. Finally, debugging and testing an edge-site application is extremely difficult. Edge-site conditions can be difficult to recreate prior to deployment, and incorrect behavior can arise from unanticipated interactions among adaptation strategies. Developers need better tools for choosing the right adaptation strategies as well as frameworks for testing solutions under realistic conditions. We further discuss these future research directions in §4.

2 Edge Compute Deployments

We discuss an array of real-world applications of edge computing (§2.1) and their deployment characteristics (§2.2).

2.1 Edge Applications

We studied over 20 different deployed edge-based applications across a variety of market segments. Table 1 summarizes the main application areas with real edge deployments and a variety of use-cases in each. In this section, we elaborate on these main areas and why/how they use the edge.

1) Business Intelligence. Customer experience and safety is gaining considerable interest from a wide range of businesses, such as retail stores (Walmart and Coca Cola), restaurants (Chick-fil-A), and gas stations (Shell). As an example, the Chick-fil-A restaurant chain has an IoT application to forecast when more food needs to be fried [11]. They built an in-restaurant prediction platform that relies on edge computing. They use video analytics and machine learning to predict the number of customers and cars entering the store. In this case, making a reliable prediction was the stated main motivation for using the edge. If the prediction fails (due to disconnection) or takes too long (due to a large amount of data transferred), the customer is left waiting.

Edge computing is preferred for these business scenarios as it avoids provisioning expensive outlink bandwidths to continuously transfer large data volumes to the cloud. Further, businesses expect their operations to function even when connectivity to the cloud is unavailable. Any downtime could mean large financial loss. Hence, the preference is a solution that is not reliant on always being connected to the cloud.

2) Smart Cities. City governments (e.g., City of Bellevue) and private entities (e.g., construction companies such as PCL and ATF Services), have deployed millions of cameras and sensors across the city: at intersections, in parking lots, and in construction zones. This data is used for improving efficiency and safety by using a nearby edge cluster. For example, the City of Bellevue uses cameras at traffic intersections for both controlling the traffic flow across the city and alerting drivers to avoid fatal accidents (e.g., a bicyclist approaching on the right). They employ the edge to overcome challenges with limited bandwidth and disconnection.

With the deployment of HD cameras, the data volume can be multiple Mbps for a single video stream. In city deployments, there is often sufficient bandwidth for all these streams to reach the edge (e.g., at the local traffic control center), but not beyond that to reach the cloud. This is while such use-cases can tolerate high application latencies in the

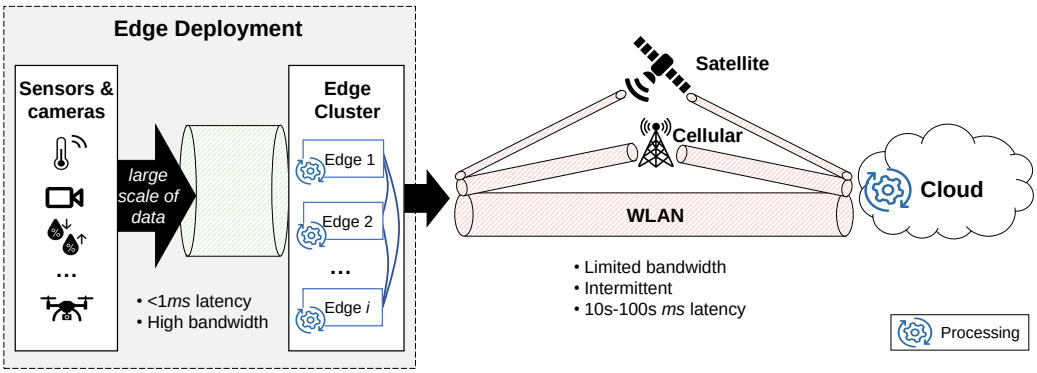


Figure 1: **Typical edge-site deployment architecture in enterprise environments.**

range of hundreds of milliseconds to minutes. Also, since many scenarios are centered around safety, continuous operation is critical even in the presence of disconnections.

3) Intelligent Transportation The transportation industry is moving to intelligent edge based solutions to ensure the safety of passengers, monitor vehicles and tracks, and improve customer experiences. The edge is used across various means of transportation including ground (Alaskan DOT), railway (CAF), and aviation (AirBus and Bombardier). As an example, the railway industry uses high-definition cameras in bungalows along the track, to detect cracks in train wheels. Cracks can cause the wheel to break and derail the entire train. The bandwidth demand for this case is dynamic. When a train passes a bungalow, it will generate GBs of data over a small period. At the same time, cracks should be detected and reported reliably within minutes to avoid severe casualties (financial and human lives).

Transportation scenarios are usually in remote and rural areas where broadband is not available.The connectivity is low bandwidth (few Kbps), intermittent, and expensive, while the scenarios are typically mission- and safety-critical. Hence, edge computing has played a major role in this industry.

4) Industrial Plants. Industrial plants deploy hundreds of thousands of sensors that continuously monitor mechanical equipment, worker safety, and production workflows to ensure issues are spotted and mitigated promptly. Several industries have already employed edge computing, such as oil and gas refineries (Schneider Electric and ExxonMobil), manufacturing (GE, CPG, DAIHEN, Airbus, and BWM), and agriculture (FarmBeats, DroneWorks, and Buhler). For example, ExxonMobile continuously monitors its multi-million dollar oil rigs. They use edge computing at each rig to detect equipment maintenance needs. They use this solution across their 11 million acres of land with ≈50,000 producing wells [27]. They cut the need for manual inspection across large fields and reduce the reaction time to unexpected issues. The motivation for using the edge is the unreliable and low-bandwidth satellite connectivity available in such areas.

Industrial plants are often in places with limited, unreliable, and expensive connectivity while generating large volumes of data (up to hundreds of TBs per day [58]). Factories in rural areas have highly intermittent connectivity and oil rigs only have expensive satellite links as their means of connectivity, which is unsuited for continuous high-volume data transfers. Any downtime can endanger the safety of workers or cause substantial financial losses, e.g., a day of downtime for a natural gas facility can cost $25 Million [37].

Summary: Enterprise driven scenarios are the dominant real deployments of edges today. Their use-cases are long-running jobs that range widely from simple filtering and aggregation to complex machine learning inferences and video processing. Across these scenarios, *network bandwidth and reliability drive the use of edge computing*, especially given the high volume of data generated, limited and intermittent connectivity to the cloud, and the mission- and safety-critical nature of these applications. On the other hand, latency is less stringent. The application's acceptable latency is typically in the range of a few seconds to minutes, and in the stricter cases it is always at least hundreds of ms.

2.2 Deployment Architecture

Figure 1 shows a representative enterprise edge deployment, e.g., a factory, store, or transportation site. We call this structure an *edge-site deployment*, where a dedicated set of edges are used as a primary place of storage and computation. If the network permits, the cloud is also used alongside the edge-site. The cloud is large pool of well-maintained resources with no management overhead imposed on the user. It provides better resource efficiency (by multiplexing across many usecases/users), high scalability, high availability, and low cost. Thus, it is preferable to utilize the cloud alongside the edge whenever possible, as shown by the growing number of hybrid (edge-cloud) management offerings [4–6, 25]. This is a main difference of edge-sites from the traditional "on-premise" only clusters used (mainly before the emergence of the cloud).

An edge-site deployment consists of a) *input devices*, i.e., an array of data-generating devices, and b) a cluster of connected edges. We next highlight the key characteristics of an edge-site deployment.

Table 2: **Edge deployments for enterprise environments (edge-site) vs. mobile-computing (cyber foraging) [62].**

Characteristic	Cyber foraging deployment	Edge-site deployment
Input devices	mobile phones, wearables, etc.	cameras, sensors, etc.
Edges	resource rich computer or cluster of computers	diverse hardware (Raspberry pi to rack of machines)
Outlink to cloud	i. single-link, ii.limited bandwidth	i. single- or multi-link, ii.limited bandwidth
Applications	i.high-volume, ii.short-running, iii.best-effort, iv.interactive (*ms* latency)	i.high-volume, ii.long-running, iii.critical, iv. latency-tolerant (hundreds of *ms – mins*)
Multi-tenancy	multi-tenant, multi-application	single-tenant, multi-application
Mobility	*input devices*: mobile *edges*: stationary	*input devices*: mobile in a boundary or stationary *edges*: stationary
Ownership & Management	*input devices*: consumers *edges*: 3rd party providers	*input devices*: enterprise *edges*: (typically) enterprise
Energy	*input devices*: battery-powered *edges*: plugged-in	*input devices*: battery-powered & plugged-in *edges*: plugged-in

Input devices: Input devices exhibit a wide diversity in their (i) types (sensors, cameras, etc.), (ii) amount of a data generated (Kb/s to tens of Mb/s), and (iii) scale (i.e., number of devices). For example, a retail store will consist of tens to hundreds of cameras, tracking items picked by customers along with several other sensors for humidity, temperature, etc.

Input devices are either a) *stationary*, e.g., cameras in a city intersection (e.g., Bellevue in Table 1), or b) *mobile but within the boundary of the edge-site deployment*, e.g., robots in a factory floor (e.g., BMW in Table 1). Regardless, they always have good connectivity to a nearby edge cluster, and do not face networking issues in migrating between edges.

Edge compute: Edges in an edge-site deployment are owned by and dedicated to one enterprise, i.e., edges are not shared among various enterprises or tenants. However, it is common for *multiple applications of a single enterprise* to share the edge cluster, e.g, run both fridge monitoring and customer tracking applications in a retail store. Not all applications have equal priority and criticality, but they share some degree of trust among each other. An example of such is the city running traffic control and accident prevention on a shared edge cluster at the intersection (Table 1).

Edges within an edge-site deployment typically form a *hierarchy* where smaller edges (less computing power) are connected to stronger edges (or racks of edges) up the tree. There is considerable diversity in processing power of edges, ranging from a small Raspberry Pi to racks of server-grade machines with high-end GPUs and FPGAs. It is common for edges to be spread geographically, e.g., across a factory floor. However, all edges are located within the enterprise's edge-site deployment boundary ("on-premise"), where there is high bandwidth connectivity between the different input devices and edge clusters in the deployment.

While input devices commonly can be battery powered (e.g., drones or sensors), edges are usually plugged-in to power with energy not being a first-order concern.

Connectivity to cloud: The edge-site deployment is connected to the cloud (and generally to the outside world) via a number of network links, which we call *outlinks*. The outlinks are shared among all input devices, edges, and applications within a deployment. However, the amount of data generated by the input devices (e.g., sensors and cameras) is usually significantly larger than the total capacity of the out-link(s) of an deployment. This is because of both a) the large volume of data generated (TBs – PBs per day), and b) the limited bandwidth of out-links (as shown in Table 3), especially in remote locations.

While it is common for deployments to have multiple outlink connectivity options such as cellular and satellite, they are usually for *backup* purposes. Only one is the primary option, as many of these backup options are considerably more expensive or have very low bandwidth (e.g., satellite).

Summary: On examining current edge computing deployments (edge-site deployments) we find significantly different characteristics than the original vision of edge computing (cyber foraging). In current deployments, edge clusters are deployed by a single entity (not multi-tenant) with enterprise applications (rather than consumer applications) driving the deployments. Table 2 expands on these points.

3 Driving factors for edge

Edge computing was originally motivated by the need for additional computation in mobile devices that experience *high network latency* to the cloud [62]. However, current edge deployments, driven by enterprises, exhibit a different motivation for using the edge. Enterprise scenarios generate high volumes of data and are usually mission-critical applications. Relying on the cloud over an unreliable network connection can result in fail-stop or worse impact. This is while the tolerable latency of such applications is typically hundreds of milliseconds to even minutes. Thus, their main motivation for using the edge is the lack of reliable connectivity and/or lack of sufficient bandwidth. In this section, we discuss the technical causes behind these trend changes.

3.1 Latency to DCs has become less of an issue

Anticipating the need for faster access to powerful compute, it was envisioned that edge compute would be placed close to the user, on the network path to the public datacenters [62]. However, due to a few key trends, latency to datacenters has become less of an issue, i.e., the $<$100ms latency of reaching

Table 3: **Real-world speeds of network technologies.**

Region	Connectivity type	Download speed (Mbps)	Upload speed (Mbps)
USA	DSL [39]	2-20	0-3
	Cable [39]	78-120	7-32
	Fiber [39]	56-78	16-85
	Satellite [39]	11-19	3-14
	4G Cellular [57]	15-22	3-7
Egypt	Fixed Broadband [23]	3-9	1-5
	Cellular [23]	5-9	2-4
Australia	Fixed Broadband [23]	18-67	9-24
	Cellular [23]	37-58	13-23

a datacenter is not a prominent factor (especially compared to the tolerable latency of current enterprise driven scenarios).

1) Wider deployment of DCs: Cloud providers have built significant numbers of datacenters across the planet. For example, Azure has 54 regions worldwide, with multiple DCs in each region, and more coming online soon [7]. Hence, the physical distance that a packet has to travel to reach a DC has shrunk dramatically in the past decade [49].

2) Deeper peering into ISPs: Cloud providers have significantly cut network latency between end users and their DCs by peering more directly with thousands of ISPs, and reducing the number of intermediate networks that packets use to traverse. For example, Google offers direct peering with their cloud network in >100 locations worldwide [13].

3) Specialized compute for ML and AI: Some of the end-to-end latency-sensitive applications that were envisioned for edge computing can now run efficiently on mobile devices through hardware specialization. Apple has built a neural engine (an 8 core dedicated ML processor) into the A12 Bionic ARM SoC in the latest iPhones that is capable of running ML apps up to 9 times faster than previously possible [2]. Similar capabilities are being deployed on security cameras. This on-device capability has reduced the need to rely on computation offloading for some applications, and allows for a different application-layer tradeoff where quick responses can be provided to the user locally, and remote DCs can be used for latency-insensitive tasks.

3.2 Bandwidth to DCs is insufficient

Sufficient bandwidth to the cloud remains a challenge. City cameras, oil refineries, and factories generate TBs of data each day, which is prohibitively high to upload to the cloud. In some extreme cases, like jet planes, the sensors put together generate over 850PB in a single 12-hour flight [16].

Deployed speeds of broadband connectivity remain surprisingly low compared to the latest advances in networking technology. The average US broadband speed was largely under 100Mb/s download and roughly 30Mb/s upload [12], and much worse elsewhere in the world [22]. More remote regions such as oil rigs and train depots often have to rely on satellite links that offer limited bandwidth [36]. Table 3 summarizes the speeds of different connectivity options.

Further, bandwidth needs may be intermittent and dynamic. For example, when a train rolls into a train station its wheels are inspected using video analytics, causing a spike. While all the data may be eventually streamed to the cloud, the bandwidth may not be sufficient to stream it and get back a response in time for mission-critical applications.

3.3 Connectivity is unreliable

Lack of reliable connectivity is reflected in a recent survey by Ofcom, the UK communications regulator. It finds that since 2016, there has been a decline in broadband customers' satisfaction with their broadband service, down from 87% to 80%. Among dissatisfied customers, the top two reasons were poor or unreliable connectivity (48%) and slow speeds (47%)." [20]. This problem is worse in non-urban environments where broadband connectivity is not available, such as remote train yards and oil rigs. Satellite connectivity is notorious for being dependent on weather conditions. There are also many occurrences of natural disasters and human error failures (e.g., cutting the network link) causing long disconnections. When coupled with the mission-critical nature of the applications we have described, even small amounts of connectivity outages can have a major impact on safety and financial viability.

4 Future Research Directions

Edge-sites help mission-critical applications remain operational in the face of transient changes to cloud-edge connectivity and workload variations. However, the current system support for developing, managing, and scheduling these applications, in a way that they use both the edge and cloud effectively, is poor. In this section, we describe three promising directions for research on edge-sites.

1) Graceful Adaptation of Applications: In the cyber foraging model, applications adapt to changes in connectivity to an edge by simply using another nearby edge, and if not possible, the cloud with presumed infinite-resource. However, in the edge-site model, if the connectivity to the cloud changes, the application must adapt by using the limited edge resources, which could often mean changes in the application logic itself. A valuable objective is to enable applications to *adapt gracefully* in the presence of disconnections, drops in bandwidth, or workload spikes.

While adaptation is a generally desired property (even in a cloud-only environment), traditionally, an application-logic agnostic solution is sufficient. For example, automatically handing off applications between edges (in the cyber foraging model) does not require to know exact details of the application logic. However, in the emerging edge-site deployments, appropriate adaptation strategies are application specific and can vary significantly. In particular, after a change to network conditions, an application could adjust the a) *input-data* quality, e.g., reducing the resolution of videos or increasing sensor reading periods, b) *compute* quality, e.g.,

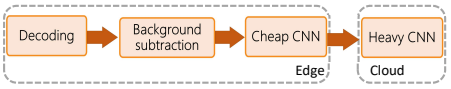
using larger batch sizes or larger aggregation windows, or c) *output-data* quality, e.g., compressing more aggressively or filtering/discarding parts of the data. Graceful adaption may also depend on the edge-site environment, an application's priority, the severity of changing conditions, and the duration of changes.

Choosing an appropriate adaptation strategy must ultimately be a developer's responsibility. However, instead of relying on application developers to manually incorporate solutions to react to changes (using an ensemble of off-the-shelf and custom built components), the system software can greatly simplify the task of building adaptive applications by automatically monitoring network conditions and managing callbacks. In addition, support for adaptive orchestrated applications presents an opportunity to reconsider the roles that applications and the network play in managing message content and queues. For example, it may make sense for applications to have exact control over how (or if) messages are dropped. Similarly, it may make sense to decouple message ordering and content, so that a delayed message can be transformed in-place rather than the existing workflow of the network dropping messages and the application retrying them.

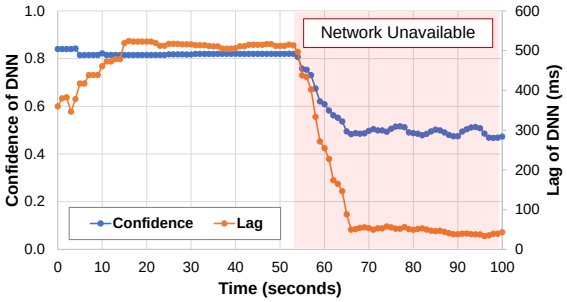
Figure 2 illustrates a simple adaptation mechanism based on cloud connectivity for a video analytics application spanning across the edge and cloud. The video stream being processed is from a fast-food restaurant chain where they detect cars pulling into their driveway and pre-make food items accordingly. The restaurant in our pilot deployment required the analytics functioning even without cloud connectivity.

The video analytics pipeline is shown in Figure 2a where a cascaded set of vision operators lead to calling the CNN (Convolutional Neural Network) in the cloud for higher confidence of results. We use Tiny Yolo as our cheap CNN and the full YoloV3 model as our heavy CNN [28]. As Figure 2b shows, the system calls on to the cloud to provide detections with higher confidence when connectivity is available, but automatically switches to an "edge only" mode when disconnected (shown in red shade). An interesting tradeoff is that calling the cloud also causes additional *lag*, and this has to be balanced against the value of the higher confidence. This experiment is our first stab at network-based adaption for a sample application, which we intend to generalize.

2) Collaborative & App-aware Network Orchestration: In the cyber foraging model, the prevailing assumption is that the edge is shared among multiple untrusted user devices, thus, much of the research effort has been expended in building performance and security isolation between tenants on edges [44,45,65,67]. On the contrary, the edge-sites today are a collaborative, single-tenant environment with no adversarial application trying to greedily use the entire link. Instead of a traditional network orchestration mechanism dividing link capacity, we see an opportunity for a new collaborative approach that leverages applications' knowledge of how to adapt. Ultimately, the application should decide how to adapt, i.e., what data to change, drop, or reorder, and the network orchestration should decide how to allocate re-



(a) **Video analytics pipeline**



(b) **Pipeline results when adapting to availability to the cloud.**

Figure 2: **Network adaptation for video analytics.**

sources across such adaptive applications in a collaborative manner.

Moreover, in the cyber foraging model, mobile devices would have intermittent connectivity to the edge and the edge would have stable connectivity to the cloud. In the edge-site reality of today, the input devices (generating data) have stable connectivity to the edge and the edge has intermittent connectivity to the cloud. Hence, the valuable research directions are in better managing and prioritizing the connectivity of edge applications to the cloud. The network orchestrator must make good use of an edge deployment's network interfaces (i.e., being cost aware) while also ensuring the needs of critical applications are met. Part of this problem requires choosing the right network technology. Many deployments support several network technologies, such as satellite, cellular, and broadband, and each technology offers different availability, cost, and performance characteristics. However, there is currently no standard mechanism for routing the messages through the right interface. This mechanism should be both priority-aware (to ensure critical application's requirements are met) and cost-aware. Of course, balancing the needs of critical applications and less urgent, background applications requires additional information from applications. Thus, a collaborative network orchestration will require policies that balance disparate application needs, performant decision making, and well designed interfaces for exposing message priorities to the network.

3) Test and Verification Frameworks: Adding adaptation logic to applications in the edge-site model increases the application complexity. This differs from the cyber foraging model where the complexity of adaptation is mostly placed on the edge infrastructure, e.g., transparently handing-off applications to other edges [45]. Hence, an important research direction is to simplify choosing the right adaptation strategies for developers and to verify their choices. We need to build frameworks that can test and verify the correctness of adaptation decisions under various conditions. The

framework should allow developers to specify baseline performance, e.g., recognition accuracy above 80%, and allow for their code to be tested under various conditions, such as short and long disconnections, low bandwidth, and congestion.

There are two main challenges for building such a framework. First, what is an easy yet comprehensive way to define baseline performance for an application? Second, what is the minimum set of test-cases an application should be tested against? The testing search space is massive, with many variables, such as the length of disconnection, bandwidth values, and the set of other competing applications. Providing good testing coverage in a timely manner will be challenging.

5 Related Work

Adapting Environment: Researchers have investigated adapting applications to changing network conditions for over 20 years [30]. The best known early work on this topic focused on strategies for adaptive data access, such as giving mobile clients low-latency access to reduced-quality media [43, 56] or offering weaker file consistency [48, 53] when the network degraded. More recent work has focused on adaptive computation, exploring the trade-offs of using adaptation at the remote procedure call interface [32], the managed runtime interface [34, 38], the hardware virtual-machine interface [35,62], and the machine-learning pipeline interface [59].

Despite this large body of work, practitioners have not embraced a common set of abstractions for adaptive edge applications. Most prior work targets mobile applications, which usually execute as a single process. In contrast, today's edge applications are a graph of several diverse software components spanning multiple programming models, languages, processes, and machines, with a variety of synchronous and asynchronous communication patterns. No existing adaptation approach is a good fit for these kinds of applications.

Optimizing Outlink Bandwidth: There is a rich body of research on scheduling network traffic in devices with access to multiple networks [31]. Much of this work falls into one of two categories: clients with multiple onboard radios [33, 46, 52] and clients with a single radio and access to multiple networks [47, 54, 55]. Nearly all of the work on this topic focuses on placing traffic-controlling mechanisms on the same physical hardware as application code. However, current edge applications are built with cloud programming models that do not match the interfaces for prioritizing mobile traffic over wireless networks.

The limited bandwidth to the cloud has been a well-observed factor for the edge [61–63, 65], with several approaches to optimizing bandwidth, especially for video streams [64, 68, 69]. However, for enterprise scenarios, this limitation is different as: a) bandwidth is a *primary* motivation for the edge, and b) all applications belong to a single administrative domain enabling collaborative and cooperative solutions.

Autonomous Edge: Primarily relying on the edge has been proposed in the context of *hostile environments*, e.g., in military bases [51, 61, 63]. However, hostile environments are extreme cases with many sources of unreliability: devices move, leave, and their batteries die, which requires complex solutions that are an overkill for edge-site deployments.

6 Conclusion

In this paper, we challenge long-held predictions on why and how edge computing will be deployed. We cite numerous examples of real-world deployments of edge computing across several different applications. These applications are not end-user interactive mobile applications opportunistically using the edge as originally envisioned. Rather, they are geographically constrained, mission-critical, industrial or enterprise applications that primarily rely on the edge and opportunistically use the cloud. These deployments have not been motivated by the need for low latency access to the cloud, but rather by the lack of sufficient and reliable bandwidth to the cloud. We have outlined a number of interesting research challenges that need to be solved in this context.

7 References

[1] Airbus leveraging IoT and IBM's Watson for connected aircraft. https://internetofbusiness.com/airbus-leveraging-iot-and-ibms-watson-for-connected-aircraft/. Accessed: 03/2019.

[2] Apple's A12 Bionic chip runs Core ML apps up to 9 times faster. https://venturebeat.com/2018/09/12/apples-a12-bionic-chip-runs-core-ml-apps-up-to-9-times-faster/. Accessed: 02/2019.

[3] Atos becomes official IoT partner for Coca-Cola Hellenic Bottling Company. https://atos.net/en/2018/press-release_2018_07_03/atos-becomes-official-iot-partner-coca-cola-hellenic-bottling-company. Accessed: 03/2019.

[4] AWS Greengrass. https://aws.amazon.com/greengrass/. Accessed: 03/2019.

[5] Azure Internet of Things. https://azure.microsoft.com/en-us/suites/iot-suite/. Accessed: 03/2019.

[6] Azure IoT Edge. https://azure.microsoft.com/en-us/services/iot-edge/. Accessed: 03/2019.

[7] Azure Regions. https://azure.microsoft.com/en-us/global-infrastructure/regions/. Accessed: 02/2019.

[8] Building a new future: Transforming Australia's construction industry with digital technologies. https://customers.microsoft.com/en-us/story/atf-services. Accessed: 03/2019.

[9] CAF Increases Train Safety with AWS IoT. https://aws.amazon.com/solutions/case-studies/caf/. Accessed: 03/2019.

[10] Developers help build a safe, standardized flight platform for industrial drones with Azure IoT Hub. https://customers.microsoft.com/en-us/story/droneworks. Accessed: 03/2019.

[11] Edge Computing at Chick-fil-A. https://medium.com/@cfatechblog/edge-computing-at-chick-fil-a-7d67242675e2. Accessed: 02/2019.

[12] Fixed Broadband Speedtest Data for United States. https://www.speedtest.net/reports/united-states/2018/fixed/. Accessed: 02/2019.

[13] Google Network Edge Locations. https://cloud.google.com/vpc/docs/edge-locations.

Accessed: 02/2019.

[14] How Alaska outsmarts Mother Nature in the cloud. https://customers.microsoft.com/en-us/story/alaskadotpf-government-azure-iot. Accessed: 03/2019.

[15] Improving Safety and Efficiency in BMW Manufacturing Plants with an Open Source Platform for Managing Inventory Delivery. https://www.microsoft.com/developerblog/2018/12/19/improving-safety-and-efficiency-in-bmw-manufacturing-plants-with-an-open-source-platform-for-managing-inventory-delivery/. Accessed: 03/2019.

[16] Internet Of Aircraft Things: An Industry Set To Be Transformed. https://aviationweek.com/connected-aerospace/internet-aircraft-things-industry-set-be-transformed. Accessed: 03/2019.

[17] Oil and gas experts use machine learning to deploy predictive analytics at the edge. https://customers.microsoft.com/en-us/story/schneider-electric-process-mfg-resources-azure-machine-learning. Accessed: 03/2019.

[18] One grain at a time: how Bühler is combining advanced data analysis with machine learning to tackle a global food chain problem. https://customers.microsoft.com/en-us/story/buhlergroup-azure-machine-learning-iot-edge-switzerland. Accessed: 03/2019.

[19] PCL Construction uses IoT with Azure to revolutionize the construction industry. https://customers.microsoft.com/en-us/story/pcl-construction-professional-services-azure. Accessed: 03/2019.

[20] Residential landline and fixed broadband services. https://www.ofcom.org.uk/__data/assets/pdf_file/0015/113640/landline-broadband.pdf. Accessed: 02/2019.

[21] Shell invests in safety with Azure, AI, and machine vision to better protect customers and service champions. https://customers.microsoft.com/en-us/story/shell-mining-oil-gas-azure-databricks. Accessed: 03/2019.

[22] Speedtest Market Report for Egypt. https://www.speedtest.net/reports/egypt/. Accessed: 02/2019.

[23] Speedtest Market Reports. https://www.speedtest.net/reports/. Accessed: 03/2019.

[24] Transportation:What You Need to Know, on the Go. https://www.foghorn.io/transportation/. Accessed: 03/2019.

[25] VMware Edge–Innovate at the Edge. https://www.vmware.com/solutions/edge-internet-of-things.html. Accessed: 03/2019.

[26] Walmart establishes strategic partnership with Microsoft to further accelerate digital innovation in retail. https://news.microsoft.com/2018/07/16/walmart-establishes-strategic-partnership-with-microsoft-to-further-accelerate-digital-innovation-in-retail/. Accessed: 03/2019.

[27] XTO Energy taps into IoT and the cloud to optimize operations and drive growth with Azure. https://customers.microsoft.com/en-us/story/exxonmobil-mining-oil-gas-azure. Accessed: 03/2019.

[28] YOLO: Real-Time Object Detection. https://pjreddie.com/darknet/yolo/. Accessed: 03/2019.

[29] ANANTHANARAYANAN, G., BAHL, V., BODÍK, P., CHINTALAPUDI, K., PHILIPOSE, M., SIVALINGAM, L. R., AND SINHA, S. Real-time video analytics – the killer app for edge computing. *IEEE Computer* (2017).

[30] BADRINATH, B. R., FOX, A., KLEINROCK, L., POPEK, G. J., REIHER, P. L., AND SATYANARAYANAN, M. A conceptual framework for network and client adaptation. *MONET 5*, 4 (2000), 221–231.

[31] BAHL, P., ADYA, A., PADHYE, J., AND WOLMAN, A. Reconsidering wireless systems with multiple radios. *SIGCOMM Comput. Commun. Rev. 34*, 5 (Oct. 2004), 39–46.

[32] BALAN, R. K., SATYANARAYANAN, M., PARK, S. Y., AND OKOSHI, T. Tactics-based remote execution for mobile computing. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services* (New York, NY, USA, 2003), MobiSys '03, ACM, pp. 273–286.

[33] BRIK, V., MISHRA, A., AND BANERJEE, S. Eliminating handoff latencies in 802.11 wlans using multiple radios: Applications, experience, and evaluation. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement* (Berkeley, CA, USA, 2005), IMC '05, USENIX Association, pp. 27–27.

[34] CHUN, B.-G., IHM, S., MANIATIS, P., NAIK, M., AND PATTI, A. Clonecloud: Elastic execution between mobile device and cloud. In *Proceedings of the Sixth Conference on Computer Systems* (New York, NY, USA, 2011), EuroSys '11, ACM, pp. 301–314.

[35] CHUN, B.-G., AND MANIATIS, P. Augmented smartphone applications through clone cloud execution. In *Proceedings of the 12th Conference on Hot Topics in Operating Systems* (Berkeley, CA, USA, 2009), HotOS'09, USENIX Association, pp. 8–8.

[36] CISCO. New Realities in Oil and Gas: Data Management and Analytics . Tech. rep., Cisco, 2017.

[37] CLIFFORD, M. J., PERRONS, R. K., ALI, S. H., AND GRICE, T. A. *Extracting Innovations: Mining, Energy, and Technological Change in the Digital Age*. CRC Press, 2018.

[38] CUERVO, E., BALASUBRAMANIAN, A., CHO, D.-K., WOLMAN, A., SAROIU, S., CHANDRA, R., AND BAHL, P. Maui: Making smartphones last longer with code offload. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2010), MobiSys '10, ACM, pp. 49–62.

[39] FEDERAL COMMUNICATIONS COMMISSION, OFFICE OF ENGINEERING AND TECHNOLOGY. Measuring Fixed Broadband - Eighth Report December 14, 2018. https://www.fcc.gov/reports-research/reports/measuring-broadband-america/measuring-fixed-broadband-eighth-report. Accessed: 03/2019.

[40] FOGHORN. DAIHEN Automates Production of Industrial Transformers, Improves Materials Quality Monitoring and Collaboration. Tech. rep., FogHorn Systems, 2018.

[41] FOGHORN. GE Detects Early Defects and Improves Capacitor Production Yield with Edge Intelligence . Tech. rep., FogHorn Systems, 2018.

[42] FOGHORN. Global Consumer Packaged Goods Company Improves Yield through Real-time Insights. Tech. rep., FogHorn Systems, 2018.

[43] FOX, A., GRIBBLE, S. D., BREWER, E. A., AND AMIR, E. Adapting to network and client variability via on-demand dynamic distillation. In *Proceedings of the Seventh International Conference on Architectural Support for Programming Languages and Operating Systems* (1996), ASPLOS.

[44] GARCIA LOPEZ, P., MONTRESOR, A., EPEMA, D., DATTA, A., HIGASHINO, T., IAMNITCHI, A., BARCELLOS, M., FELBER, P., AND RIVIERE, E. Edge-centric computing: Vision and challenges. *SIGCOMM Computer Communication Review 45*, 5 (2015).

[45] HA, K., ABE, Y., EISZLER, T., CHEN, Z., HU, W., AMOS, B., UPADHYAYA, R., PILLAI, P., AND SATYANARAYANAN, M. You can teach elephants to dance: agile vm handoff for edge computing. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing* (2017), ACM, p. 12.

[46] HIGGINS, B. D., REDA, A., ALPEROVICH, T., FLINN, J., GIULI, T. J., NOBLE, B., AND WATSON, D. Intentional networking: opportunistic exploitation of mobile network diversity. In *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking, MOBICOM 2010, Chicago, Illinois, USA, September 20-24, 2010* (2010), pp. 73–84.

[47] KANDULA, S., LIN, K. C.-J., BADIRKHANLI, T., AND KATABI, D. Fatvap: Aggregating ap backhaul capacity to maximize throughput. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation* (Berkeley, CA, USA, 2008), NSDI'08, USENIX Association, pp. 89–104.

[48] KISTLER, J. J., AND SATYANARAYANAN, M. Disconnected operation in the coda file system. *ACM Trans. Comput. Syst. 10*, 1 (Feb. 1992), 3–25.

[49] LI, A., YANG, X., KANDULA, S., AND ZHANG, M. Cloudcmp: Comparing public cloud providers. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement* (2010), IMC '10, ACM, pp. 1–14.

[50] LOEWENHERZ, F., BAHL, V., AND WANG, Y. Video analytics towards vision zero. *ITE Journal 87* (March 2017), 25–28.

[51] MAHADEV, S., LEWIS, G., MORRIS, E., SIMANTA, S., BOLENG, J., AND HA, K. The role of cloudlets in hostile environments. *Pervasive Computing 12*, 4 (2013).

[52] MIU, A., BALAKRISHNAN, H., AND KOKSAL, C. E. Improving loss resilience with multi-radio diversity in wireless networks. In *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking* (New York, NY, USA, 2005), MobiCom '05, ACM, pp. 16–30.

[53] MUMMERT, L. B., EBLING, M. R., AND SATYANARAYANAN, M. Exploiting weak connectivity for mobile file access. In *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles* (1995), SOSP.

[54] NICHOLSON, A. J., CHAWATHE, Y., CHEN, M. Y., NOBLE, B. D., AND WETHERALL, D. Improved access point selection. In *Proceedings of the 4th International Conference on Mobile Systems, Applications and Services* (New York, NY, USA, 2006), MobiSys '06, ACM, pp. 233–245.

[55] NICHOLSON, A. J., WOLCHOK, S., AND NOBLE, B. D. Juggler: Virtual networks for fun and profit. *IEEE Transactions on Mobile Computing 9*, 1 (Jan. 2010), 31–43.

[56] NOBLE, B., SATYANARAYANAN, M., NARAYANAN, D., TILTON, J. E., FLINN, J., AND WALKER, K. R. Agile application-aware adaptation for mobility. In *Proceedings of the Symposium on Operating Systems Principles (SOSP)* (1997), ACM.

[57] OPENSIGNAL. Mobile Network Experience Report January 2019. `https://www.opensignal.com/reports/2019/01/usa/mobile-network-experience`. Accessed: 03/2019.

[58] OWEN, P., AND MARTIN, R. The Business Case for IIOT Edge Intelligence. Tech. rep., ABI Research, 2018.

[59] RA, M.-R., SHETH, A., MUMMERT, L., PILLAI, P., WETHERALL, D., AND GOVINDAN, R. Odessa: Enabling interactive perception applications on mobile devices. In *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2011), MobiSys '11, ACM, pp. 43–56.

[60] SATYANARAYANAN, M. Pervasive computing: Vision and challenges. *IEEE Personal Communications 8* (2001), 10–17.

[61] SATYANARAYANAN, M. The emergence of edge computing. *Computer 50*, 1 (2017).

[62] SATYANARAYANAN, M., BAHL, P., CACERES, R., AND DAVIES, N. The case for vm-based cloudlets in mobile computing. *Pervasive Computing 8*, 4 (2009).

[63] SATYANARAYANAN, M., SCHUSTER, R., EBLING, M., FETTWEIS, G., FLINCK, H., JOSHI, K., AND SABNANI, K. An open ecosystem for mobile-cloud convergence. *Communications Magazine 53*, 3 (2015).

[64] SATYANARAYANAN, M., SIMOENS, P., XIAO, Y., PILLAI, P., CHEN, Z., HA, K., HU, W., AND AMOS, B. Edge analytics in the internet of things. *Pervasive Computing 14*, 2 (2015).

[65] SHI, W., CAO, J., ZHANG, Q., LI, Y., AND XU, L. Edge computing: Vision and challenges. *Internet of Things Journal 3*, 5 (2016).

[66] VASISHT, D., KAPETANOVIC, Z., WON, J., JIN, X., CHANDRA, R., SINHA, S. N., KAPOOR, A., SUDARSHAN, M., AND STRATMAN, S. FarmBeats: An iot platform for data-driven agriculture. In *Proceedings of the Symposium on Networked Systems Design and Implementation (NSDI)* (2017), USENIX.

[67] WANG, J., AMOS, B., DAS, A., PILLAI, P., SADEH, N., AND SATYANARAYANAN, M. A scalable and privacy-aware iot service for live video analytics. In *Proceedings of the 8th ACM on Multimedia Systems Conference* (2017), ACM, pp. 38–49.

[68] WANG, J., FENG, Z., CHEN, Z., GEORGE, S., BALA, M., PILLAI, P., YANG, S.-W., AND SATYANARAYANAN, M. Bandwidth-efficient live video analytics for drones via edge computing. In *Proceedings of the Third ACM/IEEE Symposium on Edge Computing* (2018), SEC '18, IEEE.

[69] YI, S., HAO, Z., ZHANG, Q., ZHANG, Q., SHI, W., AND LI, Q. Lavea: Latency-aware video analytics on edge computing platform. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing* (2017), SEC '17, ACM.