eCloud: A Vision for the Evolution of the Edge-Cloud Continuum

Joy Arulraj, Abhijit Chatterjee, Alexandros Daglis, Ashutosh Dhekne, and Umakishore Ramachandran, Georgia Institute of Technology

We present a holistic vision for the edge-cloud ecosystem, with the intent of spurring the creation of next-generation technologies for futuristic applications that operate at computational-perception speeds to convert sensed data to actionable knowledge.

If the past is an indicator of the future, what we do in our everyday lives will not change drastically. From time immemorial to the present, an average human life is defined by the same aspirational goals. Be productive citizens, raise a good family, educate children, enjoy life. In other words, what we do as a society will not change, but how we do it will. Increased automation will dramatically transform our daily lives—from transportation, to shopping, to entertainment.

Figure 1 depicts the infrastructure that will power future societal needs, providing the computational muscle for converting sensed information from devices to actionable knowledge and enabling the automation of everyday services. Fog/edge computing[1] extends the cloud's centralized utility computing model to quickly process sensor streams and filter uninteresting data at their source by provisioning computational resources in a geodistributed manner (for example, compute capability in a cellular tower) closer to end-user devices and sensors and by migrating computation and state commensurate their with application mobility.[2]

Digital Object Identifier 10.1109/MC.2021.3059737
Date of current version: 7 May 2021

TERMINOLOGY

We classify the computational continuum (processing, networking, and storage) into the following four strata.

0018-9162/21©2021IEEE

1. *Device*: a platform with sensing and limited processing capability [for example, vehicles, drones, Array of Things,[3] and augmented reality (AR)-enabled platforms], which simultaneously serves as a sensing and actuation destination for several applications. The devices are orchestrated by gateway nodes (for example, the edge nodes described in 2, below) and are typically not multitenant.
2. *Edge node*: a micro data center housed in a small-footprint location (for example, the central office of a telecommunication company, base of a cellular tower, or a closet in an office building), typically housing a few server racks. Edge nodes are multitenant and cater to the computational needs of devices in their immediate vicinity (that is, one network hop away).
3. *eCell*: a federated set of geographically adjacent edge nodes. The federation is dynamically determined based on the context of interest (COI) that is meaningful to a given application. Such a federation can be leveraged for system-level optimizations, such as balancing a load across the eCell's edge nodes, and for application-specific enhancements. For example, merging the information from multiple adjacent edge nodes would improve autonomous vehicle (AV) control decisions in the face of an emerging traffic situation spanning several miles (that is, an area exceeding any single edge node's coverage). Many latency-critical applications require horizontal communication among edge nodes, and the eCell terminology captures the federation of these edge nodes, sharing a specific COI for a given application. We assume that the latency associated with peer-to-peer communication among edge nodes within an eCell does not fluctuate significantly.
4. *Cloud*: a data center (for example, Amazon Elastic Compute Cloud) with virtually infinitely more resources compared to an edge node. The edge nodes access the

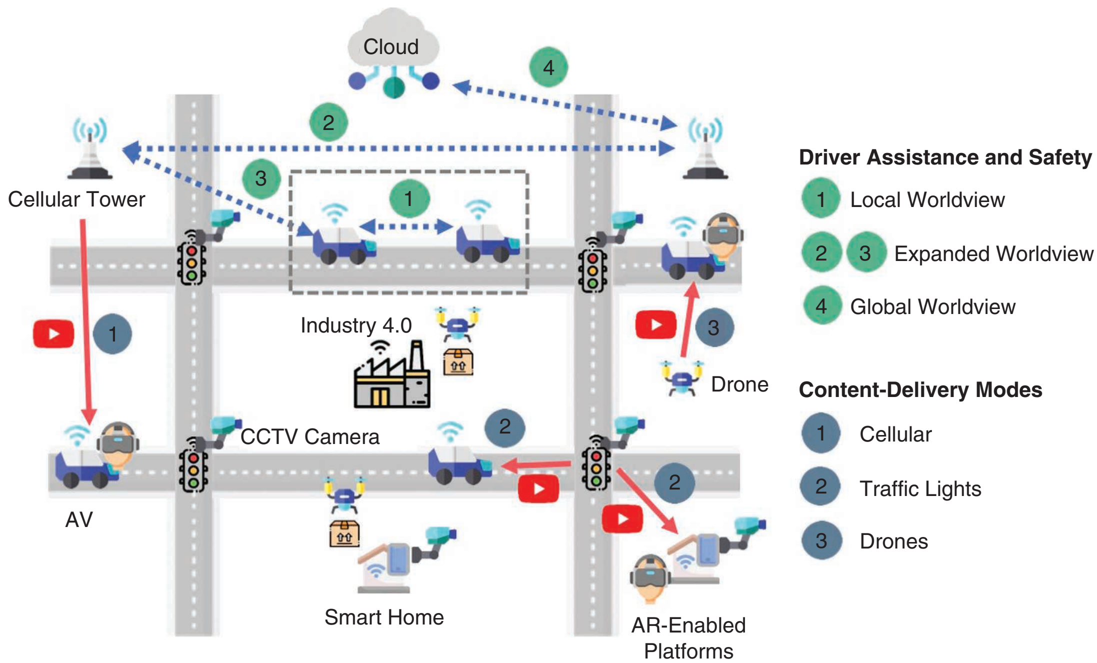

FIGURE 1. The infrastructure that transforms how a futuristic society performs everyday tasks. The devices [for example, autonomous vehicles (AVs)] are information producers and consumers, with edge nodes and the cloud providing the computational muscle. AR: augmented reality; CCTV: closed-circuit TV.

cloud via wide-area Internet, rendering edge-cloud communication latency unpredictable.

APPLICATION DRIVER: AV

We use AVs as a driver, grounding our vision for the evolution of the edge-cloud ecosystem. Our AV application's first aspect is collaborative vehicle control and traffic management from a safety as well as a transit-time perspective. The second aspect is onboard entertainment. We discuss these aspects in the next section.

Collaborative vehicle control and traffic management

Fully AVs[4], projected to be on the road in roughly a decade, will make short-term motion and long-term path- and route-planning decisions in a collaborative manner across a multitude of vehicles. To enable such collaboration, distributed multimodal sensing and data fusion need to be performed with vehicle-to-vehicle and vehicle-to-edge communication to allow each AV to reconstruct its immediate environment as accurately as possible. Existing vehicle-to-vehicle infrastructure[5] allows for short-range (~1,000 ft) 360° inter-AV communication. The information about position, velocity, acceleration, transmission status, and past trajectory is transmitted to vehicles in the vicinity.

Emerging 5G communication systems will allow other data modalities to be shared as well. For truly collaborative vehicle control and traffic management, a hierarchical traffic management methodology that manages AV movement across local, expanded worldviews as well as global geographical domains is necessary. At the hierarchy's lowest level, each AV would make its own short-range motion and path-planning decisions based on data received from other vehicles and edge nodes. Distributed, risk-averse, consensus-based algorithms will facilitate such decision making at the individual AV level. At the intermediate, expanded world-view level, the envisioned eCell will have access to vehicular-state information across wider geographical areas, which will be used to perform path planning that spans eCell-granularity-coverage zones. For example, the expanded worldview would contain locations of pedestrians occluded from the local worldview of an individual AV [see Figure 2(a)] or enable the detection of occluded traffic coming from the opposite direction [see Figure 2(b)]. Because of the rapid mobility of vehicles and objects of interest (for example, pedestrians), high-quality collaborative vehicle control at the expanded world-view level requires low-latency communication of emerging traffic situations. At the highest level of the traffic management hierarchy, end-to-end route planning will be performed in the cloud using global traffic data.

In-vehicle, anytime-anywhere video on demand

Video on demand (VoD) already dominates Internet traffic, accounting

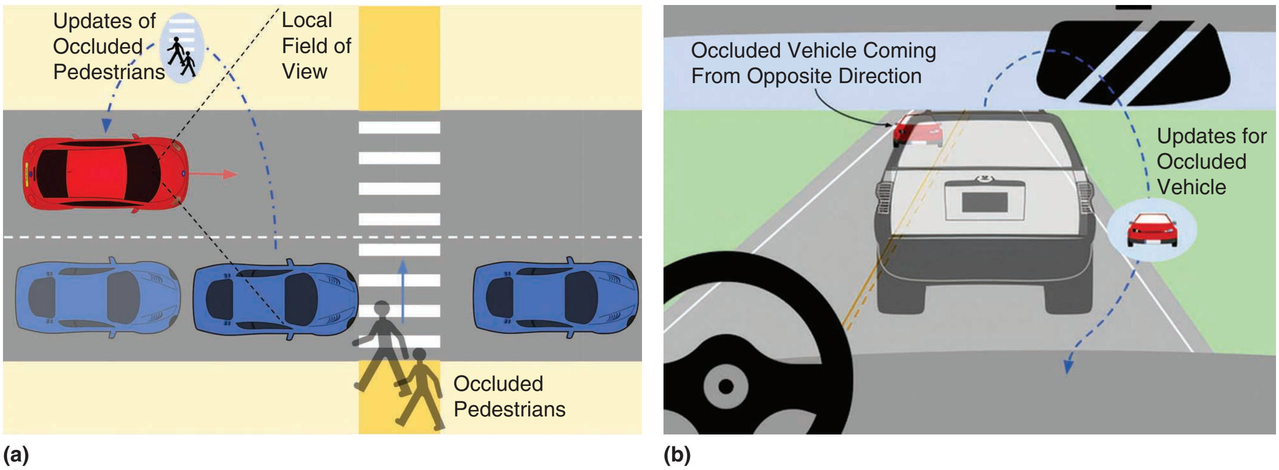

FIGURE 2. Edge-assisted AVs. The scenarios illustrate the importance of edge-assisted collaborative information sharing across AVs. (a) Occluded pedestrians and (b) occluded traffic.

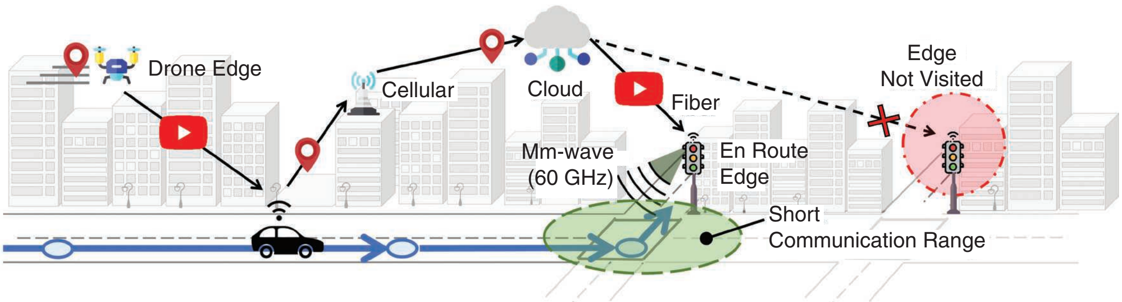

FIGURE 3. A VoD service. The AV shares its intended route. The video segments are prefetched at the edge nodes and delivered via fast mm-wave links when the AV arrives. The drones follow the AVs, delivering content as needed.[7]

for 65% of downstream bandwidth usage (51% in cellular networks).[6] The advent of AVs will inflate this demand as the vehicle transforms into an extension of the home or office. We envision an AV as a mobile entertainment hub, wherein passengers consume videos (for example, the latest news, movies, and so on), while commuting to their destination. Such an increase in high-mobility video consumption would be unsustainable on current cellular networks, requiring a rethinking of networks. However, video content presents a unique opportunity: VoD is latency tolerant. VoD benefits from supply elasticity by buffering larger chunks of video when surplus bandwidth is available and draining those buffers on bandwidth scarcity. Thus, VoD should be modeled as a static-content delivery problem instead of a real-time video-streaming problem.

Figure 3 shows the overall VoD delivery vision. We imagine edge nodes that prefetch per-AV video data, and the AVs download video bursts upon entering an edge node's range. The availability of the AV's route plan and its knowledge of current vehicular traffic removes the guessing game from prefetching and subsequently caching a video for other AVs that may need it in the future. As edge nodes become more prolific, including drones that can follow traffic,[7] VoD traffic can be increasingly offloaded to the edge, freeing up cellular bandwidth for real-time data.

A HOLISTIC VISION

The AV application, with its control and entertainment components, serves as an exemplar for different kinds of autonomous services for futuristic societal needs. We holistically discuss the research issues in supporting such services. As suggested in the literature,[8] software is the main obstacle to realizing futuristic applications such as AVs because hardware, namely, sensing technologies and compute power, has evolved rapidly. We go further and assert that a coordinated and interconnected research agenda spanning sensing and control algorithms, data management, and system architecture (encompassing system software, networking, and computer architecture) is essential to realizing our vision. eCloud is a distributed ecosystem for assembling the basic building blocks for supporting such futuristic services and requires innovation across the system stack presented in Figure 4. The remainder of this article is structured

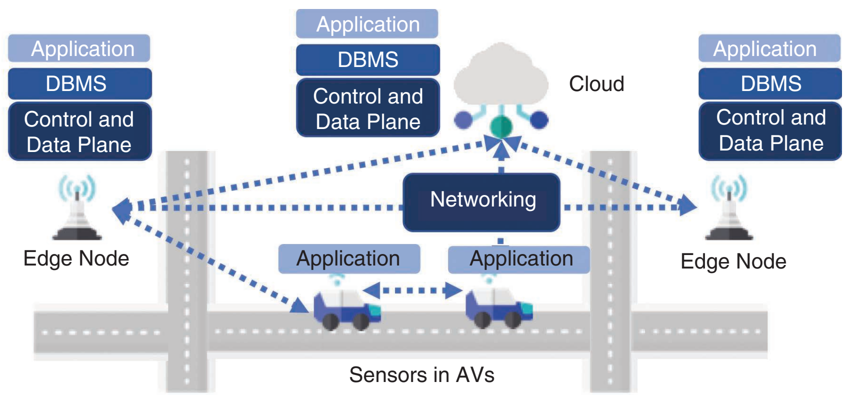

FIGURE 4. eCloud, an end-to-end, full-stack solution for the envisioned futuristic autonomous services. DBMS: database management system.

to demonstrate connectedness across the several research areas involved in eCloud's full-stack vision.

APPLICATION-LEVEL CHALLENGES

AV control and traffic management

Collaborative AV control and traffic management require problem solving at multiple levels. The AV itself performs short-range motion planning with the objective of minimizing transit time to the next waypoint on its path while reducing the risk to itself and other vehicles in its vicinity (called the *influence zone*) by its decisions.[9] There are several challenges to this problem. First, every vehicle must know the occupancy maps of where every other vehicle in its influence zone is likely to be within a prescribed time horizon. The overlapping occupancy maps for different vehicles indicate higher risks of collision at those locations. Steering and braking decisions, which determine such occupancy maps, must be made to minimize the aggregate risk of collision for all the vehicles concerned and therefore must be reached using distributed consensus. These decisions must factor into the failure of vehicles' electromechanical subsystems.[10] Given tight decision-making constraints, a consensus must be reached quickly with edge assistance (sub-100 ms).

Second, such decisions must remain reliable in the presence of edge-node failures and dynamically varying latencies of vehicle-to-vehicle and vehicle-to-edge communication. Ideally, all of the nodes within an eCell must have a consistent expanded worldview so that all of the vehicles in the eCell's region make mutually consistent path-planning and risk-averse decisions. As an AV moves, the expanded worldview of each passing eCell changes as well; therefore, there is a need to maintain logically consistent views at all of the edge nodes. In practice, this is difficult, and vehicle-control algorithms must be designed to tolerate logical inconsistencies by minimizing vehicle risk in a probabilistic manner. To minimize such inconsistencies, we expect coordination at the 50–100-ms range across an eCell's edge nodes.[11] Finally, low-level vehicle motion needs to be integrated with the expanded worldview (performed in eCells) and global route planning (performed in the cloud). The bottom line is that the latency and bandwidth requirements of AVs are moving targets; therefore, it is essential that 1) the execution environments in the edge nodes are agile and nimble and 2) the communication software stack for sharing information across each eCell's edge nodes is lean.

VoD

Today, most video applications stream content from cloud servers using Internet connectivity provided by the mobile device.[12] However, as video bandwidth requirements increase, the task of delivering high-quality content to mobile AVs becomes challenging for the existing infrastructure. With the envisioned eCloud architecture, video-quality improvements and bandwidth conservation will result from downloading content from edge nodes en route to the AV's planned journey. The video application will determine how, when, and from where content is fetched. The AV's video app will collaborate with the onboard path-planning module, obtaining real-time updates for the expected route and the timing information. The app shares this information with the app's cloud service, which determines the optimal set of edge nodes. The edge nodes prefetch the appropriate video segments that are ready for the arriving AV (1–2 GB of data). The video app is informed about data availability at the edge nodes and performs the appropriate network actions to facilitate fast video-segment downloads (1 Gbps for 10–15 s) for the passing AV. Due to finite edge-to-cloud and edge-to-AV bandwidth availability, the key research question is determining which AVs to serve from eCloud's edge nodes based on AV arrival times, video demands, and AV-edge contact times. The goal is to maximize data delivery while minimizing missed deadlines, thus maximizing eCloud's benefit.

DATA MANAGEMENT

eCloud manages data in support of the application layer across the computational continuum. The following are the illustrative concerns for the data management layer:

- *AV controls*: find paths from the current location to the desired destination in a spatial network. They also find road obstacles beyond the radio horizon of vehicle-to-vehicle communication systems.
- *VoDs*: find the set of edge nodes that the vehicle is expected to reach from its current location within a given time.

Spatiotemporal data management

eCloud stores the GPS records of AVs in a geodistributed database. The collaborative path-planning algorithms use this database to answer the aforementioned concerns for a given AV.

Canonical, spatiotemporal graph-processing algorithms have two limitations for addressing these concerns.[13] First, they are tailored for static graphs that do not evolve over time to adapt to environmental conditions. Second, they assume that the entire data set is locally available on the device. In contrast, eCloud answers these questions by processing the data spread across the computing continuum while operating on time-varying road networks.

Visual data management

Instead of merely storing the raw visual data (for example, videos from the AV's cameras), eCloud would also compute and store annotations from off-the-shelf deep learning models, such as labels from an image classifier.[14] These annotations have a significantly smaller storage footprint than do raw data and enable the application to quickly derive actionable insights from the data [for example, detecting occluded pedestrians, as shown in Figure 2(a)]. They can also be reused across different instances of the application, thereby lowering the computational demands placed on the edge nodes.

Opportunities and challenges

The space-time partitioning of trajectories; new specialized auxiliary-index structures; and the adaptations of canonical, spatiotemporal graph-processing algorithms are all promising approaches for improved computational efficiency and availability. Leveraging historical trajectories to derive a reachability index that keeps track of the set of road segments reachable within the desired time interval could drastically reduce the input–output operations for an eCell-specific spatiotemporal index.

State-of-the-art deep learning models contain dozens of computational layers that maximize prediction accuracy; however, this comes at the cost of increased latency. The accuracy requirements are application specific: for example, identifying occluded pedestrians requires the highest accuracy while aggregating the number of AVs passing an intersection does not. Commensurate with requirements, there is an opportunity to pick the appropriate model and trade off accuracy for latency by short-circuiting the inference.[15]

SYSTEM ARCHITECTURE

eCloud is a large-scale distributed system with unique characteristics, bringing new challenges and innovation opportunities. The key challenges include extreme resource heterogeneity at the device and edge layers, device mobility, and potential resource fragmentation at the edge layer. Fragmentation is fueled by technological evolution: although emerging 5G technology offers 10–100-times-higher bandwidth and 5–10-times-lower latency than 4G, each 5G access point has smaller area coverage. Future technologies (6G and beyond) will exacerbate this phenomenon. Consequently, an edge infrastructure covering a given area will comprise more edge nodes, resulting in increased resource fragmentation and leading to higher vulnerability to load imbalance, therefore requiring higher per-node resource overprovisioning. One approach used to address these challenges is system support for effective resource pooling to enable uniform utilization of the aggregate edge resources while meeting scalability and response-latency requirements. Conceptually, resource pooling enables the dynamic composition of a virtual "supernode" by allowing individual edge nodes to borrow resources from neighboring nodes, thus enabling cost-effective aggregate resource provisioning and increased tolerance to localized load spikes.

eCloud builds on the concept of eCells: groups of edge nodes capable of pooling their resources together. Figure 5 depicts two eCells. eCell boundaries are fluid and are dynamically (re)configured as a function of an application's computational and latency requirements, device-generated load, and resource availability across a geographical region's edge nodes. Establishing such a logical federation of distributed resources requires a control plane that considers application-specific requirements, network conditions, and global resource availability. Within an established eCell, a fast data plane determines which of the eCell's nodes will handle each incoming request.

eCloud's control plane

eCloud's control plane comprises two logical components: resource orchestration and monitoring. Orchestration is further decomposed into two levels with different scopes, per application and infrastructure, as depicted in Figure 5. Application-level orchestrators (AppOs) control resources for a given application (for example, AV path planning). An AppO manages the lifecycle of that application's multiple instances across edge nodes, including initial deployment and continuous adaptation, to ensure that the application's quality goals are met. The AppO uses the application's latency requirements to define eCell granularity. A logically centralized infrastructure-level orchestrator (InfraO) coordinates multiple AppOs to ensure an infrastructure-wide balanced load and resolve potential inter-AppO resource-allocation conflicts.

A device's entry point to eCloud is its home node—typically the device's closest edge node. The home node's data plane determines which node in its eCell will service each incoming request. If the home node does not belong to an eCell and decides to expand its resource pool by joining one, it establishes an AppO, which submits a resource request to the InfraO. The InfraO makes a resource-allocation decision based on overall (inter-AppO) resource utilization in the broader area. eCell deployment is an infrequent control operation, triggered only when an application is deployed on an edge node for the first time or when eCell resource rearrangement is required.

Each edge node of an eCell is constantly monitoring the performance metrics that are indicative of whether a deployed application meets its quality goals. Aggregated metrics are periodically presented to the application's AppO, which determines whether an eCell reconfiguration by the InfraO is required.

eCloud's data plane

Each edge node of an eCell may decide to leverage the capability of resource pooling by deflecting a request received from a device in its range to a peer node for servicing, instead of servicing it locally. Making this approach practical requires making such decisions rapidly while considering local compute load conditions, network conditions, and the availability of other nodes in the eCell. Given that deflection decisions will be most needed when a node is experiencing an increased load, the mechanism making such decisions should remain insensitive to the queuing effects that introduce long, unpredictable delays.

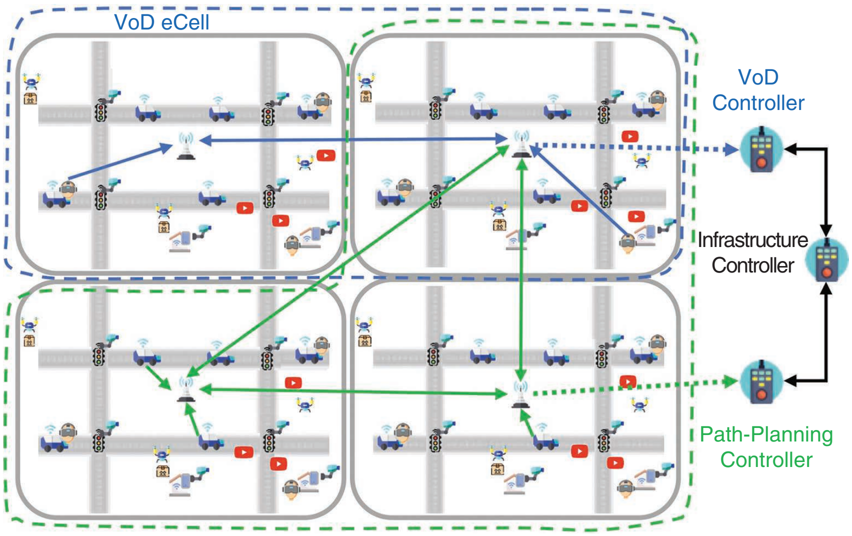

FIGURE 5. Dynamic eCells. A grid with four edge nodes dynamically grouped together into two eCells (dashed lines): VoD and path-planning eCells. The edge nodes within an eCell are federated and can distribute the load among them. The dotted arrows indicate the performance-monitoring metadata that are sent to each eCell's dedicated application-level orchestrator controller.

Opportunities and challenges

eCloud's system architecture calls for research efforts on the unique challenges of a geodistributed, heterogeneous distributed system with strict quality goals. Although it is a new system architecture, eCloud should lay its foundations on the mature technologies developed in the context of distributed systems and data centers. Existing container ecosystems are a promising starting point for application deployment at the edge as rapid deployment and portability are key. Although ongoing research efforts are focused on addressing the cold-start problem of containers, eCloud further urges pushing the envelope on that front as its targeted applications come with particularly stringent latency requirements.[16]

The malleability of eCells and the internode collaboration within an eCell will necessarily involve state replication/migration, raising data-consistency concerns. The differences in application domain requirements, infrastructure deployment, and failure modes warrant revisiting the large body of previous work on managing data consistency in data center environments.[17] Finally, while eCloud's control plane will bear most of the complexity, the data plane will be the primary performance determinant. Stringent latency goals require any intra-eCell request-deflection decisions to be rapid, especially when a node is temporarily overloaded (which is when resource pooling shines). Given that deflection decisions are largely independent from each application's specific business logic, it is worth considering specialized hardware that filters and acts on incoming messages within a fixed latency to avoid detrimental queuing effects. Data center practices can also be leveraged in the design of eCloud's data plane. For instance, deflection decisions need not be binary (that is, a request can be processed on the home node, a remote node, or both): latency-tolerant techniques from data centers[18] are highly relevant.

NETWORKING

An interesting dichotomy exists between AV safety and control, and in-vehicle VoD. eCloud relies on a latency-sensitive data transfer to and from the AV for safety-critical control and path planning. Compared to AV control, VoD requires a high-bandwidth connection (>1 Gb/s) but can tolerate significant latencies because data can be easily prefetched.

Low-latency data transfer

Every AV generates large quantities of sensor data for its own control and navigation use. The aggregation of these data into bounding boxes describing various entities that the AV currently observes is done in vehicle. eCloud then streams it to local edge nodes over 5G millimeter-wave (mm-wave) links. eCloud's control plane enables the knowledge of edge-node locations and connection information, reducing link-initiation delays as AVs move. The edge nodes, when computing the extended worldview, broadcast this information using rateless encoding,[19] which can be received simultaneously by other AVs without an explicit link or flow negotiations and at different distances from the edge node.

ABOUT THE AUTHORS

JOY ARULRAJ is an assistant professor in the College of Computing, Georgia Institute of Technology, Atlanta, Georgia, 30332, USA. His research focuses on the design and implementation of database management systems. Arulraj received a Ph.D. from Carnegie Mellon University in 2018. He is a Member of IEEE. Contact him at arulraj@gatech.edu.

ABHIJIT CHATTERJEE is a professor in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, 30332, USA. His research interests include the design of self-aware and safe autonomous robots and transportation systems. Chatterjee received a Ph.D. from the University of Illinois at Urbana-Champaign in 1990. He is a Fellow of IEEE. Contact him at abhijit.chatterjee@ece.gatech.edu.

ALEXANDROS DAGLIS is an assistant professor in the College of Computing, Georgia Institute of Technology, Atlanta, Georgia, 30332, USA. His research interests lie in computer architecture with a focus on data centers and cloud computing. Daglis received a Ph.D. from Ecole polytechnique fédérale de Lausanne in 2018. He is a Member of IEEE. Contact him at adaglis@gatech.edu.

ASHUTOSH DHEKNE is an assistant professor in the College of Computing, Georgia Institute of Technology, Atlanta, Georgia, 30332, USA. His research interests include wireless networking, wireless sensing, and mobile computing. Dhenke received a Ph.D. from the University of Illinois at Urbana-Champaign in 2019. He is a Member of IEEE. Contact him at dhekne@gatech.edu.

UMAKISHORE RAMACHANDRAN is a professor in the College of Computing, Georgia Institute of Technology, Atlanta, Georgia, 30332, USA. His research interests include parallel and distributed systems with an emphasis on edge computing. Ramachandran received a Ph.D. from the University of Wisconsin–Madison in 1986. He is a Fellow of IEEE. Contact him at rama@gatech.edu.

High-bandwidth data transfer

In-vehicle VoD requires dedicated high-bandwidth download links. A key enabler is the vast data-rate difference between peak download throughput using next-generation mm-wave networks (multiple gigabits per second) and the consumption data rate of video playback (a few megabits per second). Therefore, a few edge nodes can satisfactorily cater to a large number of AVs. The edge node would fetch the correct video chunks over a fiber-optic connection from the cloud. Once an AV arrives within range, the edge node could rapidly transfer the video content to the AV using mm-wave links.

Challenges and opportunities

A continuous, collision-free upload of AV digests to the edge nodes is a challenging networking problem. Similarly, robust broadcasts are difficult to achieve. In the VoD use case, given short contact times, it is important to swiftly initiate and tear down the edge-AV connection. However, today's mm-wave devices must first run a time-consuming peer scan to steer directional mm-wave beams.[20] The opportunities for sharing the wireless medium are present due to the availability of real-time location information from AVs to automatically steer the mm-wave beam. Throughput variations as the AV approaches the edge node and then departs could be precalculated. Finally, utilizing the eCell architecture allows for horizontal content movement across edge nodes. Thus, if an edge node is unable to deliver the requested data to the AV, the VoD AppO's control plane could notify the eCell's next edge node, and video segments could be transferred to it through the data plane.

Autonomy for everyday services will be the watchword for a futuristic society. Several technological pieces spanning the entire stack, that is, control algorithms, data management, system architecture, and networking, must come together to enable the transformation of services from human intensive to fully autonomous. We identified the challenges and opportunities of these technologies using an AV as the driver application. Although the AV application is used as an exemple, the system stack's components are general and can serve other verticals, including 1) safety-critical applications (for example, surveillance), 2) emergency response (man-made and natural disasters), and 3) community engagement (real-time AR-enabled social networking).

REFERENCES

1. M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, 2009. doi: 10.1109/MPRV.2009.82.
2. K. Hong, D. Lillethun, B. Koldehofe, B. Ottenwaelder, and U. Ramachandran, "Mobile fog: A programming model for large-scale applications on the Internet of Things," in *Proc. Workshop Mobile Cloud Comput.*, 2013, pp. 15–20.
3. "Urban center for computation and data." Array of Things (AoT). http://arrayofthings.github.io/ (accessed Mar. 16, 2021).
4. A. Eskandarian, *Handbook of Intelligent Vehicles*, vol. 2. New York: Springer-Verlag, 2012.
5. J. Barbaresso et al., "USDOT's intelligent transportation systems (ITS) strategic plan 2015-2019," Transportation Research Board, Washington, D.C., Tech. Rep. FHWA-JPO-14-145, 2014.
6. "Cisco annual Internet report (2018–2023) white paper," Cisco, San Jose, CA, White Paper, 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html
7. A. Dhekne, M. Gowda, and R. R. Choudhury, "Extending cell tower coverage through drones," in *Proc. 18th Int. Workshop Mobile Comput. Syst. Appl.*, 2017, pp. 7–12. doi: 10.1145/3032970.3032984.
8. K. Heineke, P. Kampshoff, A. Mkrtchyan, and E. Shao, "Self-driving car technology: When will the robots hit the road?" McKinsey & Company, Automotive & Assembly, May 2017. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/self-driving-car-technology-when-will-the-robots-hit-the-road
9. A. Best, S. Narang, D. Barber, and D. Manocha, "Autonovi: Autonomous vehicle planning with dynamic maneuvers and traffic constraints," in *Proc. 2017 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pp. 2629–2636. doi: 10.1109/IROS.2017.8206087.
10. M. I. Momtaz, A. Chandramouli, and A. Chatterjee, "Concurrent error detection in embedded digital control of nonlinear autonomous systems using adaptive state space checks," in *Proc. Int. Test Conf.*, 2020, pp. 1–10. doi: 10.1109/ITC44778.2020.9325229.
11. M. Abdallah, S. Abdallah, H. Parisa, L. Adel, and B. Richard, "Edge-enabled V2X service placement for intelligent transportation system," *IEEE Trans. Mobile Comput.*, early access, Jan. 2020. doi: 10.1109/TMC.2020.2965929.
12. "Ericsson mobility report," Ericsson, Stockholm, Sweden, June 2020. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/june-2020/mobile-traffic-by-application-category
13. N. Mamoulis, H. Cao, G. Kollios, M. Hadjieleftheriou, Y. Tao, and D. W. Cheung, "Mining, indexing, and querying historical spatiotemporal data," in *Proc. 10th ACM SIGKDD Int. Conf. Knowledge Discovery Data

Mining, 2004, pp. 236–245. doi: 10.1145/1014052.1014080.
14. X. Tiantu, L. M. Botelho, and F. X. Lin, "Vstore: A data store for analytics on large videos," in *Proc. 14th EuroSys Conf.*, 2019, pp. 1–17. doi: 10.1145/3302424.3303971.
15. J. Cao, R. Hadidi, J. Arulraj, and H. Kim, "Video analytics from edge to server: Work-in-progress," in *Proc. Int. Conf. Hardware/Software Codesign Syst. Synthesis Companion*, 2019, pp. 1–2. doi: 10.1145/3349567.3351733.
16. A. Hall and U. Ramachandran, "An execution model for serverless functions at the edge," in *Proc. ACM/IEEE IoT Design Implementation (IoTDI)*, 2019, pp. 225–236. doi: 10.1145/3302505.3310084.
17. H. Gupta, Z. Xu, and U. Ramachandran, "DataFog: Towards a holistic data management platform for the IoT Age at the network edge," in *Proc. USENIX Workshop on Hot Topics Edge Comput. (HotEdge)*, 2018.
18. J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, 2013. doi: 10.1145/2408776.2408794.
19. J. Castura and Y. Mao, "Rateless coding over fading channels," *IEEE Commun. Lett.*, vol. 10, no. 1, pp. 46–48, Jan. 2006. doi: 10.1109/LCOMM.2006.1576565.
20. I. Mavromatis, A. Tassi, R. Piechocki, and A. Nix, "mmWave system for future ITS: A MAC-layer approach for V2X beam steering," in *Proc. IEEE Veh. Technol. Conf.*, 2017, pp. 1–6. doi: 10.1109/VTCFall.2017.8288267.

Digital Object Identifier 10.1109/MC.2021.3072509