ClairvoyantEdge: Prescient Prefetching of On-demand Video at the Edge of the Network

Manasvini Sethuraman* , Anirudh Sarma*, Adwait Bauskar, Ashutosh Dhekne and Umakishore Ramachandran
College of Computing
Georgia Institute of Technology, Atlanta, Georgia 30332

***Abstract*—On-demand video contributes a large fraction of the data traffic on mobile networks. This share is expected to increase even more drastically in the coming years. While the cellular infrastructure is continuously evolving to keep pace with this increasing demand, it is necessary to ensure that sufficient bandwidth is reserved for other latency-sensitive real-time applications like video conferencing and multiplayer video games. A tangible approach involves reducing on-demand video load on cellular networks, especially from users on the move. We see an opportunity for cellular load reduction using edge nodes based on two observations: (1) video streaming is mostly a download-only operation with sequential data access; and (2) short-range mmWave links can deliver an extremely high throughput for nearby recipients of data. The knowledge of the user's planned travel route creates opportunities for *prescient prefetching* and delivering the content as the vehicle passes through *just in time*, using mmWave devices on *en route* edge nodes.**

***ClairvoyantEdge* is a novel networked system infrastructure that leverages inter-edge node communication and the knowledge of users' trajectories to plan and deliver buffered video segments to the vehicles passing by. To evaluate *ClairvoyantEdge*, we built a comprehensive end-to-end emulation-based workflow that incorporates *in situ* field measurements of mmWave links into our own homegrown emulation framework. With a minuscule 0.12% coverage of a $46\,km^2$ geographical area employing 20 edge nodes distributed in that area providing short-range mmWave access to passing vehicles, we achieve an average reduction of up to 21% in cellular bandwidth usage for video downloads, using a real-world workload comprising 758 vehicles. Our results validate the promise of *ClairvoyantEdge* for incorporation in future edge infrastructure evolution.**

***Keywords*-Edge Computing, mmWave, video streaming**

I. Introduction

An exceedingly large amount of Internet traffic is being consumed by mobile devices. Today it is estimated that mobile data traffic is approximately 49 Exabytes (EB), which is expected to increase five-fold to around 273EB by 2026 [10]. Of the 49EB of data consumed today, 66% is attributed to videos, and this fraction is expected to increase to 77%. This statistic is startling because, mobile networks, which were originally created to carry real-time data (audio and video conferencing), are being increasingly used for consuming *static* videos with significantly higher latency tolerances, and are yet sharing the total available bandwidth

*Both authors contributed equally to this work

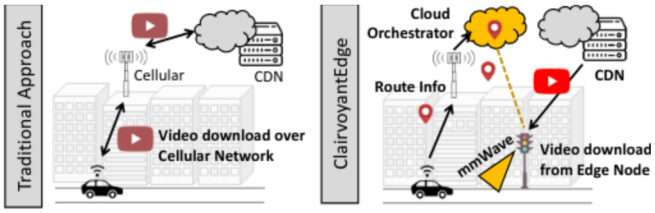


Figure 1: *ClairvoyantEdge* enables a road-side edge node to deliver video content to users over fast mmWave links. Users share their travel route to enable prefetching of content at edge nodes, freeing up cellular bandwidth.

with real-time traffic. For example, YouTube and Netflix videos occupied 21% of the total mobile data traffic in 2019, while real-time video conferencing and audio calls continued to suffer [15], [22]. Evidently, if the portion of data bandwidth used up by videos can be reduced, more resources will be available for carrying real-time traffic. In this work, we ask the question: *Can we ride on the emerging Edge Computing evolution to deliver video data to mobile users without consuming the last-mile real-time cellular bandwidth?* We see an opportunity to alleviate the pressure on cellular networks due to static video data by utilizing the edge infrastructure [41] that is already being deployed city-wide in support of 5G roll-out [4], [7], [8], [41]. From the Cloud provider side, the evolution of edge infrastructure has primarily been motivated by the need to satisfy the requirements of latency sensitive applications such as Autonomous Vehicles (AVs) and Augmented and Virtual Reality (AR/VR) [19], [42]. From the cellular provider side, Multi-Access Edge Computing (MEC) is deemed as a way of statistically multiplexing the computational resources needed for processing cellular data before sending the data on to the wide-area Internet [34]. Thus there is a confluence of technology indicators influencing the evolution of the edge infrastructure. The thesis of *ClairvoyantEdge* is to ride on this evolution to serve video data from the edge infrastructure to passing vehicles by augmenting the edge with short-range mmWave links. In order to use edge nodes for prefetching video content, two primary questions must be answered: (1) *which* edge nodes should prefetch *what* content so that mobile users will benefit? and (2) how will edge nodes transfer the prefetched content to user's devices? Predicting user behavior is complex. We instead leverage

2022 IEEE/ACM 7th Symposium on Edge Computing (SEC) | 978-1-6654-8611-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SEC54971.2022.00010

978-1-6654-8611-8/22/$31.00 ©2022 IEEE
DOI 10.1109/SEC54971.2022.00010

the fact that users frequently use map applications on their mobile devices when on the move. The route information from the mobile device can be directly utilized for *knowing* (instead of predicting) which edge nodes the user will visit at what time in the future. Therefore, the first question can be answered with reasonable accuracy. The designated edge nodes on the user's travel route can prefetch the required content in preparation of the user's arrival. As for the second question, the reducing video traffic handled by cellular network necessitates the creation of an *edge-local* infrastructure to enable wireless download of video data. The main requirements for this infrastructure would be (1) To allow *interference-free download* of video data from the edge node to a user in a moving vehicle i.e., interference should be minimized with existing cellular technology, as well as with different edge nodes in the neighborhood. Hence, a short-distance link is desirable. (2) To significantly reduce cellular video-load a *high-throughput* edge-to-user link is desirable. mmWave links satisfy both requirements. They are high throughput (multi-Gbps) and short-distance (few meters), and operate in the unlicensed 60 GHz spectrum [26]. Inclusion of 60 GHz mmWave antennas in 5G small cell deployments has been discussed in the past [32], [37]. We therefore, propose the use of mmWave links for accomplishing edge-to-user on-demand video delivery.

Fig. 1 shows our overall envisioned system. When a mobile user requests a video from the origin server (or CDN), they will also share their intended travel route with a *cloud orchestrator*. The cloud orchestrator will then contact the en route edge nodes (with mmWave capability) and negotiate with a subset of them to participate in prefetching content for this user. When the user moves within communication range of a participating edge node, the user's device will start downloading the expected video segments from that edge node. Extremely fast Gbps-speed mmWave links ensure a bulk transfer of a substantial number of video segments to fill up the user device's buffers. We expect that even within a short contact time (of a few seconds), an edge node will download substantial video data to the user's device. The device will thus have enough buffered video content for playback until it reaches the next edge node. However, if a user cannot be served by an en route edge node (either due to prior download commitments to other users or due to depletion of the user device's video buffers prior to reaching the next edge node), the device falls back to cellular connectivity to ensure that the user experience will never degrade below the cellular performance.

In this work we develop *ClairvoyantEdge*—an end-to-end system for on-demand video data delivery through a geo-distributed edge infratructure. The main contributions of this work are:

1) A novel system architecture, and an end-to-end implementation[1] that combines user's route information to *prefetch* video segments to a set of en route edge nodes, and deliver them using short-range mmWave links to the user. The elements of the architecture include: (a) a cloud orchestrator that takes video requests from mobile users and their routes to create *space and time aware* prefetch and download schedules for the edge nodes; and (b) a peer-to-peer content sharing optimization that allows sharing of previously prefetched video segments among edge nodes to reduce the pressure on backhaul networks and the load on origin content servers.
2) A detailed performance evaluation comprised of (a) field study of real mmWave links to develop a distance-download profile for incorporation into the system architecture; (b) validation of the implementation of *ClairvoyantEdge* and quantification of the expected reduction in cellular bandwidth usage for video downloads; and (c) end-to-end evaluation using realistic vehicular mobility traces to showcase the performance of *ClairvoyantEdge*.

It should be noted that while *ClairvoyantEdge* caches prefetched content at edge nodes for potential future use, we do not claim novelty on the caching strategy itself since there is considerable prior art in that space [16], [25]. In this sense, the core of *ClairvoyantEdge*, i.e., *prescient prefetching*, should be considered complementary to such prior art.

The rest of the paper is organized as follows: §II discusses the related work. §III describes the design of *ClairvoyantEdge*. Then §IV discusses the implementation details, followed by evaluations in §V. §VI discusses some of the limitations of the system and proposes future work and finally, §VII concludes the work.

II. Related Work

In this section, we review some of the prior art with regards to techniques for on-demand video content delivery at the edge, and backhaul traffic reduction. Resources at the edge, coupled with the fact that the popularity of video content follows a long tail distribution (less than 5% of videos on YouTube are frequently accessed [12]), make edge caching a feasible idea for on-demand video streaming.

Offloading last mile delivery. Offloading access to content through Wi-Fi has been explored before [9], [38], however mmWave brings new challenges and opportunities. Multi-modal content delivery over both Wi-Fi (2.4 and 5 GHz) and 5G has been discussed by Sun et al. [38] for 360 video streaming. They collect throughput traces from an 802.11ad testbed to demonstrate a proof of concept for high bandwidth 360 video delivery over mmWave. However, their system is intended for stationary users. Our system fundamentally incorporates mobility, while offering last mile delivery over unlicensed mmWave links.

[1]Source code: https://github.com/Manasvini/clairvoyant2

Backhaul traffic reduction. Ma et al. [20] incorporate geographical and temporal skews in video popularity in their caching strategy at the edge for on-demand videos, with the goal of reducing load on origin servers. Reducing traffic on the backhaul links is the main focus for Rhee et al. [39], where different channels are used for content dissemination from CDN servers to edge nodes, and content delivery to end users from the edge nodes (typically over Wi-Fi) for making efficient use of backhaul links. They do not consider inter-edge node links for content sharing. While Park et al. [25] allow for horizontal content transfer between edge nodes to reduce the stress on the backhaul, content delivery to the user is however done over licensed spectrum. In contrast, *ClairvoyantEdge* enables both backhaul traffic reduction as well as reduction of last-mile cellular traffic.

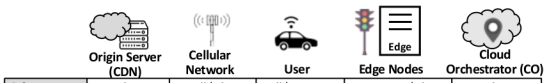
Location-aware caching. The work proposed by Santos et al. [33] uses location prediction of mobile users in a train to prefetch and cache content on virtualized CDNs collocated with 5G base stations. Video delivery to users is accomplished via licensed 5G spectrum when the users (in the train) get close to the caching edge site. While this work bears some similarity to ours, *ClairvoyantEdge* is for a more general setting wherein mobile users are in independent vehicles and the data delivery uses unlicensed mmWave links.

Improving Quality of Experience (QoE). Caching video segments at the edge is viewed as a means to improve the user's QoE by supporting higher bitrates and fewer switches between different resolutions [18], [40]. Edge infrastructure is also used to gain a better understanding of the user's bandwidth needs as a means to improve QoE [29]. Bayhan et al. [6] discuss optimizing content delivery to several users connected to a single Wi-Fi access point. In comparison, *ClairvoyantEdge* focuses on reducing cellular bandwidth use for video content. Our solutions assume that video segments are already available at a set quality.

Leveraging Information Centric Networking and 5G. Ge et al. [11] and Psaras et al. [28] propose an information centric networking based caching scheme for 5G networks, where the caching decisions are offloaded to the underlying network by leveraging Named Data Networking (NDN) principles, for reducing traffic on backhaul links. While information centric networking introduces new ways to address content, *ClairvoyantEdge* causes far less disruption to the underlying content access and rather works using existing network architectures such as web-caches and CDNs.

III. System Design

In this section, we discuss *ClairvoyantEdge*'s design. The core idea of *ClairvoyantEdge* is to use high-throughput mmWave links between edge nodes and user devices for video download. The main components in *ClairvoyantEdge* are shown in Fig. 2. The exchange of video segments between components is termed as the "Data Plane", and



Link	Fiber	Cell/Fiber	Cell/mmWave	mmWave/Fiber	Fiber
Data Plane	-	Video delivery fallback	Video downloads	Video prefetching & delivery	-
Control Plane	-	CO-user interface	Edge- & CO-user interface	Edge- & CO-Edge interface	CO-{user,Edge} interface

Figure 2: Participating networked components of *ClairvoyantEdge*; their communication technology, and their data- and control-plane actions.

exchange of operational metadata as the "Control Plane". We first discuss the rationale behind using user's location information for deciding pertinent edge nodes, then describe the communication links between components, and the rationale for mmWave last-hop links. We then describe in detail, the control plane and the data plane, and then conclude the section with data plane optimizations.

A. Tracking User Mobility

ClairvoyantEdge elevates user mobility to a first class citizen status by sharing the user's travel route with the cloud orchestrator. Given that about 77% population already uses some map application regularly [24], this requirement of sharing the route information with a trusted service is not restrictive. Instead of *predicting* the user's whereabouts, the cloud orchestrator uses *knowledge* of the user's route to plan the data delivery using geo-distributed edge nodes.

B. Communication Links

The user device communicates with the cloud orchestrator using cellular connectivity to initiate control plane actions. Then, the user device connects to the proximal edge nodes to download video segments using mmWave links. Edge nodes participating in handling the user's video request prefetch video segments to their local storage over wired links. The communication between the edge nodes and the WAN entities (cloud orchestrator, and CDNs or origin servers), as well as horizontally between edge nodes is effected using high-speed wired network connectivity. The user device uses cellular connectivity for its data plane actions as a last resort when edge nodes fall short of meeting the user's needs. Fig. 2 summarizes the characteristics of the communication links in *ClairvoyantEdge*.

C. mmWave for Last-Hop Connectivity

Currently, on mobile networks the last-hop is served using licensed cellular frequencies. However, an important requirement for our system is to free up this cellular band for real-time traffic. Hence, while cellular frequencies can cover a large area and also provide an acceptable throughput, it is not an acceptable choice for *ClairvoyantEdge*. One option is to use unlicensed Wi-Fi spectrum either in the $2.4\,GHz$ or $5\,GHz$ bands. However, Wi-Fi offers limited throughput. Furthermore, with Wi-Fi, providing throughput assurances is difficult in uncontrolled settings since Wi-Fi implements carrier sensing while being omnidirectional. Sommers et al. report that Wi-Fi throughput is less predictable than

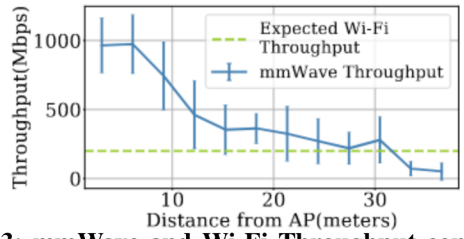


Figure 3: mmWave and Wi-Fi Throughput comparison.

cellular [36]. It has also been shown that interference is problematic for outdoor Wi-Fi coverage [30], [43].

In contrast, the recent advances in mmWave technology at the $60\,GHz$ unlicensed band provide substantial throughput advantage. mmWave uses directional wireless transmission, making it possible to transmit data to multiple spatially separated user devices. While mmWave has a limited range similar to Wi-Fi, it supports much higher throughput, making it suitable for bulk data transfer. Fig. 3 shows the comparison between Wi-Fi and mmWave throughput observed over various distances using the state-of-the-art Netgear Nighthawk X10 AD7200 router (derived from experiments detailed in §V-A). The mmWave throughput is at least three times that of Wi-Fi when close to the router. As the distance from the router increases, the throughput falls to below Wi-Fi levels around $30\,m$. Thus to effectively use mmWave frequencies, transmission distance should be limited to within $30\,m$.

This transmission range limit can be used to our advantage in *ClairvoyantEdge* since a user only needs to connect to an edge node sporadically to download video data from it when they are in close proximity to the edge node. The limit also indicates that separate mmWave links can exist beyond a detectable range of $\approx 45m$, allowing spatial reuse without interference. Cellular connectivity is primarily used for exchanging control information with the rest of the *ClairvoyantEdge* system, and as a fallback mechanism when data downloaded from edge nodes is insufficient for the playback during the travel interval between consecutive edge nodes, as illustrated in Fig. 4.

D. The Control Plane

The control plane (Fig. 5) is realized via the cloud orchestrator, a service that assigns tasks to edge nodes, and monitors and course corrects the assignments based on users' progress in their journeys.

1) Cloud Orchestrator: Control Functions

The cloud orchestrator, henceforth called CO, maintains the state of every edge node in the system as well as all user requests it receives. When a new request arrives, the CO first creates a list of edge nodes that are en route. An edge node is assigned to serve a user's request if the following conditions are met: (1) the mmWave device on the edge node is available to serve the user during the user's expected contact time, (2) the edge node has sufficient time to prefetch the data that the user requested from the CDN before the user arrives in its vicinity, and (3) the edge node has enough space to store the fetched data. Once these checks have been performed, the CO assigns the video segments for download to the participating edge nodes and informs them of the expected contact start and end times for the specific user. The CO returns a *source-list* (similar to the manifest file created by the MPEG-DASH [35]) to the user, which contains a list of edge nodes that the user can download video segments from. Fig. 6 depicts the operations of the CO.

The CO also determines where the respective video segments must be prefetched from by the edge nodes. If the video segments are already present in a different edge node, maybe as a result of fetching this video for a different user, CO instructs the edge node to fetch it from that peer edge node. If not, then the edge node must perform a fresh video prefetch from the CDN. Fetching from a peer edge node is a *hint* and not an *absolute*. An edge node is free to discard segments as part of its local cache management policy (see §III-E2).

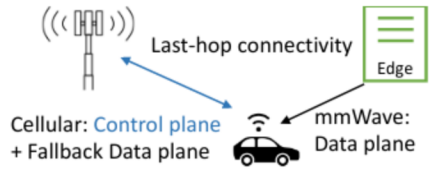


Figure 4: The user device connects with edge nodes over mmWave links for the bulk of the video segments. Cellular connectivity is used for control functions, and only as a fallback for data.

As the user travels along their route, CO continuously receives updates from the edge nodes about segments which were successfully delivered. If a user does not manage to download the promised video segments from the edge node, CO directs the next edge node on the user's route to prefetch the un-delivered segments from the previous edge node, and updates the user's source-list to retry fetching from the

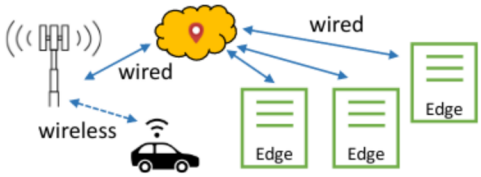


Figure 5: The control plane in *ClairvoyantEdge* performs metadata communication. The user device communicates over wireless cellular links, while other components communicate over wired links.

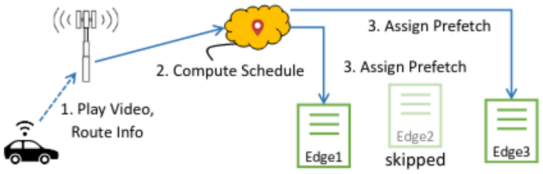


Figure 6: The user sends a video request and its travel route to the cloud orchestrator, which decides to assign a few edge nodes to prefetch video segments.

subsequent edge node. *CO* then updates its own metadata to indicate that the video segments are now available at both edge nodes. However, *CO* ignores updates from an edge node if the user's device no longer requires those video segments by the time the user arrives at the subsequent edge node. The *CO* also receives information from the user about any changes to their route, upon which it invalidates the previous segment assignments to edge nodes, and creates new ones. These actions are dubbed *reconciliation mechanism.*

2) User's Device: Control Functions

A map application on the user's device supplies waypoint and expected timing information to *CO*. The user's device also sends the information about the requested video to *CO*. The *CO*, based on the user-supplied route, informs the appropriate edge nodes about the expected arrival time and contact time, along with the video segments to prefetch. The interaction between the user device and *CO* is illustrated in Fig. 7.

In the event that the user's device is unable to download all the promised data from an edge node into its storage (e.g., a browser cache or equivalent), owing either to contact time being insufficient or the vagaries of mmWave transmissions, the user's device expects to receive those video segments from the subsequent edge node. If the playback buffer of the video player empties before the user arrives at the subsequent edge node, the segments are instead fetched over cellular backhaul, as a fallback mechanism. When the user's travel route changes, the user re-registers their new route with the *CO*.

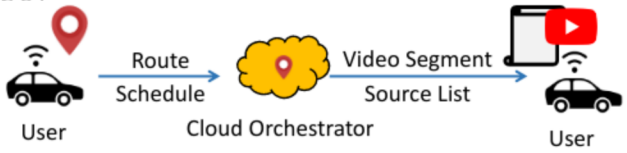


Figure 7: The cloud orchestrator analyzes the user's route information and produces a video segment source-list for the user. This list contains information about the edge nodes the user can obtain content from as it travels through the declared route.

3) Edge Node: Control Functions

The edge node monitors its physical layer characteristics (derived from experiments detailed in §V-A) and updates the *CO* (via the persistent connection it maintains with the *CO*) if its download range changes for any reason (obstructions, weather, etc.). It periodically reports these characteristics to the *CO*. The *CO* maintains a per-edge node PHY layer parameters table for appropriately generating segment assignments and source-lists.

According to instructions from the *CO*, the edge node prefetches video segments from a CDN, or a peer edge node, unless it already has the segment in its local cache. The edge node then tracks which segments were delivered to the user. It reports any shortfalls in delivery to *CO*. The *CO* then may decide to inform the next edge node in the source-list to fetch the un-delivered segments from this previous edge node (reconciliation mechanism).

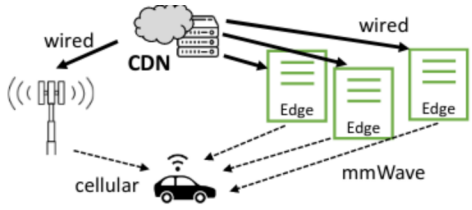


Figure 8: The *ClairvoyantEdge* data plane.

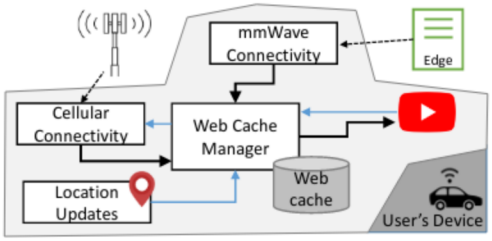


Figure 9: User's device maintains a web cache that serves the video player. The web cache receives data either through an edge node or through fallback cellular connectivity.

E. The Data Plane

The data plane comprises the movement of video segments between the various network entities (Fig. 8). The data plane actions of the origin servers or the CDNs from which *ClairvoyantEdge* components fetch video segments, are unchanged from their normal operations. We therefore delve deeper only into the operations at the user's device and the edge nodes.

1) User's Device: Data Functions

We expect that all video players will fetch "dashified" video segments by connecting with a video server online. To fetch video segments from edge nodes, the user device keeps track of the user's location and the GPS location of the edge nodes (source-list provided by *CO*). When in the vicinity of a designated edge node, the user device connects with the edge node using an mmWave link and sends the index of the first required video segment, as shown in Fig. 9. The edge node sends all available video segments starting from this index, while recording which segments were successfully delivered. When the user device moves too far, the connection is severed and the edge node relays the delivery information to *CO*.

2) Edge Node: Data Functions

The edge nodes are connected to the wide area Internet through high throughput fiber optic lines (called Internet backhaul). We also assume that physically proximal edge nodes are connected with each other over dedicated wired lines (e.g., Vapor [3]). Edge nodes maintain a local cache for storing prefetched data. When directed by *CO*, the edge nodes prefetch video segments into the cache from the origin servers over the Internet, or from peer edge nodes. *Cache policy is not the focus of this work* so we simply assume

that all edge nodes implement a least recently used (LRU) cache for storing segments.

Video segments are delivered to a user's device using mmWave links when a user in the edge node's vicinity requests them. When there is shortfall of data delivered, the edge node might be instructed by CO to transfer video segments to the next one in the source-list. Such transfer only uses direct connections between the edge nodes without occupying the Internet backhaul. A key utility of the edge node's caching and content sharing ability is that it saves the Internet backhaul bandwidth.

We now focus on strategies for managing and scheduling storage space on edge nodes, content sharing between edge nodes, and delayed prefetching of promised video segments.

Storage Management

When instructed by the CO to serve a specific user, the edge node reserves space in its storage to fetch the segments corresponding to the user's request. Unlike a regular LRU cache, which will evict the least recently accessed data, we cannot simply evict content that was fetched for a user who is yet to arrive, since the edge node has *promised* to hold that data for the user. Therefore, within an LRU cache, we make the distinction between *reserved* and *evictable* lists. The reserved list holds newly fetched data, yet to be accessed. When a video segment is inserted into the *reserved* list, the edge node tracks the accesses to the segment. The segment is maintained in the *reserved* list, until there are no pending requests for the segment. A single fetch can potentially serve multiple user requests for the segment, and the segment will stay in the *reserved* list the whole time. Once all pending requests for a segment have been served, the segment is moved to the *evictable* list, from which evictions are allowed. Additionally, it is possible for a segment to be moved from the *evictable* to *reserved* list, if a new request for the segment is received while the segment is in the *evictable* list. This way, we are able to guarantee that a segment will be available for a user, and also effect any eviction policy for the segments that are no longer required.

Content Sharing

Caches allow reuse of the segments by primarily two actors: a) A user who requests the same content in the future and b) A peer edge node that commits to storing segments on their local cache for a different request. The rationale behind reuse is that video popularity is considerably skewed [12], where some videos are exponentially more popular than most others. Local reuse and content sharing via horizontal communication between edge nodes confer the added benefit of reducing backhaul traffic to the CDNs. The various policies for establishing links between peer nodes for the horizontal communication with optimal cost vs. performance tradeoff is left as future work. We note that a new policy does not affect the control & data plane functions of *ClairvoyantEdge* which currently realizes all-to-all connections.

Deadline Based Delayed Prefetching

When the CO instructs an edge node to prefetch video segments for a user, the edge nodes does so in an eager fashion. Storage space on edge nodes is limited. Eager prefetching could result in edge nodes hoarding video segments, leading to suboptimal performance in terms of number of users served at the edge. For example, if user $u1$ makes a request to CO at time $t = 0$ and $u2$ makes a request at time $t = 10$. If $u1$ only arrives in the vicinity of an edge node at time $t = 100$ but $u2$ arrives at time $t = 20$, the edge node should intelligently download the segments corresponding to $u2$ first, even though $u1$ made the request to CO before $u2$, to efficiently use the available storage. We explore procrastination as a mechanism to use storage space efficiently. With procrastination, edge nodes fetch segments in a manner that is consistent with the user's arrival at the edge nodes. We delay segment downloads at the edge nodes until a certain threshold (e.g., if d is the estimated time it takes for the user to arrive, the threshold could be $d/2$, or a delay by 50%) of the user's arrival in the vicinity of the edge node. However, too much delay could result in not being able to complete the downloads in time, thereby affecting the edge data delivery (discussed in §V).

F. Design Summary

In summary, the design choices in *ClairvoyantEdge* are:

- The orchestrator is centralized in the cloud.
- The user requests a video and supplies their intended route information to the cloud orchestrator.
- The orchestrator prepares a source-list of video segments and sends it to the user, and participating edge nodes.
- The user's device connects to the edge node when in its vicinity to download the promised segments using mmWave links.
- The user's device and the edge nodes keep the orchestrator informed about successful (and unsuccessful) downloads, enabling course correction by the orchestrator.
- The cellular network is used as a fallback to fetch missed video segments as needed by the user's device.

IV. Implementation

Implemented in C++, Go, and Python (5000+ lines of code), *ClairvoyantEdge* comprises components that run on the user's device, cloud, and edge nodes. gRPC [1] is used for inter-process communication, and Redis [2] (in-memory key value store) is used for metadata management (written in C++). The CO is implemented as a gRPC service in Python. The edge and emulated client functionalities (both written in Go) are implemented as a gRPC server and client, respectively.

V. Performance Evaluation

We first evaluate the performance of mmWave links via field trials in §V-A. Next, in §V-B, we describe the experimental setup. Then, in §V-C, we present the evaluations of *Clair-*

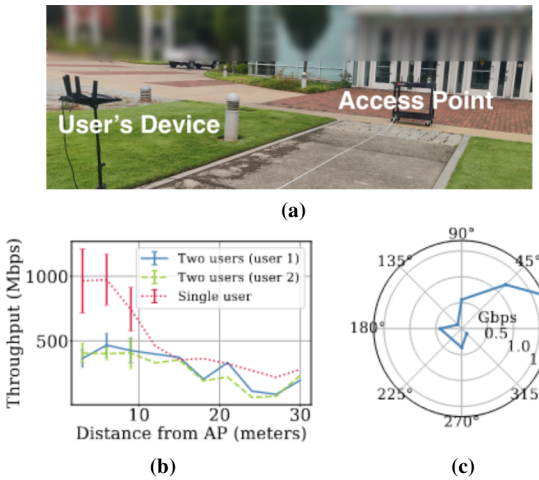


Figure 10: Throughput characteristics of mmWave links from field trials. (a) Photograph of the setup. (b) Throughput variation with distance. (c) Radial throughput characteristics for a single user.

voyantEdge on a synthetic dataset to quantify the impact of the intrinsic and extrinsic factors. Finally, we evaluate the performance of *ClairvoyantEdge* on a real-world dataset of San Francisco cabs [27].

A. Performance of mmWave links

We examine the throughput characteristics of the 60GHz mmWave links using Netgear X10 [23] routers, which operate on the 802.11ad [26] mmWave standard. Fig. 10a shows a photo of our outdoor field experiment where we assigned one of the two routers as the access point (AP) and the other as the user device. We measure the maximum throughput sustained by *iperf3* [13] between a stationary user and AP for a fixed distance. We then gradually increased the distance until the observed throughput fell below the expected WiFi throughput (see Fig. 3). Fig. 10b captures the throughput degradation with distance. While for the first $10\,m$, the mmWave link maintains a peak throughput of nearly 1000 Mbps, the next $10\,m$ saw a 50% drop in throughput. Beyond 30m, the throughput drops below normal WiFi speeds (< 200 Mbps) operating at the $2.4\,GHz$ and $5\,GHz$ frequencies. When a second client is placed in vicinity of the first, we observe that the two clients equally share the available bandwidth (as seen from the coinciding blue and orange lines in Fig. 10b).

In a companion study, we changed the angle between a single client and the AP to observe the angular throughput characteristics for our mmWave devices. We moved the client radially at a distance of $3\,m$ from the AP and observed a strong directional preference as seen in Fig. 10c, showing the feasibility of directional and spatial reuse. Given our current hardware it is possible to serve two users simultaneously by incorporating two mmWave devices on each edge node in different directions. Further spatial reuse is possible using MU-MIMO, or multiple simultaneous mmWave devices, as shown by other studies [14], [21], and an increased range is possible using outdoor and industrial grade routers [5]. In this work, we treat the set of mmWave devices on one edge node as a single unit, and specify the number of concurrent users supported in our evaluations. Similar experiments were performed in different environments where signals were influenced by various obstacles in path (including walls, pedestrians, vehicles etc.). We observe that the throughput by distance profile varies based on the environment, which points to the need for different mmWave link characteristic models per edge site. We have omitted graphing those throughput profiles due to space limitations.

B. Evaluation Setup

Compute. The underlying hardware setup consists of a datacenter grade server with an AMD EPYC 7501 32 core processor and 256 GB RAM. *ClairvoyantEdge* consists of 3 main actors - 1) CO, 2) Edge Nodes and 3) Users' devices. The CO (16 GB RAM/ 8 core CPU) and Edge nodes (4 GB RAM / 2 core CPU) run on dedicated VM's on the server for providing resource isolation. User devices are emulated via multithreading on a VM (16 GB RAM/ 8 core CPU).

Networking. The users' devices, CO and the edge nodes operate on a single LAN. We assume 1 Gbps links for CO-Edge, CDN-Edge and Inter-Edge connectivity. The user is assumed to be connected to the CO and CDN servers over LTE with 40 Mbps downlink.

mmWave Emulation Each edge node is equipped with an mmWave device over which the edge node delivers data to the user. The bandwidth delivery capability of the mmWave device is modeled based on the experiments described in (§V-A). As a baseline, we assume support for two concurrent users per edge node, unless otherwise specified.

Data plane Simulation. We simulate the data plane by keeping track of the amount of bandwidth being used by each link and estimate the time required to complete data transfers as opposed to actually performing the data transfers, to speed up evaluations.

Route Simulation. Each user in the dataset has a corresponding route, which is a timestamp-ordered list of coordinates. The client-emulator uses this list to simulate user movement, and also shares this list with the cloud orchestrator (CO) via video requests. The CO uses this list to negotiate with edge nodes for video-delivery. We use "user" and "route" interchangeably to mean the same thing.

C. Experiments on Synthetic Dataset

This section explores factors that affect the performance of *ClairvoyantEdge*. The main objective of *ClairvoyantEdge* is to ensure high data delivery from the edge (to reduce reliance on the cellular network). As an effect of utilizing horizontal communication links between edge nodes, we also hope to see a reduction in the use of backhaul bandwidth (edge-cloud network). We perform a systematic study of *ClairvoyantEdge*'s performance on a synthetic dataset (con-

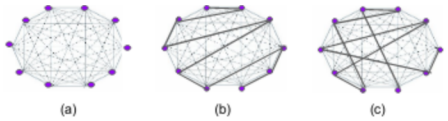


Figure 11: (a) Edge node positions on the vertices of a regular polygon (b, c) Different ordering of edge nodes selected as a user's simulated route.

sisting of travel routes) in the subsequent sections, which can be categorized as follows: (1) Impact of mmWave capacity, (2) System capacity under varying arrival rate, edge density, allowed concurrent mmWave transmissions, and user speed, (3) uncertainties caused due to mobility, and (4) Optimizations which explore the role of cache size and data delivery strategies (i.e., eager vs. lazy).

Edge Node Setup and Route Generation. We assume an n edge node setup, where edge nodes are placed at the n vertices of a *hypothetical* fully connected polygon of constant area ($68km^2$) in order to create a node configuration that can be re-used for ensuring consistency across experiments which analyze different parameters. Increasing n for the polygon (while keeping area constant) amounts to increasing the density of edge nodes in the topology. Each route traversed by a user is a list of timestamped-ordered coordinates which visit every node in the polygon in a random order, allowing us to generate comparable routes (see Fig. 11 for illustrations).

1) mmWave Capacity

We now examine the *implication* of a limit on the number of concurrent users using mmWave links, assuming the baseline scenario where only 2 users are allowed.

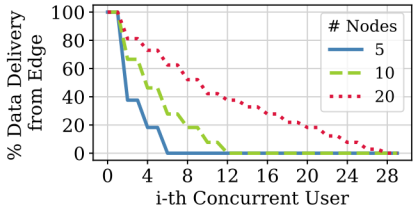


Figure 12: Edge delivery for concurrent users under different edge node densities.

Route Setup. Users traverse the same route, start at the same time and visit a given edge node simultaneously throughout the duration of travel. The experiment is repeated for three configuration of edge nodes: 5, 10, 20.

Fig. 12 shows the percentage of data delivered via *ClairvoyantEdge* for each user, for different edge node densities. Every user requests the same video of size 2 GB. While all users simultaneously arrive at a given edge node, the mmWave link at that edge node is arbitrarily allocated to the first 2 concurrent users who request for the data over the mmWave link. For the chosen workload (2 GB video), a user needs to visit only a small number of edge nodes to download the data they need; i.e., despite arriving simultaneously, different users get serviced by different edge nodes, creating an opportunity for more users to be served by *ClairvoyantEdge*. This opportunity improves with increasing density of edge nodes. For e.g., the 8^{th} user gets roughly 20% of data delivered via mmWave when the route has 10-nodes. However, if we increase the edge node density, and deploy 20 nodes instead, the 8^{th} user experiences a 30% increase in data delivery over the edge. The number of users plateaus at 28 since additional users get no data download from the edge for the densest (i.e., 20 edge nodes) configuration.

TAKEAWAY: Despite the 2-user constraint for each node, we observe a graceful degradation in edge data delivery across all nodes with increase in concurrent users, since just a few edge nodes suffice for delivering an entire video to a user.

2) System Capacity

We now examine system performance under a realistic user arrival pattern, with users traveling different along routes at different points in time.

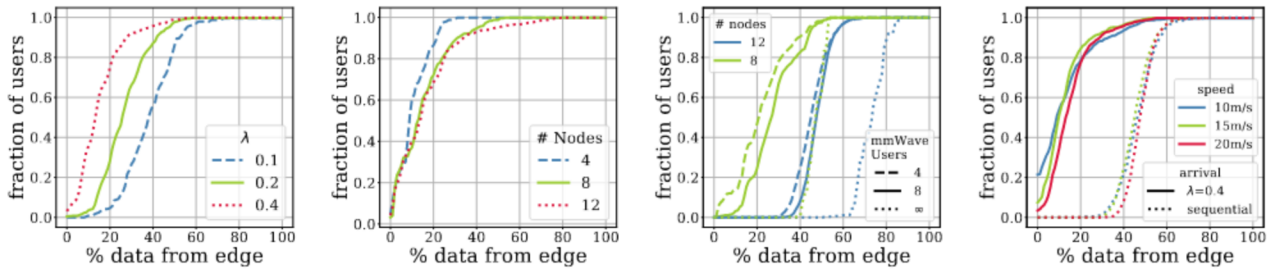
Route Setup. We generate routes in the same manner described in §V-B. We assume that users visit edge nodes in some random order. Each user may start at some point in time, independent of other users and therefore, we model user arrivals as a Poisson process expressed by Eqn. (1).

$$P(K) = e^{-\lambda}\frac{\lambda K}{K!} \tag{1}$$

Here, λ is the mean arrival rate of users and $P(K)$ is the probability of K users arriving in the system at that instant. The parameters which affect edge delivery include: varying λ, varying the edge density or alternatively the inter-node distance, concurrent users that a mmWave link can support, and speed of the user. All users request videos of the same size (2GB). Fig. 13 shows the CDF plots of % of data delivered to users by edge when we vary the aforementioned parameters. We now summarize our observations below.

Varying λ. User arrival rate is controlled by varying λ. The edge node setup for this experiment assumes a 10-node polygon. We have selected λ values of 0.4, 0.2 and 0.1 (λ = 0.4 corresponds to 833 users per hour, which is the observed traffic in a mid-sized city street). Each user travels at a constant speed of 15m/s. The average data delivery over the edge falls from 38.57% for λ = 0.1 to 12.72% for λ = 0.4. Higher arrival rates increase contention on the mmWave links resulting in a lower edge delivery, as illustrated by the CDF curves shifting left for higher values of λ in Fig. 13a.

Varying edge density. For a fixed value of λ = 0.4, we generate routes in a 4-, 8- and 12-node polygon setup. For a constant polygon area, increasing the node count decreases inter-node distance and increases node density. Fig. 13b shows that an increase from 4 nodes to 8 nodes yields a higher edge delivery (CDF shifts right). The shift indicates that lowering the inter-node distances causes more data to be fetched from edge nodes. However, a similar increase is absent when nodes are increased from 8 to 12 nodes. Like in §V-A, each mmWave link only supports a maximum



(a) mean arrival rate of users under Poisson arrivals. **(b)** inter-node distance. **(c)** maximum concurrent users supported by mmWave. **(d)** user speed

Figure 13: CDF plots of fraction of users vs. % data delivery from edge for the users for different parameters.

of 2 simultaneous users, and in the next experiment we demonstrate that this limit is the primary bottleneck in edge data delivery for node-densities beyond 8 nodes.

Varying support for concurrent users. To quantify the impact of the mmWave limitation on edge data delivery, we vary the number of concurrent users that an mmWave node can support. We assume that each user gets the same high throughput, that a single user would. Zooming in on only the 8-node and 12-node configurations, we increase the number of concurrent users that an edge node can support (assuming sectorized mmWave antennnas) as follows: 2, 4, 8, and unlimited (hypothetical). Fig. 13c shows that for the same node density, increasing mmWave capacity improves edge node delivery by a significant amount. For example, the 8 node configuration sees a 32% increase in average edge delivery when the concurrent user limitation is dropped. Hence, we confirm our original hypothesis outlined before that the 12 node configuration is handicapped by the mmWave capacity. The 12 node configuration results in a 13% higher average delivery over the 8 node setup when the concurrent users in mmWave was increased to 8 from 2.

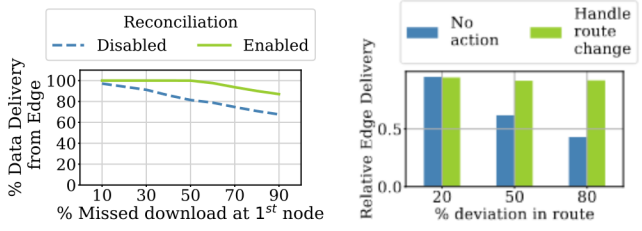
Varying user speed. We vary the user speed from $10\,m/s$ through $20\,m/s$ in increments of $5\,m/s$, for a 10-node setup and $\lambda = 0.4$. Increasing speed has two effects. First, it reduces the time taken by the user to traverse the inter-node distance (which should have the same intended effect as increasing the density of nodes on the route). Second, it reduces the contact time of the user with the edge node resulting in less data downloaded from an edge node. Fig. 13d captures these counteracting effects resulting in only minor variations in the edge data delivery with varying speeds. This observation holds true regardless of the arrival pattern. Even when users arrive sequentially (each user arrives after the previous ones have completed their routes) with no contention on the mmWave links, the edge delivery over various speeds remains similar: A slow moving user is likely to run out of buffered segments and download data from the cellular network, while a fast moving user would experience a shorter contact time with the edge node, reducing the amount of data downloaded from the edge.

TAKEAWAY The positioning of edge nodes (inter-node distance), arrival rate of users, and ability of mmWave devices to support concurrent users play an important role in determining the benefit derived from the system.

3) Handling Uncertainties

In this section we analyze how *ClairvoyantEdge* manages vagaries in user mobility, such as changes to expected contact time with edge nodes, and change in travel routes.

Accommodating Non-Exact Schedules. *ClairvoyantEdge* creates a schedule based on the initial route information provided by the user. However, uncertainties in real world could result in the user having reduced contact time with an edge node than anticipated in the schedule. Consequently, less content is delivered to the user from edge nodes. To handle such uncertainties, *ClairvoyantEdge* identifies the segments that were not delivered to the user and forwards them to downstream nodes to increase the chances of edge delivery, referred to as the *reconciliation mechanism*. A mismatch of schedules at request time and run time is equally likely to happen at every edge node along the route. However, a shortfall at even the first edge node suffices to observe the detrimental effects of such a mismatch. To demonstrate this phenomenon, we reused the route generated for §V-C1. We then varied the amount of missed deliveries that would occur as a consequence of mismatched schedules to observe *ClairvoyantEdge*'s behavior with and without the reconciliation mechanism. Fig. 14a shows that when



(a) **(b)**

Figure 14: *ClairvoyantEdge* handles uncertainties in routes. (a) Correcting for mismatched schedules to improve overall delivery. (b) Relative improvement in Edge data delivery on user-initated route change.

less than 50% data was missed at the first edge node, *ClairvoyantEdge*'s reconciliation mechanism was still able to

guarantee 100% edge delivery. Beyond 50% data miss at the first edge node, there is some loss in overall edge delivery. The loss can be attributed to the fact that the segments accumulated by the user at the first edge node is not sufficient to match the playback rate, and the user instead falls back to cellular data to fetch the remaining segments between the first and second node.

TAKEAWAY Reconciliation is helpful when the user's contact time with an edge node is shorter than expected, and the user has enough content buffered to play out until reaching the next edge node.

Accommodating Route Changes. In real life, owing to congestion and user unpredictability, it is possible that the routes of users change midway through the journey. We investigate the impact of such changes on data delivery over mmWave by simulating route changes in the synthetic dataset. We construct alternative routes for users and vary the amount by which the users deviate from their original route. We have used a 10-node polygon setup, with users traveling at $15\,m/s$. The users follow Poisson arrivals with $\lambda = 0.4$, and change the routes for 20% of the users. Two cases are considered: 1) No action is taken by the system; 2) CO is contacted to re-trigger segment assignments to edge nodes.

Fig. 14b shows the edge data delivery experienced by the deviating routes, as a fraction of the edge data delivery that would have been possible had the routes not changed after the journey began. We found that not handling route changes causes a significant drop in edge data delivery (38-57%) when the users deviate beyond 50% of the originally planned route, whereas, re-issuing the request for the new route, only sees 8.5% drop in edge data delivery on an average.

TAKEAWAY Route changes are handled by re-triggering segment assignments to edge nodes, and does not require any other special handling.

4) Data Plane Optimizations

In this section, we investigate the impact of content sharing with peer edge nodes on cache size, and the implications of delayed prefetching on edge data delivery to the user.

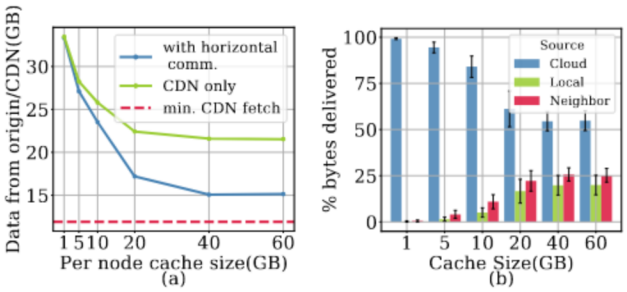


Figure 15: (a) Effect of prefetching from peers on backhaul bandwidth reduction (b) Breakdown of download sources for segments prefetched at an edge node.

Cache size. Edge nodes require a persistent storage to a) cache video data to be delivered to the user, and b) to facilitate horizontal communication between edge nodes which reduces backhaul bandwidth. However, storage requirements for caching cannot be unbounded. Here, we investigate the extent to which the storage sizes influences backhaul traffic. For generating user travel routes, we assume a Poisson user arrival with $\lambda = 0.4$ for a 10-node polygon. Each edge node is equipped with an LRU cache. Each user requests a video which is picked at random from a Zipfian-distributed pool of 2000 videos. Fig. 15a shows a backhaul bandwidth reduction of 45% when the cache size increases from 1 GB to 40 GB. Further increase in cache size does not yield a reduction in backhaul bandwidth, indicating that for the given configuration, a 40 GB cache suffices. Every new video that is prefetched must be retrieved from the CDN at least once, creating a lower-bound on backhaul traffic (red dashed line at the bottom). Additionally, Fig. 15a also illustrates the bandwidth reduction with and without horizontal communication. Unsurprisingly, horizontal communication between peer nodes effectively reduces backhaul bandwidth needs. However, the gap between mandatory prefetches from the CDN and peer prefetching remains as large as 6.5 GB even at larger cache sizes, which is likely an artifact of concurrent user requests resulting in multiple fetches for the same video over the backhaul.

Finally, Fig. 15b, shows the source of video segments for the user requests when horizontal communication is enabled. Increasing the cache sizes allows edge nodes to hold a larger number of segments for delivery to committed users as well as mutually benefit from each other's cache of video data.

TAKEAWAY Opportunities for improving backhaul savings are constrained by cache size, and availability of segments at peer edge caches.

Prefetching Strategy. The core idea in *ClairvoyantEdge* is that edge nodes prefetch video segments in response to user requests. However, there could be a significant gap between the time a user requests a video and the time the user actually arrives at a specific edge node. Eager prefetching of segments could result in hoarding of the segments by edge nodes which prevents them from participating in serving new routes due to limited cache size. Instead, a lazy prefetching strategy which is driven by the deadline of user arrival leads to better use of the cache. If the time between user request and arrival is t, then the amount of delay (in %) corresponds to the fraction of t which an edge node must wait before prefetching from the CDN. We observed that the total cache occupancy reduces by 15-44% when the segments are prefetched in accordance with the user's arrival time, as shown in Fig. 16a. Additionally, delayed prefetching does not induce drastic changes to the backhaul bandwidth consumption as confirmed by the coinciding lines for various delay rates in Fig. 16b, since all nodes follow the same strategy.

Fig. 17a shows the data delivery from the edge to users for cache sizes: 5, 10 and 20 GB, under different delays. Across

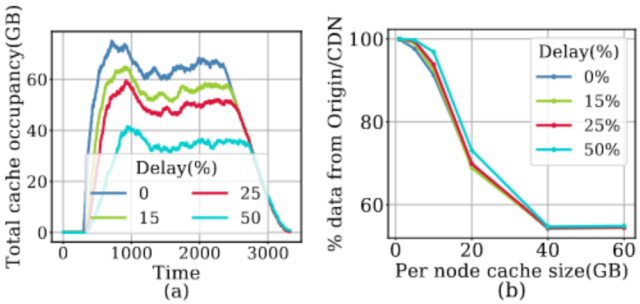


Figure 16: (a) Effect of prefetching from peers on in eager vs lazy fashion (b) Backhaul savings for eager vs deadline based prefetching.

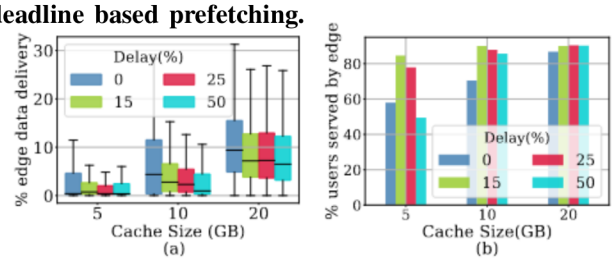


Figure 17: (a) Data delivery to users with 10 edge nodes to users under per node cache sizes (b) Users who got data from edge under various cache and delay parameters.

the cache sizes, the different delays do not affect the edge data delivery. However the delay percentages dictates the number of users served by *ClairvoyantEdge*. In Fig. 17b, a delay of 15% sees a marked increase in the number of users served in comparison with eager prefetching (zero delay). For an increase in delay beyond 15%, the corresponding median edge delivery percentage fell, indicating insufficient time to prefetch the segments.

TAKEAWAY Deadline based prefetching makes more efficient use of cache space than eager prefetching and allows more users to download data through *ClairvoyantEdge*.

D. Evaluating on real world data

We now study the performance of *ClairvoyantEdge* on a realistic trace using the San Francisco Cabs dataset [27]. We evaluate *ClairvoyantEdge* on (1) offloading last mile delivery to edge nodes, and (2) reduction in backhaul traffic.

Route Setup. We simulated travel routes for users from the San Francisco Cabs dataset [27], which contains GPS data of cab routes from 500 cabs, over a 24-day period. Each taxi has its own GPS route file containing a time-ordered list of GPS coordinates. A tuple in this dataset is described as `<userid, latitude, longitude, time, velocity>`. We partition the GPS route file into smaller routes, each representing a unique cab ride. A single taxi GPS file corresponds to over 1000 routes spanning across 24 days. We process 500 such GPS files to create our evaluation dataset. Fig. 18a summarizes the variation in hourly ridership over the span of 24 hours. The number of routes within the hour vary between 300 and 800 (corresponding to approximately $\lambda = 0.1$ and $\lambda = 0.2$). The arrival distribution from 10 AM to 11 AM (Fig. 18b) closely

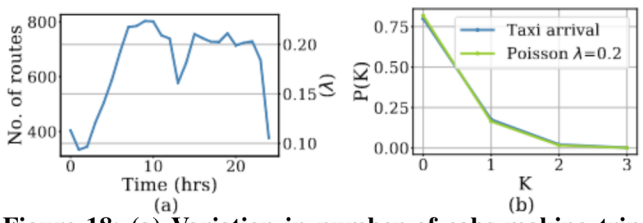


Figure 18: (a) Variation in number of cabs making trips (b) Probability distribution for 10 AM (red line) and ideal Poisson process with $\lambda = 0.2$ (resemble each other closely).

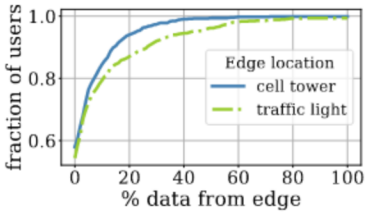


Figure 19: Sensitivity of *ClairvoyantEdge* to choice of edge node placement strategies.

resembles Poisson arrivals with $\lambda = 0.2$ which reinforces the validity of our assumptions for the arrival patterns in §V-C.

Edge Node Placement. We evaluate *ClairvoyantEdge*'s performance with 20 edge nodes in the system which roughly translates to one edge node for every $6.5km^2$ if the edge nodes are uniformly distributed. We examine two approaches for the placement of 20 nodes: 1) K-means (K=20 clusters) clustering on cell tower locations in San Francisco to find cluster centroids, each representing an edge node 2) the traffic light locations which observe heavy vehicular flow.

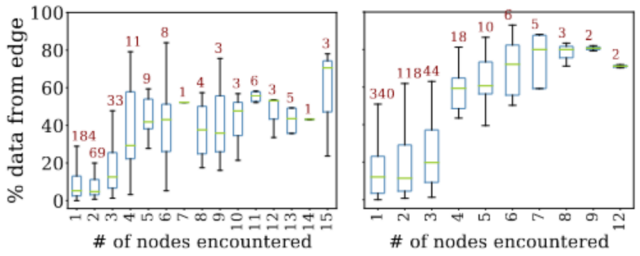
1) Offloading Last mile delivery to Edge Nodes

We first evaluate the choice of using cell towers vs. traffic lights for positioning edge nodes. We filter the routes to ensure that a user passes through at least one edge node to avail the benefits from *ClairvoyantEdge*. As shown in Fig. 19, traffic lights perform marginally better than cell towers. However, the average edge data delivery remains poor for both configurations, as reported in Table I. Fig. 20 reveals the root cause for the poor *average* performance reported in Table I. The amount of edge data delivery is directly proportional to the number of edge nodes a route visits. Even for users passing through just 5 edge nodes, median deliveries are as high as 42% for 2 concurrent users, which further grows to 60% upon increasing the concurrent users to 8. This underscores the importance of edge node placement over the absolute count of edge nodes on a route.

For the next experiment, we assume that a user passes through at least two edge nodes—a reasonable assumption given that the edge nodes are collocated with traffic lights. We now study

Configuration	Mean Edge Delivery(%)
Cell tower	4.49
Traffic light	7.07

Table I: Effect of edge node location on data delivery



(a) 2 concurrent users **(b)** 8 concurrent users

Figure 20: % of data downloaded from edge nodes. Routes are grouped by the number of edge nodes they encounter. The number of routes in each group is indicated above the corresponding boxplot.

the impact of arrival rates, and number of concurrent users supported by mmWave on the edge data delivery. Per Table II, the edge data delivery is inversely related to the arrival rate of users. With mmWave supporting only 2 simultaneous users, $\lambda = 0.1$ provides nearly 2× the improvement in mean edge deliveries over the case when $\lambda = 0.2$. We further find that for a fixed $\lambda = 0.1$, the mean edge delivery of 19% for 2 simultaneous users increases to 27% for 8 simultaneous users, thereby capturing mmWave capacity as the primary bottleneck for the system with varying arrival rates.

TAKEAWAY A city-scale deployment of *ClairvoyantEdge* requires proper placement of edge nodes in addition to scaling mmWave link capacity with demand to ensure high edge deliveries and reasonably offset the cellular burden.

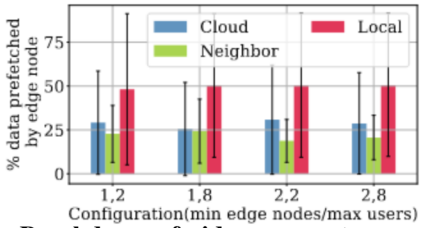


Figure 21: Breakdown of video segment sources for data fetched to edge nodes.

λ	no. of simultaneous users	mean edge delivery (%)
0.1	2	19.23
0.1	8	27.13
0.2	2	9.87
0.2	8	21.22

Table II: Mean data delivery percentage over mmWave

2) Reduction in Backhaul Traffic

We now investigate the ability of *ClairvoyantEdge* to reduce traffic on backhaul links for real world data. Fig. 21 plots the savings in backhaul traffic (i.e., reduction in requests to CDN/origin servers) when users pass through at least 1 or 2 edge nodes (out of possible 20), for arrival rate of $\lambda = 0.2$, and for two mmWave configurations (max number of users = 2 or 8). Users request videos from a pool of 2000 videos in Zipfian fashion, and each edge node is equipped with 100GB local storage for caching segments. We observe a reduction in backhaul traffic by 50% on average, for all data delivered through the edge infrastructure (either locally or from peers). This saving comes from content reuse at edge nodes.

VI. Discussion

We discuss some of the directions for future research that would enhance the utility of *ClairvoyantEdge*.

Challenges in using mmWave Technology. *ClairvoyantEdge* abstracts the inner workings of mmWave links as a system capable of delivering data over a high throughput channel. It does not explicitly account for the time taken to establish connection between the AP and client. Recent work has explored ways to accurately model mmWave beamforming [31]. Further, *ClairvoyantEdge* focuses on single antenna systems, but in the future, a single mmWave access point could be equipped with multiple antennas in order to handle several simultaneous multi-gigabit channels.

Monetary Cost. mmWave has huge potential in terms of its throughput capabilities, but its limited range means that only a handful of users can be served by a single AP at a time which in turn a entails dense deployment of APs, the monetary cost of which we have not factored.

Centralized vs Decentralized Control Plane. *ClairvoyantEdge* uses a centralized control plane. An alternative design choice would be to explore a fully decentralized control plane, giving more autonomy to edge nodes in making scheduling decisions. Such a design choice might reduce the metadata maintenance overhead at the *CO* and provide a finer control based on instantaneous mmWave link performance.

Edge Node Positioning. We explored two possibilities for placement of edge nodes: on traffic lights, and close to cell tower locations. However, the utility of an edge node at a specific location depends on the traffic flow. In some congested areas, multiple edge nodes may be necessary in order to serve the incoming requests. Positioning of edge nodes is an open problem worthy of further investigation.

User Behavior and Prefetching. There is recent work on predicting users' video-watching behavior to inform caching strategies [17]. Inclusion of such works into *ClairvoyantEdge*'s prefetching strategies is another fruitful exploration.

VII. Conclusion

In this work, we explored offloading content delivery for on-demand video to a network of mmWave enabled edge nodes through *prescient prefetching*. Our evaluations show a reduction of cellular traffic by about 20% on average, in a geographical area of 46 km^2, even though the edge nodes' mmWave range covers only about 0.12% of the total area.

Acknowledgments

We thank our anonymous reviewers, and our lab members for their insightful feedback and suggestions. This work was funded in part by NSF I/UCRC FiWIN Center (IIP-1821819), NSF CNS-1909346, and a gift from Microsoft Corp.

References

[1] grpc, 2020. https://grpc.io/.

[2] Redis, 2020. https://redis.io/.

[3] Vapor, 2020. https://vapor.io/.

[4] Where can i get 5g edge, 2021. https://www.verizon.com/business/solutions/5g/edge-computing/.

[5] Ubiquiti, 2022. https://store.ui.com/.

[6] Suzan Bayhan, Setareh Maghsudi, and Anatolij Zubow. Edgedash: Exploiting network-assisted adaptive video streaming for edge caching. *IEEE Transactions on Network and Service Management*, 18(2):1732–1745, 2021.

[7] Carlos Colman-Meixner, Pedro Diogo, Muhammad Shuaib Siddiqui, Antonino Albanese, Hamzeh Khalili, Alexandros Mavromatis, Luca Luca, Alexandre Ulisses, Jordi Colom, Reza Nejabati, et al. 5g city: A novel 5g-enabled architecture for ultra-high definition and immersive media on city infrastructure. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–5. IEEE, 2018.

[8] Carlos Colman-Meixner, Hamzeh Khalili, Konstantinos Antoniou, Muhammad Shuaib Siddiqui, Apostolos Papageorgiou, Antonino Albanese, Paolo Cruschelli, Gino Carrozzo, Luca Vignaroli, Alexandre Ulisses, et al. Deploying a novel 5g-enabled architecture on city infrastructure for ultra-high definition and immersive media production and broadcasting. *IEEE Transactions on Broadcasting*, 65(2):392–403, 2019.

[9] Savio Dimatteo, Pan Hui, Bo Han, and Victor OK Li. Cellular traffic offloading through wifi networks. In *2011 IEEE eighth international conference on mobile ad-hoc and sensor systems*, pages 192–201. IEEE, 2011.

[10] Ericsson. *Ericsson Mobility Report*, June 2021. https://www.ericsson.com/4a03c2/assets/local/mobility-report/documents/2021/june-2021-ericsson-mobility-report.pdf.

[11] Chang Ge, Ning Wang, Severin Skillman, Gerry Foster, and Yue Cao. QoE-Driven DASH Video Caching and Adaptation at 5G Mobile Edge. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking*, ACM-ICN '16, page 237–242, New York, NY, USA, 2016. Association for Computing Machinery.

[12] Phillipa Gill, Martin Arlitt, Zongpeng Li, and Anirban Mahanti. Youtube Traffic Characterization: A View from the Edge. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 15–28, 2007.

[13] iPerf. *iPerf3*, August 2021. https://iperf.fr/iperf-download.php.

[14] Suraj Jog, Jiaming Wang, Junfeng Guan, Thomas Moon, Haitham Hassanieh, and Romit Roy Choudhury. Many-to-Many beam alignment in millimeter wave networks. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 783–800, Boston, MA, February 2019. USENIX Association.

[15] Nusratullah Khan, Muhammad Usman Akram, Asadullah Shah, and Shoab Ahmad Khan. Important attributes of customer satisfaction in telecom industry: A survey based study. In *2017 4th IEEE International Conference on Engineering Technologies and Applied Sciences (ICETAS)*, pages 1–7. IEEE, 2017.

[16] Shashwat Kumar, Doddala Sai Vineeth, et al. Edge assisted dash video caching mechanism for multi-access edge computing. In *2018 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE, 2018.

[17] Shruti Lall, Uma Parthavi Moravapalle, and Raghupathy Sivakumar. Mantis: Time-shifted prefetching of youtube videos to reduce peak-time cellular data usage. In *Proceedings of the 11th ACM Multimedia Systems Conference*, MMSys '20, page 112–125, New York, NY, USA, 2020. Association for Computing Machinery.

[18] Chenglin Li, Laura Toni, Junni Zou, Hongkai Xiong, and Pascal Frossard. Qoe-driven mobile edge caching placement for adaptive video streaming. *IEEE Transactions on Multimedia*, 20(4):965–984, 2018.

[19] Luyang Liu, Hongyu Li, and Marco Gruteser. Edge assisted real-time object detection for mobile augmented reality. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[20] Ge Ma, Zhi Wang, Miao Zhang, Jiahui Ye, Minghua Chen, and Wenwu Zhu. Understanding performance of edge content caching for mobile video streaming. *IEEE Journal on Selected Areas in Communications*, 35(5):1076–1089, 2017.

[21] Sohrab Madani, Suraj Jog, Jesus O. Lacruz, Joerg Widmer, and Haitham Hassanieh. Practical null steering in millimeter wave networks. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 903–921. USENIX Association, April 2021.

[22] Maluambanzila Minerve Mampaka, Wemambolo Carel Tokombe, and Mbuyu Sumbwanyambe. Brand-aware dropped-call rate optimization for voice over lte. In *2020 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems (icABCD)*, pages 1–6. IEEE, 2020.

[23] Netgear. *Netgear Nighthawk X10*, August 2021. https://www.netgear.com/home/wifi/routers/ad7200-fastest-router/.

[24] Riley Panko. The popularity of google maps: Trends in navigation apps in 2018, 2018. https://themanifest.com/app-development/trends-navigation-apps.

[25] Gi Seok Park and Hwangjun Song. Cooperative base station caching and x2 link traffic offloading system for video streaming over sdn-enabled 5g networks. *IEEE Transactions on Mobile Computing*, 18(9):2005–2019, 2018.

[26] Eldad Perahia, Carlos Cordeiro, Minyoung Park, and L Lily Yang. Ieee 802.11 ad: Defining the next generation multi-gbps wi-fi. In *2010 7th IEEE consumer communications and networking conference*, pages 1–5. IEEE, 2010.

[27] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. CRAWDAD dataset epfl/mobility (v. 2009-02-24). Downloaded from https://crawdad.org/epfl/mobility/20090224/cab, February 2009. traceset: cab.

[28] Ioannis Psaras, Onur Ascigil, Sergi Rene, George Pavlou, Alex Afanasyev, and Lixia Zhang. Mobile data repositories at the edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[29] Waqas Ur Rahman, Choong Seon Hong, and Eui-Nam Huh. Edge computing assisted joint quality adaptation for mobile video streaming. *IEEE Access*, 7:129082–129094, 2019.

[30] Dinesh Ramasamy, Radhakrishna Ganti, and Upamanyu Madhow. On the capacity of picocellular networks. In *2013 IEEE International Symposium on Information Theory*, pages 241–245, 2013.

[31] Hem Regmi and Sanjib Sur. Argus: Predictable millimeter-wave picocells with vision and learning augmentation. *Proc. ACM Meas. Anal. Comput. Syst.*, 6(1), feb 2022.

[32] Rony Kumer Saha. A technique for massive spectrum sharing with ultra-dense in-building small cells in 5g era. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–7, 2019.

[33] David Santos, Rui Silva, Daniel Corujo, Rui L Aguiar, and Bruno Parreira. Follow the user: A framework for dynamically placing content using 5g-enablers. *IEEE Access*, 9:14688–14709, 2021.

[34] Enrique Saurez, Harshit Gupta, Alexandros Daglis, and Umakishore Ramachandran. Oneedge: An efficient control plane for geo-distributed infrastructures. In *Proceedings of the ACM Symposium on Cloud Computing*, SoCC '21, page 182–196, New York, NY, USA, 2021. Association for Computing Machinery.

[35] Iraj Sodagar. The mpeg-dash standard for multimedia streaming over the internet. *IEEE multimedia*, 18(4):62–67, 2011.

[36] Joel Sommers and Paul Barford. Cell vs. wifi: On the performance of metro area mobile connections. IMC '12, page 301–314, New York, NY, USA, 2012. Association for Computing Machinery.

[37] Matthias Steeg, Nathan J Gomes, Adrian A Juarez, Michal Kosciesza, Mason Lange, Yigal Leiba, Hiroshi Mano, Hiroshi Murata, Michal Szczesny, and Andreas Stohr. Public field trial of a multi-rat (60 ghz 5g/lte/wifi) mobile network. *IEEE Wireless Communications*, 25(5):38–46, 2018.

[38] Liyang Sun, Fanyi Duanmu, Yong Liu, Yao Wang, Yinghua Ye, Hang Shi, and David Dai. *Multi-Path Multi-Tier 360-Degree Video Streaming in 5G Networks*, page 162–173. Association for Computing Machinery, New York, NY, USA, 2018.

[39] Jihoon Sung, Minseok Kim, Kyongchun Lim, and June-Koo Kevin Rhee. Efficient cache placement strategy in two-tier wireless content delivery network. *IEEE Transactions on Multimedia*, 18(6):1163–1174, 2016.

[40] Mehmet Fatih Tuysuz and Mehmet Emin Aydin. Qoe-based mobility-aware collaborative video streaming on the edge of 5g. *IEEE Transactions on Industrial Informatics*, 16(11):7115–7125, 2020.

[41] Antonio Virdis, Giovanni Nardini, Giovanni Stea, and Dario Sabella. End-to-end performance evaluation of mec deployments in 5g scenarios. *Journal of Sensor and Actuator Networks*, 9(4):57, 2020.

[42] Wenxiao Zhang, Feng Qian, Bo Han, and Pan Hui. Deepvista: 16k panoramic cinema on your mobile device. In *Proceedings of the Web Conference 2021*, WWW '21, page 2232–2244, New York, NY, USA, 2021. Association for Computing Machinery.

[43] Yibo Zhu, Zengbin Zhang, Zhinus Marzi, Chris Nelson, Upamanyu Madhow, Ben Y. Zhao, and Haitao Zheng. Demystifying 60ghz outdoor picocells. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, MobiCom '14, page 5–16, New York, NY, USA, 2014. Association for Computing Machinery.