Deep Science Publishing
https://doi.org/10.70593/978-93-49307-75-9_4

DeepScience
Open Access Books

Chapter 4: Enhancing real-time communication and minimizing latency by deploying edge computing across network nodes

4.1 Introduction

Real-time communication and collaboration applications have transformed the way we interact and can be seen as foundational technologies in knowledge sharing and remote work. The applications are built upon real-time media processing and networked systems, and hence are typically deployed using cloud infrastructures. With the ubiquity of deployment, these approaches may return poor quality to sensitive end hosts due to high volume of requests and usage spikes, high network latency, and packet loss. As such, demand has always existed for more effective communication and collaboration systems across geographically dispersed members.

The rise of edge computing has stimulated research interest in enabling low-latency flows. In the context of advanced networks, integrated scope can greatly enhance the efficiency of real-time communication and collaboration. The integrated scope can be supported by different network segments, such as edge cloud and optimized network slicing. Combining network intelligence with flexible mobile edge computing, we enable end users to fully enjoy ultra-high-quality real-time media services. Ensuring end host service deployment involves various and difficult challenges. With system management methods, engineering "slim" end-user devices can be activated and fully integrated, offering long-lasting and high-quality connectivity management. Moreover, the new end-user system can also improve both the economy and performance.

4.1.1. Overview of Key Concepts and Scope

Section 1. Introduction Classical cloud computing offers a powerful infrastructure that enables real-time communications, which cannot meet the requirements of the next generation of networks. There are several serious issues associated with CDNs. In global routing, the AS path length is still longer than necessary and thus can offer reduced global latency and inferior performance on the network. Further, links of CDN data centers to end user groups may also be congested. As a result, particularly for security purposes, the requests for data and the replies to requests are typically forwarded to the same CDN data center, which can preclude anycast use and design flexibility problems. A subsequent source of latency is the round trip to the CDN data center, which could also be much longer than desirable. CDNs can also be susceptible to DDoS-style attackers. With all of these limitations through the advanced multimedia applications, the next generation of networks, smart cities, and the future Internet of Things will require real-time guarantees to support a new range of machine-based real-time consumers. This type of content delivery network is also needed to meet the quality demands of live-streaming video applications and for use in edge networks. This chapter presents key technologies for accomplishing an edge network that is built around both established and cloud-edge computing system acceleration. These technologies include object distribution, wireless and virtualization techniques, 5G wireless support, and cloud services. They then describe potential edge network applications and provide a comparison of the internet by using selected client-server architectures. It then looks to the future of edge network technology.

4.2. Understanding Real-Time Communication

Real-time communication refers to instantaneous communication that occurs over a given medium (Chiang & Zhang, 2016; Mach & Becvar, 2017; Abbas et al., 2018). Real-time voice and video communication, in particular, greatly enhance the productivity and effectiveness of a team whose members may be in different locations. Indisputably, video conferencing is a clear demonstration of real-time communication in action. It enables team members who are not physically located together to see and speak with one another simultaneously. One obvious advantage of real-time communication is an almost immediate response time, leading to a faster decision-making process. This is particularly important when timely communication of information among team members is critical in the decision-making process. In certain work or life-threatening situations, an inappropriate decision can be lethal. In many cases, being in the moment means being a matter of seconds ahead. Hence, real-time communication is essential.

Real-time communication can only be effective when there is a very low delay in the transmission. Data transmissions over the Internet occur over a long and fragile chain, with each component contributing a small amount of delay. Therefore, the round-trip delay incurred in data transmission is long. Consequently, real-time communication, especially real-time video or real-time voice communication, over the Internet is difficult. The long delay means that the conversation can be unnatural and distracting, as the rapid exchange of both verbal and nonverbal cues is challenging. Additionally, it is not pleasant conversing with another person who is speaking in a slightly delayed manner. These problems can be resolved by reducing the delay in the transmission of real-time communication. There are certain industries where real-time communication is particularly useful. Examples include law enforcement, news agencies, and the financial sector.

4.2.1. Definition and Importance

The Edge Computing concept has its origins in the decentralization of the traditional client-server relationship model from cloud computing, promoted by the increasing

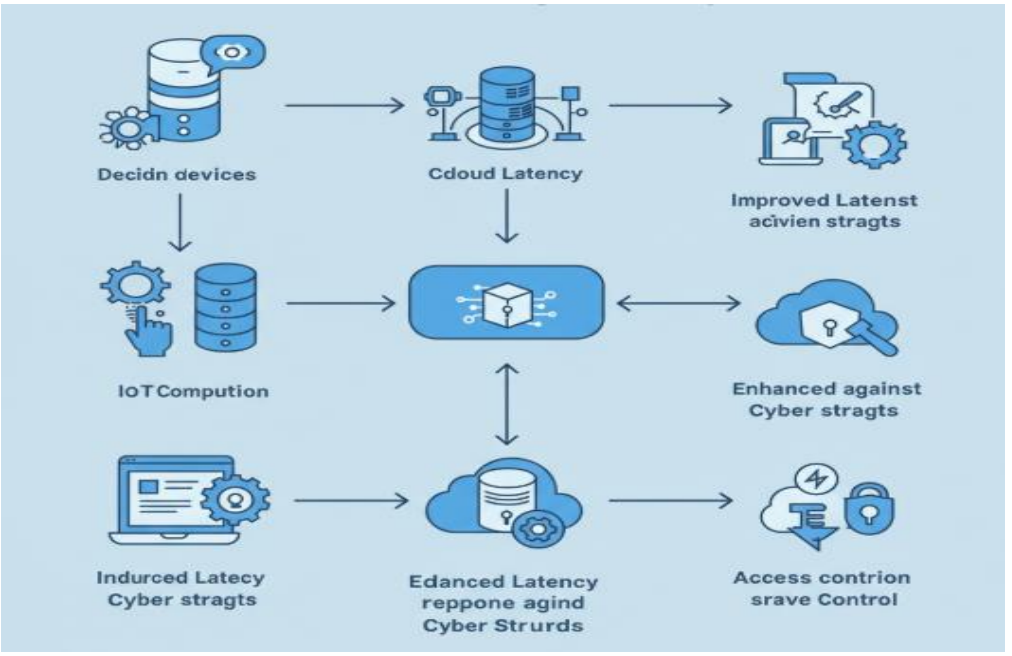

Fig 4 . 1 : Combining Edge Computing-Assisted Internet of Things Security

commercialization of mobile devices provided with computing power. It aims to provide computing and storage resources close to where they are consumed, that is, at the

extreme edge of the network, at the so-called Fog Computing, or where data from these devices are collected, transmitted, and processed. This decentralization trend also aims to alleviate the center by reducing the volume of data circulating in the network. It facilitates the development of applications for the Internet of Things and other real-world applications, such as robotics, remote health care, and vehicular networks. Among the Edge Computing requirements, we can highlight the requirement to be geographically distributed according to the network topology of the served geographical area, thus ensuring that existing time constraints will be preserved, interoperability, resource sharing, energy efficiency, guaranteeing Quality of Service, contention in terms of access to communication resources, among others.

One of the most important aspects of the concept of Edge Computing is that cloud computing, because it prioritizes wide area communication, as well as the geographical concentration of physical resources needed to store and process data, cannot meet the requirements of bandwidth, low latency, and QoS required by devices embedded in the connected world, especially for terrestrial vehicles, ships, and aircraft. It is pointed out that the processing in this scheme is done on the first intermediate switches, a few hops from the source sensor, which generates small content parts or aggregates them into larger parts and transmits them to the cloud to be stored or processed. The problem that is sought to be addressed is to identify the communication architecture and the hardware requirements necessary to disperse the processing of micro and macro data, which can offer the required parameters without the need to store a large amount of data on smart devices. This would allow them to be built with simple storage resources and software, and capable of long periods of use without the need for excessive recharging.

4.2.2. Challenges in Real-Time Communication

In the current and future era of the real-time communication system, there are many significant challenges to face. The main challenges include faster design and deployment, intelligent capability enhancement, and hardware acceleration (Mouradian et al., 2017; Taleb et al., 2017). The first significant problem is how to simplify the design of real-time communication. It is difficult for system manufacturers to face the various application requirements and the unknowns in the communication system at the same time. It is essential to provide a simple method that does not require the manufacturer to use any driver software to achieve real-time communication. In addition, the real-time communication system is also very complex. After the manufacturer's acceleration algorithm analyzes and manipulates different message types and commands for different applications and purposes, these manipulation objects will

be different, which makes the system rather complicated. Therefore, the research work of manufacturers in the real-time communication process is also a great challenge.

At present, regular software cannot meet the needs of real-time communication protocol stack processing. Under the general availability of deep learning training applications, manufacturers find that deep learning accelerators have extensive use as hardware acceleration in terms of enhanced computing capabilities in communication systems. Therefore, in recent years, many intelligent hardware acceleration algorithms have been proposed using deep learning technology. High flexibility with lower cost is the key requirement for real-time communication technology development and application. The consumer VPN gateway is a low-cost prototype that has been validated through real-world testing. The key technical problem is that the bitstream and route produced by the current tools cannot meet the delay requirements of the real-time communication technology; the key solution is the long time-consuming bitstream implementation. To enable flexible implementation of configuration and reconfiguration, the key advantage of the prototype concerning the gateway is the source independence, and it is programmable through only one external port to produce the VPN system. This corresponds to a change in the architectural options and tools used for real-time communication applications. The consumer platform is in this context. To make it more flexible and better suited to the required operating environment, IP cores are often designed specifically for real-time communication applications, with an emphasis on reducing the required reconfiguration time. The goal is to design a prototype with application flexibility and a low-cost ability to reconfigure the IP.

4.3. Edge Computing Fundamentals

Edge computing is a contemporary paradigm that pushes real-time data processing and inferences to a technology's physical edge. This paradigm has been embraced as a solution to end users' high availability and low latency requirements. Computing at the edge is a challenging task due to inherent system processing limitations caused by resource limitations such as limited storage and power of the technology. In contrast to cloud computing, data processing occurs on a computerized edge device or close to the data source.

All three computing paradigms—edge, fog, and cloud computing—embrace horizontal and vertical scaling. Horizontal scaling utilizes multiple distributed systems horizontally to improve performance based on a specific requirement, while vertical scaling uses more computing devices to increase the capacity of a single multi-core device. However, vertical scaling amplifies a device's capability instantly, while horizontal scaling feeds a

requirement with distributed devices. Horizontal scaling can be demonstrated with data centers because data resides in multiple centers to improve network working capacity. Horizontal scaling is very useful when it comes to edge computing because the phenomenon of placing a computing device close to the data source is promoted with multiple distributed computing devices. The proposed edge computing diagram highlights how different edge devices—namely smartphones, drones, robots, and automobiles—connect and the cloud, share data, and perform computing in the form of their capability.

4.3.1. Concept of Edge Computing

Edge computing in real-time communication has gained significant attention from researchers and developers to achieve desired communication requirements like low latency, high bandwidth, and location-based services. Edge computing integrates computing and storage at the edge of the network, where data is collected and analyzed for processing. Instead of deploying centralized clouds, edge computing reduces the distance between data generation and data processing, thereby enhancing the quality of service. This deployment will meet real-time constraints such as low latency, which plays a key role in real-time communications. Currently, edge computing architectures include micro data centers, telecommunication central offices, distributed cloudlets, distributed mobile clouds, and Internet of Things gateways. The core objectives of edge computing are to minimize latency, and network congestion, minimize data backhaul, and improve transmission optimization.

Edge devices capture real-time events at the sensor level, process and analyze them at the network edge, and share them, creating an environment for real-time communications and services. Hence, real-time communication-based platforms can be hosted in edge computing locations to serve end users in a more optimized way for various real-time interactive applications supporting media communications like voice assistants, telehealth, virtual reality, augmented reality, smart grid, public safety, emergency responses, real-time gaming, and 360-degree video streaming. To facilitate and support these real-time communication applications, various frameworks with the IoT, IIoT, and network edge entities should be established, requiring computing resources. In addition, deploying edge computing to support real-time real-world communication demands careful consideration and design of hardware, software, and algorithms given the comprehensive characteristics of real-time communications to satisfy the communication requirements.

4.3.2. Benefits of Edge Computing

Edge environments leverage the concept that the closer users or data activity happens near the sources, the better the satisfaction users gain from the applications. Although the edge concept has been used interchangeably with fog computing, the main difference is that the edge refers to a new architecture providing compute and storage options close to IoT devices using physical proximity criteria. This is accomplished by using gateways or routers to connect devices. In addition, high-performance services with microsecond delays are possible when placing the microservices and resources close to users. Benefits include lower costs, faster response times, more security, and flexibility using services designed for edge architectures.

Under a traditional cloud computing scenario, real-time large data transmission is challenging. Latency issues arise in applications where split-second decision-making is required, leading to results that are infeasible for performing complex analysis in real time. In financial applications like real-time fraud detection, trading analysis, and consumer applications such as e-commerce, social network processing, and online gaming, high-performance infrastructure is mandatory. Moreover, serverless or event-driven compute solutions have attracted interest from teams that want to scale up the infrastructure demands quickly without requiring in-depth knowledge of the underlying physical infrastructure. By using edge, infrastructure barriers no longer limit the applicability of real-time big data analysis.

4.3.3. Edge vs. Cloud Computing

Edge computing, also known as fog computing, refers to the new architecture in which traditional cloud data centers and edge equipment of the network are utilized together to process and store data. Cloud computing is a kind of computing pattern that allows users to interact with applications using a browser regardless of their location. In fog computing, a bridge is required between the edge and cloud, and the upper device and edge intelligent device can communicate through the upper device. By comparing edge, cloud, and fog technologies, studies show that the edge has the characteristics of reducing network bandwidth consumption, low latency, and proximity to user equipment, meeting the real-time requirements of complex perception, decision-making, and resource-constrained collaborative processing. Compared to cloud computing, the edge has the characteristics of high fault tolerance and controllability, which can intelligently distribute tasks, make quick responses, and optimize task arrangements promptly, dealing with complex and changeable intelligent processing services.

In summary, through edge computing, it is possible to reduce the pressure of cloud computing processing services, optimize the real-time response speed of untethered mobile communication services, optimize the processing resource allocation strategy of service quality, avoid the harm caused by inappropriate identification access, reduce the delay waiting time of mobile real-time communication services, and enhance the intelligent serviceability of untethered mobile communication users. Mobile communication systems often provide high real-time demand services. In the next section, the specific process of how to use edge computing technology to optimize the processing and networking strategies in real-time untethered mobile communication service scenarios is detailed. The rapid increase of untethered mobile devices introduces more diversity to remote VR service requirements. The continuous capacity upgrade cost of cloud centralization makes communication and computing resources in the cloud too expensive to expand. Furthermore, some untethered mobile communication service users cannot enjoy the current services and experiences provided by public or private cloud networks.

4.4. Latency in Communication Networks

Applications expecting prompt interactions with a large and geographically dispersed user group are increasingly using modern-day communication infrastructure. Access to 4G, LTE, NB-IoT, or Wi-Fi enables mobile devices, IoT sensors, and actuators to be easily added to a network and communicate with each other. However, each communication network imposes a timing overhead called latency. While mobile broadband communication provides a mode of operability with user plane latency of down to only 1 ms, this comes at great expense. Achieving single-digit latencies with LTE involves having base stations operate in TDD or high-frequency bands, typically at full power. These conditions not only increase deployment costs but also reduce network capacity. Additionally, latency along the path considered the 'last mile' to the user greatly depends on the actual status of the network and the user's device, as well as the user's application.

The Tactile Internet aims for communication with a latency of less than 1 ms, which ideally is similar to deadlines achievable by the human perception system, namely by touch and sight. The overarching goal of TI is to enable humans to interact through a distributed computing system with awareness and feelings, situated afar, ideally everywhere across the globe. While it is commonly assumed that TI must wait for 5G or beyond to arrive, this paper proposes that the performance requirements of TI can be met by communicating important data in real-time through low-latency proxy servers situated at the network edge, using current communication infrastructure.

4.4.1. Factors Contributing to Latency

In a general Internet of Things (IoT) based framework, elements like sensors, microcontrollers, actuators, and other low-power wireless access network (LP-WAN) units send data and event notices to cloud-based servers. These cloud-based servers host the data analysis and decision-making process, and they send back control instructions, including run-time updates based on event detection and decision-making actions on the network of the IoT-embedded devices. This long data round trip might be unwanted, especially in use cases where immediate real-time decisions need to be supported. Removing, or at least shortening, the transmission from IoT edge elements to remote cloud-based servers mainly provides three benefits: 1. enhanced reaction times to execute measured actions in a shorter period; 2. reduced communication traffic over the network; 3. increased dependability against end-to-end network latency impairments.

Displayed thresholds for delay in real-time systems usually differ depending on the specific situational context. While the definition of done (DoD) for a simple time-sensitive metronome application is set to a value close to one-fourth of the duration of one single heartbeat, the display of speed limit warnings for special-purpose vehicles or safety-relevant functions in autonomous industrial traffic systems might also need a significantly smaller delay. In contrast to these tight limits, higher reaction times may be appropriate for non-critical use case fast-event occurrences.

4.4.2. Measuring Latency

An essential part of deploying real-time applications is understanding the amount of time it will take for a message to travel between two points over a network. This delay is otherwise known as latency in the networking world and is generally the most important characteristic of the network in these applications. Attempts have been made to decrease the overall time and work associated with communicating through a network. Although these have generally been successful to some extent, they also have a significant trade-off. Decreasing the amount of data that may be communicated can lead to a decrease in the total amount of useful work that may be performed. In the case of multimedia data, such as live voice, video, or collaboration tools, decreasing the quality of the data communicated may not allow those at the receiving end to understand the content from those at the sending end.

The operations under consideration must be both realistic and sufficiently large to approximate what would happen in a real-world scenario. It is also important that such operations be network-bound and mostly or entirely compute-bound. In other words, some sort of communication along the network needs to occur, but the amount of

processing that each message receives is minimized. Additionally, measures of time that accumulate the cost of sending data across the network, such as the time for a broadcast or a large-scale reduction, should be included in the performance model. The specific measurements described in this section focus on measuring the impact of the network on the communication latency between two optical sensors. Nonetheless, the techniques for measurement may be applied to other types of network communications as well.

4.4.3. Impact of Latency on User Experience

Real-time and interactive applications, such as gaming and trading, are latency-sensitive, with strict requirements in terms of latency or interaction time. For instance, in gaming, the action response time should be no longer than 50 ms to 100 ms for players to be engaged and not experience lag; for e-trading, limit order latency should be as low as 50 ms to 70 ms for an algorithm to gain market efficiency. An efficient trading algorithm can be influenced by every millisecond. Even in non-latency-critical applications, network latency will have an impact. For instance, rendering a webpage can take up to 200 ms, which makes it important to minimize network latency so users do not feel slow. Instant messaging combined with phone calls, video conferencing, or multimedia sessions will make communication easier and closer. All these applications can be enhanced through reduced network latency.

Users may change their behavior due to lengthy latency or interaction delays. This is not a purely technical issue. For latency-critical applications, users may give them miss or feel regret after a bad experience. For example, web delays have a direct impact on increasing web transaction abandonment. Users leave a web transaction, leading to a loss of potential earnings due to the waiting time involved. The abandonment rate grows very quickly when users need to wait longer. If a user has to wait more than 200 ms, their attractiveness to a website starts to decrease; it drops off at 400 ms and almost disappears at 600 ms. A full 100 ms to 1-second delay in website loading time can cause a conversion rate reduction of 7%; a 1 second to 3 seconds delay can cause a conversion rate reduction of 14%; and a 1 second to 3 seconds delay can cause an 18% increase in page value.

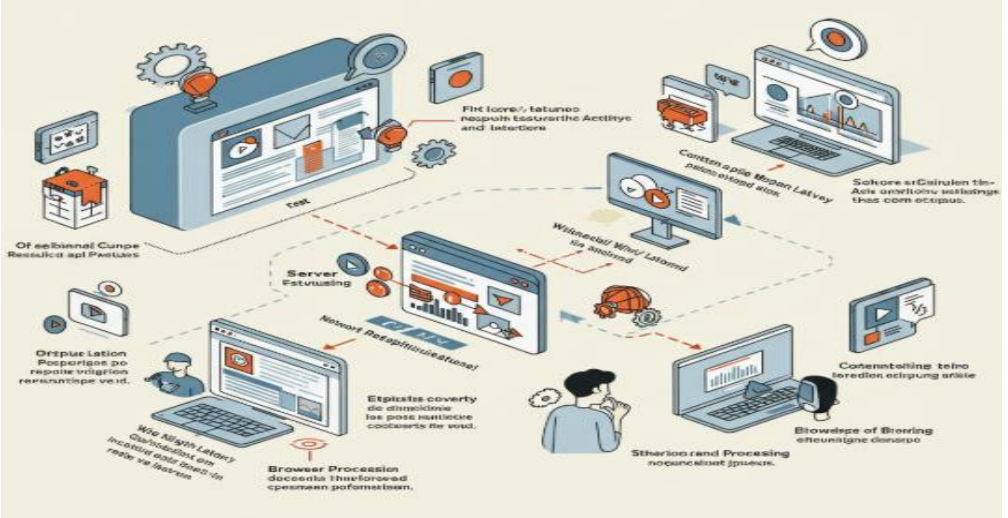

Fig 4 . 2 : The Impact of Network Latency on Web Performance

4.5. Deploying Edge Computing

In this section, with a case study, we present a reference model to deploy DApps at the edge. The principal idea of placing components at the edge is to improve the user experience by decreasing latency and gaining resilience. In contrast with traditional counterparts, the DApp executes closer to clients, and the local operator does not respond to possible information transaction short-circuiting attacks. First, aiming to provide better quality of service to the applications that interact with the blockchain, the network provides a scalable and flexible approach that allows light clients to obtain proof for their transactions, for several requests proportional to their commitment to the network, within well-known quality parameters. Stakeholders implementing the model can use the network as a membership service, which they can model at will, allowing them to obtain proof. Additionally, the network is transparent, since all stakeholders can check the QoS level observed by the light customers. As part of the solution, it proposes an artificial and auditable way of testing the SLAs, ensuring that the commit and proof of the message take a superlinear cost to the application.

Second, looking ahead, placing the application's monitoring at the edges and reducing the number of monitoring servers that execute local intelligence applied to the blockchain, as well as the amount of information to be exchanged with the network service. The weak points of the proposed solution are external: the uncertainty about the reliability of the private information that feeds the proposed model, and the existence of a historical mixing attack that could hinder tier-one services. Council service, which

allows stakeholders to partner with those who want to obtain privileges on the level of QoS, for them, not on behalf of the customers. Finally, in calming a potential dispute that has yet to come: when some providers need to take down some of their guarantees. It is worth noting that both the amount of money from the solution implementation, as well as the player's invocation, timing, and perimeter can adiabatically optimize the system.

4.5.1. Architecture of Edge Computing

Edge computing shows great potential for enabling real-time communication. This is attributed to the distributed architecture, which shifts the computing burden from the cloud to the network edge, and to the proximity to data sources, resulting in shorter network and processing delays. By deploying computation in a physically closer area, edge computing decreases the data propagation delays caused by network latency. These computational resources can be used to conduct preliminary data processing and analytics, transmitting only the refined and abstract data to the cloud or remote servers. Therefore, edge computing minimizes internet traffic and cloud server workload, resulting in lower backhaul and cloud service provisioning costs. The potential for providing real-time data communication while requiring less energy and space has made edge computing an important technology in the context of IoT and smart cities.

The concept of edge computing also raises concerns about its architecture, including device, edge, and cloud layers, and deploying various algorithms and mechanisms in these layers. This innovative structure results in numerous design and implementation issues. In this section, we discuss several existing architectures of edge intelligence in the context of communication quality and introduce our novel architecture to provide low-latency wireless communications. Device and edge intelligence play a significant role in bridging wireless communication and the promising edge computing capabilities. To deal with the high communication latency and limited computation capacity of the devices, the design of low-latency and energy-efficient architectures is needed so that data can be processed with high accuracy in a timely fashion.

4.5.2. Edge Node Placement Strategies

As for the edge node placement strategy, the simulation result shows that it will need to consider various factors such as the positioning of the edge nodes, the edge node capacity, and the change of traffic models. Generally, it could be a "minimal mean distance placement," "random placement," and "hotspot placement." Selecting the

minimal mean distance placement can mitigate the effect of interference among edge nodes, but since the traffic flow of users is random, it might increase the packet end-to-end latency because of an imbalance of the distribution. For the random placement case, the probability of achieving successful transmission goes to zero as the distance between edge nodes becomes increasingly large. It is still possible to work on this strategy, however, by targeting real traffic locally and assigning more edge nodes to these areas. To adjust to changes in the popularity of areas, the efficient method is also to place more edge nodes in top traffic areas, reducing the bandwidth consumed by data backhaul to long-distance collection systems.

The main problem for edge node placement and capacity is focused on placing the fewest number of edge nodes to minimize the deployed cost. There can be plenty of heuristics trying to solve this problem. It can be implemented by using either a greedy solution or by employing some pre-established conditions in the decision-making methodology. Therefore, it will calibrate the requirement for highly responsive MEMS mirrors, as large bandwidth and complex design and signal processing are needed to support real-time and low-latency processing. The small number of edge servers also leads to problems if many users are requesting resources in the same region with slow processing. For the change of traffic congestion models, new areas are emerging and growing. For instance, laying out edge nodes nationally while concentrating some of the nodes in very congested cities. Fewer user requests are calculated and would have to be backhauled to the edge gateway in the data center instead of processed by edge servers in this case. The edge node can and should be located between the fixed infrastructure and the location in which data originated or received to execute different tasks, resulting in lower power consumption during data backhaul.

4.5.3. Integration with Existing Infrastructure

A significant issue to address concerning edge computing is how it fits into existing infrastructure for IoT systems. While it is possible to install new infrastructure made from the ground up to support edge computing, most of the time it will have to interface with existing WSNs and gateways for many practical systems. This includes those systems that are running on frameworks that provide cloud-based analysis functions, often with custom scripts or bespoke virtual instances.

Many WSN systems do not have centralized software running on VMs or cloud services. Rather, they collect data at the nodes and shuttle this data to one or many gateways that are connected to border routers, which then interface with the external Internet. This is done using the Internet protocol layer. While some gateways are simple, dedicated

devices, others have more than one network interface (e.g., some run a Wi-Fi LAN for a user WLAN along with a modem for Internet access and analysis purposes). Some hardware systems integrate border routers within a wireless protocol bridge that interfaces with a home router. A WSN can include systems with one of any number of the above kinds of 'WSN gateway' devices in any configuration.

4.6. Performance Optimization Techniques

The goal of the performance optimization techniques is to enhance real-time communication through edge servers and IoT devices. Some IoT devices cannot be decentralized into two components with different computing and communication power. If the computing part of an IoT device is heavier, more computing resources are needed, and the construction and maintenance costs increase. On the contrary, if the communication part is lighter, more data needs to be transmitted, and the transmitting energy and time decrease. Few works have designed QoS-guaranteed real-time communication systems in such a scenario, and no works optimize runtime performance by replacing computation with communication after the construction of the system. In this section, we present part of the implementation techniques used in an edge-enhanced project. The sound recognition computational model can also be internalized but at the cost of an unacceptably high energy drain. The goal of the proposed project is to design a general approach in the aforementioned scenario.

We use the proposed performance optimization techniques for the project. The object recognition algorithm is moved to the server. Real-time audio processing is performed locally with the pre-developed detector. The detection results are sent to the edge server. Since the device understands commands and they do not need to be recognized, we detect whether the person would like to issue a command in the first few seconds of each audio (usually 2 seconds) and turn off the recognizer if not. The communication map is optimized due to the existence of the object detector. If there is no object of interest, the device does not need to transmit the non-target result or feature to the edge server. If there are objects of interest (including potential hazards), the device transmits the bounding boxes of those interesting objects. The proposed project stores the models and attaches several additional components to the access point. Each of these project components is responsible for a kind of computation, such as activating or deactivating the object detector. The visual indicator is also attached to the access points. The indicator lights up in response to the command and blinks with intensity proportional to the distance to the potential hazard. With noise suppression and keyword spotting supported right on the edge, users can fluidly experience a prompt and elegantly delightful interaction only with a pair of wearable microphones and the help of edge-

native AI assistance. The energy cost remains moderate, then, with the use of those enhancements seamlessly reflecting in the end-to-end latency.

4.6.1. Data Caching Strategies

To maximize the benefits of edge computing, efficient and intelligent data caching strategies should be embraced. This is because, given the limited capacity of the edge network architecture, data caching must guarantee that only the most relevant data to the users is stored at the edge cache, thereby reducing the latency involved in fetching the data. To achieve this, an efficient data caching policy should be able to effectively predict the most relevant data that is likely to be accessed by the end users. There are two major components in the process of cache data prediction. The first component represents the process of sequential pattern discovery, while the second component represents the mining of future demand data from the sequence patterns. These two processes are accomplished through a significant technique generally known as the Negative Association Rule technique. As a significant data caching strategy, NAR proves to be able to accelerate the speed of the data discovery process.

Caching of data at the network edge is an approach used in edge computing to foster real-time communication between devices and reduce the latency involved in delivering data at the end users' point. To ensure that only the most relevant data is cached at the network edge to achieve these benefits, edge caching strategies should be embraced. The focus of this paper is to present a practical solution to this problem by proposing an intelligent and efficient edge caching strategy that facilitates the caching of the most relevant data at the network edge. The proposed ECS adopts a special technique known as the Negative Association Rule technique, which fuses two important processes involved in cache data discovery, namely, sequential pattern discovery and future demand data mining, to accelerate the process. By experiment and comparison with other state-of-the-art caching approaches, our approach can demonstrate a significant caching efficiency in maximizing the highest cache hits and in keeping the cache misses at the second lowest level.

4.6.2. Load Balancing Approaches

Usually, IoT applications involve geographically distributed IoT devices, where real-time communication with IoT devices is important and is expected to have low latency and high throughput. However, traditional cloud-based computing solutions tend to have a high latency response due to the large number of hops the packets have to travel.

Moreover, within data center networks, increasing the number of hops between client and server also contributes to a longer tail network latency distribution. Using IoT, the end user may generate a request for bulk data transfer or real-time monitoring. In the first case, the data are sent from the cloud to the server which can support a delay in response. In the second case, however, a request for a particular image or footage must have a shorter delay.

Service access from user to server increasingly resembles a complex mesh rather than the historical hierarchical client-server model. Instead of having requests funneled through multiple load balancers and intermediaries to the backend server, multiple parallel exchanges with the server increase the number of cross-data data center paths clients use to reach services. To further enhance real-time communication and offer an end-to-end delay close to zero, this paper proposes processing at the edge. Such capabilities allow providing pricing models with different priorities for different groups of users, functionality that is potentially attractive from the viewpoint of both the seller and the buyer of the IoT service.

4.6.3. Content Delivery Networks

Content Delivery Networks (CDNs) have proven to be a cost-effective solution for enhancing the user experience in terms of website loading time, quality of videos, and overall multimedia streaming. The basic principle of CDNs is to replicate the most popular content on the web, usually stored by website owners in their data centers, and place it in geographically dispersed servers managed by a CDN provider. When a website user requests access to multimedia content, the request is redirected and served to the user from a resource point closer, in terms of Internet distance, than the website's original data centers. This is usually done to exploit resources that are located closer to the user to speed up the delivery time of the content and provide quality levels more suitable for the user's device. In addition, the fact that a CDN server may be part of the backbone or close to the peering points of the web network assures that QoS levels are stricter than if these requirements were accomplished through standard Internet data flows.

More recently, the realization that dynamic, user-generated content requires caching infrastructures that collaborate in the most effective way to store, replicate, update, and delete new content has pushed the development of elite CDNs to adopt a new breed of technology for deploying caches at the edge of the network. Push and pull techniques have been found inefficient in managing this set of content that is characterized by a temporal popularity distribution with a high variance. When the amount of data stored

at the edge of the network increases enormously, this turns into the enabling technology of mobile edge computing. In that, the original idea of distributed computing is extended to provide infrastructures that are capable of executing sections of code that may be executed or augmented at a closer point on the Internet.

4.7. Security Considerations

Initially, the requirement for efficient edge computing suggested that a cloud service in a private network be the first point of attack. Nonetheless, the idea of creating a private cloud service can lead to service availability, reliability, and scalability issues under various attack scenarios. More seriously, placing an excellent assault point upon edge computer services may lead to data privacy issues. Given that much of the data stored by the cloud service is inconsistent with the user's information, the user's privacy could suffer as a result of this. These factors underscore the importance of considering security during the architectural design of building a next-generation edge computing platform.

Among the multitude of individual breach assault versions, such as distributed denial of service attacks, unintentional attacks, and man-in-the-middle attacks of varying dimensions, the probability of a cloud data center going offline due to management problems or a DDoS assault is high. Unauthorized access to this information stream can significantly benefit producers of disruptive software, such as proxy systems and forging network providers that generate suspicious network traffic. Unencrypted, undetected data exchange will result in a data breach. Data confidentiality could be compromised if the edge servers are globally compromised, which is helpful to attackers in violation of regulations.

4.7.1. Threats in Edge Computing

The main issue of enabling computation at the edge is the increased potential threat surface. As the number of edge devices grows, the threats rise significantly compared to the multitude of edge infrastructure. However, the central cloud saves the day by reclaiming the cyber threats in case of a data breach. Edge security might lack the robustness of traditional computing, which might mean one small breach could lead to significant data loss, including all the device data surrounding it. The current security issues around edge devices create skepticism among users who might not opt for state-of-the-art devices to avoid security implications.

Security issues with data packets are one of the topics currently being discussed. The part of the data that is offloaded to the core servers from the radio base stations is

vulnerable to data theft. If we add to this the issue that it is not always encrypted and is visible in plaintext, the vulnerability of such a model is expanded exponentially. Many times, that data also identifies users, putting the radios and their teammates at risk. The very high data rates requested may exclude encryption as a potential remedy. Imposing edge computing on existing infrastructure may lead to breakdowns as well. With the increasing number of MEC servers and edge caches, the virtual network infrastructure needs to have an architecture able to handle the additional edge servers without significant additional provisioning.

4.7.2. Best Practices for Securing Edge Nodes

The variety of edge computing hardware, each with its specifications, prevents simple blanket prescriptions for securing them. However, starting with the security controls in effect at the time of the hardware's fabrication, these best practices can provide a basis for building a secure edge computing environment: - Secure Boot: Ensure that the hardware has a secure boot process that can validate and authenticate firmware updates. Secure updates can prevent attacker-supplied software from tampering with the edge device's performance and functionality. - Lockdown of Boot and OS: Boot devices can be write-protected to prevent unauthorized modification. Initially, configured operating system images can be created by a trusted source and built with configuration and tuning settings specific to the performance demands of the network function. - Strong Cryptography: The edge computing device can employ hardware acceleration of encryption to deliver high-integrity data into the network and be designed for cryptographic modules that exceed the standard security certification. - Remote Management Access: For hardware encryption modules and CPUs that support remote management technology, ready administrative USB ports, or an available serial console with capabilities, support of remote management is crucial. They can be used or accessed to validate, interact with logs, or repair an edge device in the field while minimizing physical branch access where such devices may be vulnerable to unauthorized access and attack. Ensuring that edge devices are secure in maintaining uptime and security in place requires the prevention of unintended data destruction or system-level reset.

4.8. Future Trends in Edge Computing

In this chapter, the future-edge computing research and innovation pathway is presented. To deliver substantial real-world benefits, synergy needs to strike the right balance between future technologies and business outcomes that can recognize the economic value created by effective edge solutions. Globalization has transformed the way that

enterprises and consumers are educated throughout the world and empowers them by providing edge services. However, new digital forms of sophistication and individualism require more personalized, localized, and time-critical services. In the fog and edge of the next-generation savvy network and application management, the fact that directional distribution is decreasing and that most of this data is microservice-level small data is fully proven. Facing fog and edge computing, this chapter analyzes these challenges. Data are generated at the edge of the network, and the exponential increase in the use of mobile devices combined with the need for data at the edge of the network with lower delays has prompted communication, computing, and storage resources to be developed rapidly in preparation for capturing and processing significant data on mobile devices. Internet companies are exhausting unprecedented resources to build the next-generation edge cloud infrastructure. However, data from one enterprise customer or a single entity are geographically distributed in all types of energy-constrained networks, where energy consumption needs to be regulated as a high-priority concern. Mobile device battery life has a significant impact, particularly in the case of inferential and cognitive activities, and the progression of integrated circuits and software solutions for native recognition, which is much more lightweight. High-power architectures still require huge energy to operate.

4.8.1. 5G and Edge Computing

5G encompasses several architectural and deployment features boosting mobile telecommunications. One of the most attractive system features of 5G is its potential to deploy and operate small cell base stations and networks. These features make 5G a key enhancing technology for edge computing as it may provide several opportunities. First, 5G can offer low-latency connections for mobile users, which are crucial to accessing edge computing resources with a limited delay. This enables mobile users to efficiently offload generic tasks to the edge of the network and enhances the computational power of smart mobile devices by allowing them to seamlessly offload computation tasks to nearby edge servers. Second, by considering 5G and edge computing jointly, we may be able to deploy so-called micro data centers. Third, by considering 5G and edge computing jointly, we can explore cooperative caching among edge servers and accordingly enhance cache performance.

Edge computing enables real-time and data-intensive applications to access resources, enabling the smart data approach. These key features of edge computing make it an enabling technology to deliver real-time communication applications and services. Moreover, the ever-growing improvements in edge servers and data-intensive technologies are driving the deployment of so-called mobile intelligence. By extending

the edge computing paradigm to leverage the most advanced communication technologies, we believe that the ultra-low latency constraint faced by mobile systems can be satisfied while enhancing cloud gaming, health, and IoT application systems. To this end, the enabling technologies of edge computing must be integrated into a cohesive ecosystem.

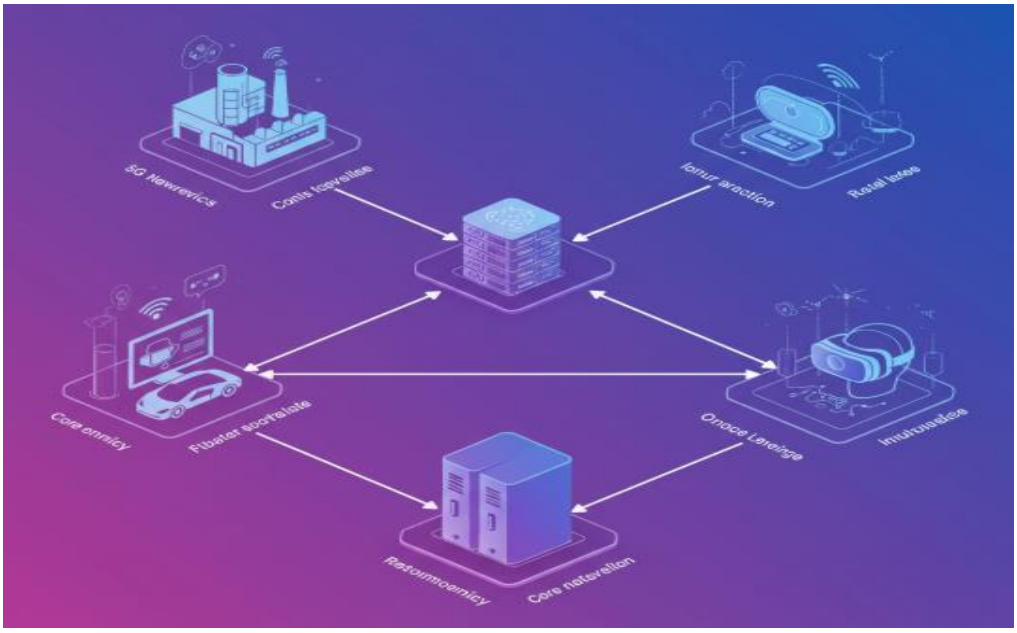

Fig 4 . 3 : 5G and edge computing synergies

4.8.2. AI and Machine Learning at the Edge

We have discussed communication-intensive IoT tasks such as streaming multimedia data, generating data-driven media, etc. Now, we will dive into why we need AI at the edge. In any architecture providing real-time communication, we need intelligence at the edge. Intelligence at the edge makes real-time systems perform optimally. The thing that is most required to be emphasized is that any AI task, whether training or inferencing, is very expensive in terms of computation, energy, and time. AI and real-time communication, thus, put heavy demands on the cloud. The limited capabilities of endpoint systems and the expense of cloud systems make these requirements hard to satisfy simultaneously, especially in a streaming setting. Edge computing, as a computing paradigm, eases the situation by providing computing services near endpoint systems.

After sampling, converting, and compositing media data, each edge device could have its copy of these pieces of data. In edge computing, we have an "edge" with a limited

amount of computing resources. In our context, end users (who request the AI/ML system for real-time decisions) could be those that are enabled by mobile, embedded, or IoT devices. These endpoint systems send their data to an edge cloud to be connected to the AI/ML server. Since the data involved is generated by clients that exist in space, this is somewhat similar to a CDN. Edge computing brings cloud computing capabilities closer to endpoints. The cloud, as an extension of the edge, is responsible for managing the connections among different endpoint devices and for the orchestration and consolidation of data and services. The cloud can also be used to get updates of the model used for inferencing. The limited capabilities of endpoint systems and the expense of cloud systems make these requirements hard to satisfy simultaneously, especially in a streaming setting. Edge computing, as a computing paradigm, eases the situation by providing computing services near endpoint systems. Edge computing architecture provides for entities called "edge clients," which can request a remote service at the edge. Edge clients also have storage resources and application execution environments.

4.8.3. Emerging Applications

This section discusses emerging applications that leverage the strengths of edge-native computation for real-time communication. We have identified that edge-native computing can take significant advantage of RTI for communication purposes. For real-time workloads, cloud computing cannot always meet the timing requirements. In contrast to the traditional cloud, when the data center is not located very close to the edges, it encompasses significant delays in communication. Decreasing the remote distances may lead to lesser network delay and enable clients to support additional services with significant requirements on latency. The applications in edge computing, therefore, turn out to be responsive to clients. With these characteristics, edge computing has emerged to address latency-sensitive applications, and opportunities are opening up for companies to explore the technologies. In this work, we investigate whether cloudlets could be exploited to support real-time interactive web applications in a hybrid cloud environment.

Mobile devices tend to serve various services, such as interactive web browsing, video streaming, and multiplayer gaming, which have powerful requirements for latency since they are regarded as hand-held mobile devices. Although mobile networks have promised to provide a better network experience, long tail latencies can frequently be noticed depending on traffic load, signal quality, and device radio blocks. In addition, mobile clients are tied to web applications at a more reasonable level in terms of being the most extensively used platforms. Consequently, generating the response at the server side and reducing the time for data transfer can greatly reduce the latency experienced

by the clients, as the relatively lower latency experienced by infrastructure providers has little advantage in enabling the performance of hand-held mobile devices, making them most preferred by consumers. In contrast, it is observed that the uplifts from video and multiplayer gaming for all the studied cloudlet locations are not very visible. Specifically, the aforementioned video and gaming services mostly involve massive data streaming or querying. Therefore, considering offloading to edge computing servers possessing network delay benefits has a limited impact.

4.9. Conclusion

Real-time communication has played a significant role by enabling the collaboration of two endpoints for the instant or simultaneous exchange of information. This is achieved by conducting marking, coding, echo cancellation, conversational user experience, and conferencing demand by deploying real-time communication frameworks such as video conferencing and voice call services. To counter the various video conferencing and voice call service-related issues and enhance the quality of the real-time communication services, we present an Edge Computing Framework architecture. ECF is designed to offload edge servers with near-end user management capabilities of two endpoints by processing traffic steering and Quality of Service-aware Virtual Network Function chain and intelligent service function selection to prevent traffic congestion in a communication network. Furthermore, it discourages over-subscription and provides an explicit service function chain in the cloud.

In this work, we have applied the proposed ECF to quality-controlling chain frameworks. We have tested the architecture functionalities and performance in a realistic and controlled experimental environment. The architecture has shown an efficient mitigation of current traffic congestion, tail latency, and throughput problems that are experienced by the existing communication QoEs. Furthermore, ECF has displayed strong performance in maintaining communication throughput by managing latency, jitter, over-subscription, packet loss, and backend data center load. In the future, we aim to implement the prototype for the 5G network and proactive traffic line prediction with supervised machine learning techniques. In conclusion, the proposed ECF acts as a complementary solution and supports the orchestration platform to deliver the service proactively and reactively and meet what users demand.

4.9.1. Final Thoughts and Implications for the Future

To adapt to the consumption of immersive content, edge cloud services will be of paramount importance, acting in a distributive way and allowing the delivery of services that depend on agile response times, such as in the augmented reality and end-to-end video market. In the context of real-time communication, edge computing is of secondary importance, but has a huge potential, since it can change the current model of a centralized cloud, installed far from the majority of the sites and facilities connected in the world, closer and closer to the entire set of terminals and connected intelligent objects. In industries, this technology reduces costs and increases data utilization and the advancement of IoT technology requires more sponsors to expand and develop faster. With the movement patterns and inquiries made in many senior economics infrastructure systems in subway systems, improved calculations based on intelligent factories are achievable.

In this study, we investigated several mechanisms that create strategies for better handling the mapping of addresses to isolated sites/physical processors from edge cloud environments. In the second, we implemented a basic video detection service that is currently part of an overall prototype being implemented to work on a project. We are currently in the process of integrating the service with a management and monitoring tool that was also part of a call. Our future work on this research includes investigating mechanisms that allow for more efficient updating mechanisms of the mapping of home settings to physical isolates. Moreover, we intend to investigate the presence of virtual functions and how they can act as end-effectiveness mechanisms for running applications on the edge. We intend to make our prototype available, update it, and test it as part of real applications.

References

Abbas, N., Zhang, Y., Taherkordi, A., & Skeie, T. (2018). Mobile Edge Computing: A Survey. IEEE Internet of Things Journal, 5(1), 450–465. https://doi.org/10.1109/JIOT.2017.2750180

Chiang, M., & Zhang, T. (2016). Fog and IoT: An Overview of Research Opportunities. IEEE Internet of Things Journal, 3(6), 854–864. https://doi.org/10.1109/JIOT.2016.2584538

Mach, P., & Becvar, Z. (2017). Mobile Edge Computing: A Survey on Architecture and Computation Offloading. IEEE Communications Surveys & Tutorials, 19(3), 1628–1656. https://doi.org/10.1109/COMST.2017.2682318

Mouradian, C., Salahuddin, M. A., Limam, N., Boutaba, R., Iraqi, Y., & Gagnaire, M. (2017). A Comprehensive Survey on Fog Computing: State-of-the-Art and Research Challenges. IEEE Communications Surveys & Tutorials, 20(1), 416–464. https://doi.org/10.1109/COMST.2017.2771153

Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., & Sabella, D. (2017). On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration. IEEE Communications Surveys & Tutorials, 19(3), 1657–1681. https://doi.org/10.1109/COMST.2017.2705720