Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.051611

ARTICLE

Information Centric Networking Based Cooperative Caching Framework for 5G Communication Systems

R. Mahaveerakannan[1], Thanarajan Tamilvizhi[2,*], Sonia Jenifer Rayen[3], Osamah Ibrahim Khalaf[4] and Habib Hamam[5,6,7,8]

[1]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, 602105, India

[2]Department of Computer Science and Engineering, Panimalar Engineering College, Chennai, 600123, India

[3]Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, 600119, India

[4]Department of Solar Energy, Al-Nahrain Research Center for Renewable Energy, Al-Nahrain University, Jadriya, Baghdad, 64074, Iraq

[5]Faculty of Engineering, Université de Moncton, New Brunswick, E1A3E9, Canada

[6]Hodmas University College, Mogadishu, 521376, Somalia

[7]Bridges for Academic Excellence, Tunis, 1002, Tunisia

[8]School of Electrical Engineering, University of Johannesburg, Auckland Park, Johannesburg, 2006, South Africa

*Corresponding Author: Thanarajan Tamilvizhi. Email: tamilvizhi.phd.it@gmail.com

Received: 10 March 2024 Accepted: 12 June 2024 Published: 12 September 2024

ABSTRACT

The demands on conventional communication networks are increasing rapidly because of the exponential expansion of connected multimedia content. In light of the data-centric aspect of contemporary communication, the information-centric network (ICN) paradigm offers hope for a solution by emphasizing content retrieval by name instead of location. If 5G networks are to meet the expected data demand surge from expanded connectivity and Internet of Things (IoT) devices, then effective caching solutions will be required to maximize network throughput and minimize the use of resources. Hence, an ICN-based Cooperative Caching (ICN-CoC) technique has been used to select a cache by considering cache position, content attractiveness, and rate prediction. The findings show that utilizing our suggested approach improves caching regarding the Cache Hit Ratio (CHR) of 84.3%, Average Hop Minimization Ratio (AHMR) of 89.5%, and Mean Access Latency (MAL) of 0.4 s. Within a framework, it suggests improved caching strategies to handle the difficulty of effectively controlling data consumption in 5G networks. These improvements aim to make the network run more smoothly by enhancing content delivery, decreasing latency, and relieving congestion. By improving 5G communication systems' capacity to manage the demands faced by modern data-centric applications, the research ultimately aids in advancement.

KEYWORDS

Information-centric networking; caching schemes; 5G communication; non-negative matrix factorization (NMF); weighted clustering algorithm

1 Introduction

Internet communication has evolved, mainly as a result of changes in the importance of content dissemination. There is a tight relationship between the management of the system and caching processes in ICN due to continuous delivery hop-by-hop, which provides the fundamental premise for caching. ICN is better suited for supporting video streaming services in automotive networks. ICN's design objective is to render the communications infrastructure content-aware, enabling the network to execute information tasks intelligently, including content resolution, distribution, preservation, and caching. Each node in 5G-based ICN systems has a set amount of storage, such as a gadget, gateway, or small core network. Unlike traditional collaborative caching, members of a single cluster cannot communicate directly with one another. In other words, the only way to communicate is through the cluster head. An edge layer's size is always considered when deciding how many clusters to use. The number of devices connected to the Internet today is increasing dramatically, directly increasing the amount of content generated and information transferred on the network. Furthermore, individuals can simultaneously consume and create/provide information, adding complexity to the existing Internet [1]. The 5G network intends to double the infrastructure of the wireless network while enhancing dependability, profitability, and speed. A promising option is to place the ICN on the upper edge of 5G due to its infrastructural development and clear communication style [2]. With a fast and executed process, the calculation starts immediately in the first access instead of waiting for scanning cached outcomes. Before completing a calculation, it is discovered that the cache outcome is sent to the requester, and execution is halted [3]. Conventional cache techniques store content centered on content attractiveness, which is sometimes appropriate because users may not be enthusiastic about accessing popular content continuously [4]. Rather than using the creative content source, copy content from many other sources. Hence, a dynamic cache management technique that bases hop counts on the attractiveness of the content to reduce redundant cache information [5]. The ICN functionality, by design, enables information distribution, allows content/user portability, and reduces transmission delay while boosting overall network efficiency.

Currently, most 5G-based ICN technologies focus on studying and addressing content cache [6]. The introduction of a clever caching mechanism is to choose where to cache content by utilizing a recommended system method to anticipate consumer activities to determine the attractiveness of the content [7]. When the cache store is filled, the content replacement strategy is activated to choose which information should be removed to make room for other essential content and maximize the use of the cache [8]. A vehicular network involving the cloud where caching is performed at the user and edge layers. Due to the use of cloud-based controllers during caching, the scheme has considerable content retrieval latencies [9]. A joint caching and communication approach is used to solve the latency problem. Describing the cache position can address the where issue, but it does not concern how to choose the content to be cached [10]. From the point of view of user preferences, a few MF-based proactive caching strategies were suggested. The authors of [11] suggested a proactive strategy that outperforms the reactive caching strategy and is based on singular value decomposition (SVD). It is thought that SVD is not practicable on real-life networks, as it could result in negative ratings as forecasts for poorly rated videos. The most current research makes caching decisions based on the projected user demands and uses proactive caching to predict users' future requests. Therefore, due to the enormous computational complexity, their suggested approach is not feasible [12]. However, since edge computing may implement the realistic case study in IoT, this study focused on edge caching because it influenced cache performance [13]. A frequently used model for ICN research is the caching strategy considered the generic hierarchical network topology when analyzed. The most crucial aspect of ICNs is massive content availability. A trend towards network-wide content and application diversity is emerging with the proliferation of major content providers and user-generated content apps. By keeping caches and applications distinct, in-network caching in ICNs enables many content types from various servers to share a single node's cache. This greatly complicates the process of analyzing network caching technologies. In the last part, we covered the concept of popularity hierarchy and how it works using a diagram. A dynamic adjustment based on the request frequency is necessary to get a categorization for the popularity level, as the popularity of material changes over time. In this part, we build a packet-of-interest request frequency collector that the router nodes may use to classify the popularity of the content entries they hold. The content is divided into pieces of information used for transmission [14]. By examining the connection between video length and user viewing habits, a cache replacement technique called predicting multimedia cache based on large pieces of popular literature was developed [15].

Using edge-based in-memory caching, computing resources, like storage spaces and energy, can be conserved in the inter-heterogeneous network. Still this in the hit ratios based on the count of the number of hops communicated in the distance [16]. However, there is currently a shortage of work on caching multimedia content for resource-constrained environments based on cache position and content attributes. The main driving motivation of the proposed work is to choose a cache taking into account cache position, content attraction, and rate prediction by the ICN-CoC method. The CHs in the edge layer of the clustered nodes are selected using the Improved Weight-Based Clustering Algorithm (IWCA), which also determines the cache position. CoC-NMF uses a probability matrix in addition to forecasting future content rates to determine the likelihood of content caching. To find effective results of CHR improvement, the mean hop minimization ratio (AHMR) and mean access latency (MAL). The remainder of the paper is organized as follows: Section 2 focuses on contributions that center on caching schemes and explores ICN technology for 5G communication. Section 3 presents the system model of our suggested caching strategy in detail, considering the selection of positions and content based on the clustering technique. Section 4 describes the results of experiments and performance evaluation. The summary of the contributed work and its future enhancement is given in Section 5.

2 Related Work

Many caching techniques have been proposed in this area, but most are naturally distributed and expandable in network topologies. Different evaluation measures are used to compare our system with various caching techniques. Anselme et al. [17] recommended Deep Learning Infotainment Cache (DL-IC) in autonomous vehicles based on user cache choices to connect wayside devices to forecast stored information. Also suggested a communication system for caching infotainment content retrieval. According to the simulation findings, it can reduce latency and has a prediction accuracy of 97% for information that must be cached. Siyang et al. [18] introduced user preferences (USR-PRF) with on-path storage of various content categories. The choices are based on the item's attractiveness. The degree of matching between a piece of content and a node is determined when it approaches an edge router. It also performs better concerning to local CHR, hop counts, and caching rejections. Yao et al. [19] suggested a Cooperative Cache Mobility Prediction (CCMP) strategy for Vehicular Content-Centric Networks (VCCNs). Using prior paths, use partial matching to predict the likelihood that mobile nodes will arrive in certain hot-spot zones. Simulation metrics show parameter results regarding success rate and content access time. Yahui et al. [20] developed the "Efficient Hybrid Content Placement" (EHCP) technique to improve contents anywhere along the data distribution process while decreasing the number of times identical contents are replicated at various sites during data dissemination through caching.

The consequences of redundancy, hop delays, and CHR are illustrated. Nguyen et al. [21] implemented a Progressive Attractiveness-Aware Cache Scheme (PPCS), a novel chunk-based caching technique to enhance content accessibility and resolve the ICN cache redundancy problem. Therefore, by allowing only one replication of a specific content to be cached, PPCS eliminates wasted cached space by keeping on-path information repetitions and enhancing cache diversity. Simulation, in terms of delay based on hop count, duplication, and CHR. Lin et al. [22] proposed a Mobility Prediction and Consistent Hash for VCCN to create a unique cooperative caching technique (MPCH-VCCN). Consistent hash algorithms are used to distribute content among devices, eliminating cache duplication. The evaluation study shows that it has a higher cache hit ratio, smaller hops, and faster content access latency. This section summarizes several comparisons demonstrating the effectiveness and rationale behind the suggested approach. Various algorithms are analyzed and taken.

3 Proposed Scheme

A quality technique known as in-network content caching involves copying content across several locations/nodes and is referred to as cache stores. By fulfilling future needs rather than sending them to the content owner, these cache stores reduce network congestion at the core, stress on the content owner, and latency in content distribution. Only a few nodes can also cache information on other units' caching activities for effective resource use. Since vehicles are truly nodes in a vehicular context, "nodes" and "vehicles" are often used interchangeably. NMF is a sophisticated machine-learning technique that falls under the MF category. NMF (non-negative matrix factorization) is frequently used in recommender systems to forecast user ratings for videos that have never been watched. NMF does not produce negative predictions, unlike other MF approaches such as SVD, which is seen to be more appropriate for video ratings in practical applications. The suggested ICN-CoC concept uses a variety of figure representations and mathematical equations with different notation and parameters to construct an improved architecture for caching techniques in ICN-based 5G communication systems, with references to [23,24].

The primary data utilized in all annotations added with the core schema of this section was acquired from the studies referenced above. As a generic approach for dimensional reduction and feature extraction on non-negative data, non-negative matrix factorization (NMF) has seen extensive application. The nonnegativity of NMF distinguishes it from other factorization techniques like SVD; with NMF, one may only combine intrinsic "parts," or hidden characteristics, in an additive fashion. Since a face is intrinsically represented as an additive linear combination of several elements, NMF can learn face parts. Negative combinations are, in fact, less natural and intuitive than positive ones. Feature selection using non-negative matrix factorization (NMF) is rather common. These days, heuristic computational approaches for NMF are standard in many data-mining and machine-learning suites. Theoretically, however, we know very little about when these heuristics work. Because NMF is a computationally demanding subject, establishing rigorous solutions is tough. Therefore, we need to impose extra data assumptions to calculate nonnegative matrix factorizations in reality. With this goal in mind, we provide a polynomial-time solution for NMF that is provably right, given suitable model inputs. Their finding illustrates a specific case where NMF algorithms reliably provide insightful results. This study may influence machine learning since accurate feature selection is a crucial preprocessing step for several other methods. Therefore, there is still a divide between NMF's theoretical research and its practical application of machine learning.

Fig. 1 explains the workflow proposed for the ICN-CoC technique with all parameters. In the ICN-5G component, basic ICN architecture helps us to perform an efficient caching scheme in a hierarchical setup where core and edge layer separation gives a detailed diagrammatic representation of the gadgets and network topology used for the caching scheme. Content servers provide effective caching storage positions for each requested content through access gateways with a cluster model.

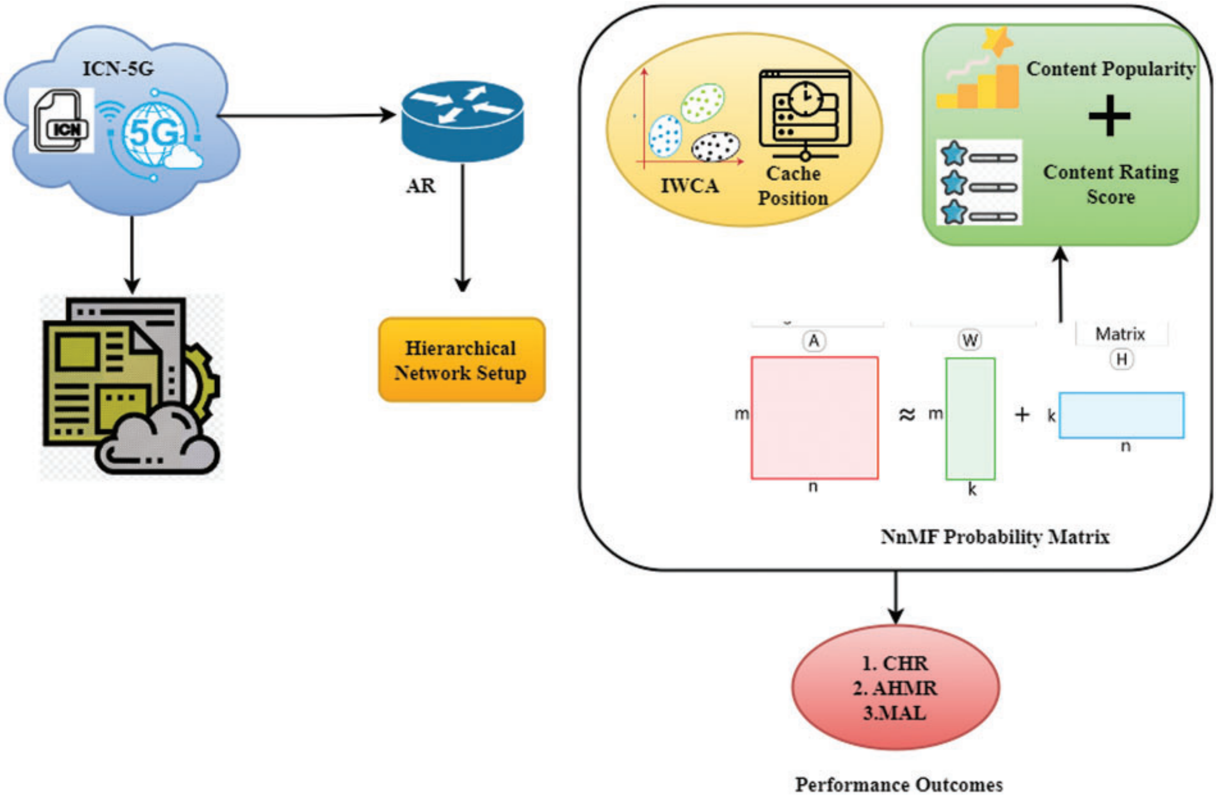

Figure 1: Implementation flow of ICN-CoC technique

For vehicle users, data transmission is transferred with the base station and roadside units (RSUs) through the 5G ICN-based core vehicle network from the content provider servers. These wireless communication capabilities require effective infrastructure along roadways. Costing a pretty penny, these roadside units (RSUs) are often overseen and financed by federal governments. With 5G, new vehicle-to-vehicle (V2V) connections will be possible, paving the way for a future when two automobiles coming from opposite directions might use their onboard computers to decide which one should yield to the other when their paths intersect. Intelligent transport systems (ITS) may provide a range of security, entertainment, and shared mobility applications for cars via wireless communications enabled by the roadside unit (RSU) cloud and its vehicle-to-infrastructure (V2I) connection. Vehicle ad hoc networks (VANETs) rely on roadside units (RSUs) and other components, such as autonomous and connected automobiles. To improve transportation throughput in areas such as safety, congestion avoidance, route planning, etc., RSUs have been used to host many traffic sensors and control systems. No vehicles can vary their motion, so they all travel in the exact position. On the network edge, ICN is used as the fundamental network design. Both motorists and passengers using AV can query the video source to acquire video. Eq. (1) describes all configured network users as a group called users.

$$User - domain = ur_{1,2,\ldots,N} \tag{1}$$

where N represents the total users in a network environment. C is used to represent each node component in the network, including base stations (BS), RSUs, and vehicles, where c represents any node among them. The edge network elements are being considered for caching to enhance the network's capacity for environments with limited resources in terms of decreased load and content availability latency. Centralized, decentralized, and distributed resource management are the three main approaches to edge computing. A cloud server or other central node processes data supplied from all the edge nodes and makes choices in the centralized method. This is the way to go when one thing can consistently gather data about the full system. Decisions may be made by both the central node and other nodes (such as edge nodes) under the decentralized method. Under the distributed model, each of these entities may act autonomously. Keeping tabs on the availability of resources, such as incoming user devices, at various network tiers is essential for resource management. An additional layer of edge nodes between user devices and the cloud may solve this monitoring problem. Adding a new layer, however, increases the monitoring burden significantly; this becomes a considerably bigger problem when federating clusters of edge nodes from various regions to build a global architecture. The components of the network infrastructure are interconnected using a mesh architecture and can connect to as many edge nodes as necessary.

$$a_i = a_i^{1\ldots CH} \tag{2}$$

Users can only receive a portion of the requested multimedia file due to the low transmission range of the edge nodes and the maximum speed of the connected vehicles. Any file a_i chunks with information to a necessary chunk size would be displayed in Eqs. (2) and (3). Therefore, caching the entire data wastes most edge nodes' cache capacity. The edge node can only cache information in chunk format from the whole number of multimedia assets A that are accessible. Assume that all of the chunks are the same sized S. Any file $a_i \in A$ can be split into a higher number of chunks, where CH denotes the total count of file chunks a_i. Eq. (3) represents the demands by users (RQ) for any single chunk at any period p might be displayed since every content request will be provided in chunk format. A text file may store expressions and class definitions according to the chunk file format. To restore the definitions, open the transcript and go to File > File. The client uses a chunk cache to avoid downloading a chunk that has not changed since it was last viewed. The server first asks the client to confirm that he does not require the real chunks by sending a list of hashes rather than the chunks themselves. Chunking is recoding data from smaller, less recognizable units into bigger, more familiar ones. It is often believed that chunking may assist in overcoming the limitations of working memory (WM).

$$RQ\,(x,p) = RQa_{i,k,A}^{1,j,CH}\,(x,p) \tag{3}$$

where the number of requests for a certain piece j of a file a_k at period p on node x. Such that $1 \leq j \leq CH$ $1 \leq a_k \leq A$.

The suggested method allows content caching by all the vehicular network edge nodes. As a result, Eq. (4) gives information about any node's cached chunk (CCH) information that can be expressed as a collection of Booleans. So, the CC information for node x at time p is shown as

$$CCH\,(x,p) = CCHa_{i,k,A}^{1,j,CH}\,(x,p) \tag{4}$$

The represents information j of a_k is buffered in x at period p. The total amount of pieces cached in each cluster cannot exceed that node's maximum storage capacity since each node has little capacity for caching bits of data, which is given in Eq. (5).

$$\mathrm{W} = CCHa_k^j\,(x, p) * S \tag{5}$$

The size of the cache used in node x is represented as CS_x. With a condition of $W \leq CS_x$, Chunk size is represented as S. The total cache in each cluster is represented as W.

3.1 Setting Up a Hierarchical Network

Create a network structure by dividing it into core and edge layers. Numerous network activities, including content streaming, forwarding, demand processing, and information caching, depend on hierarchical network topologies. An effective caching system is one of the key prerequisites for successfully implementing ICN. The core-layer routers do not execute caching to prevent routing interruption. The core layer consists of a 5G core network to decrease network transmission delays and redundant information transmission and increase the effectiveness of streaming content and network computer processing capacity, the deployment of caching and computational capabilities in 5G wireless transmission networks. The edge gateway is the essential component of what is known as "edge" or "fog computing" nowadays. Research shows mobile edge caching is more efficient and can handle jobs faster. Almost every node in the network architecture engages in caching, from end users' devices to the network's foundational base stations. Edge replays are now capable of basic computation and have been empowered. Almost every use case scenario has specific vulnerabilities in various caching techniques. We also have different requirements and standards to fulfil depending on the situation. Additionally, four caching methods have provided fresh insights into our recent efforts and will guide our future investigations. To illustrate, static ones are not useful for models that want to find the popularity hit ratio as they cannot detect changes and variations. Regarding security, things like wired and wireless networks should not be considered interchangeable. We have discussed several points, and we have also outlined some good avenues for further study. As the name suggests, it performs gateway duties, connecting nodes at one end while performing one or more local operations and extending two-way communications to the core and edge layers. Edge cache offered open caching of information material, including web pages, movies, and streamed content.

One technique that helps apps run better and get data and information to consumers faster is edge caching. You may improve your platform's performance by transferring content distribution from the global network to your network's edge, speeding up the process. This eliminates the need to store data in a central data centre; instead, servers located closer to the end users are used to store data. Data transmission across long distances may introduce delay and slow down application performance. Edge caching is used to mitigate this. Some examples of edge servers are content delivery networks (CDN), which allow end users to get material, including web pages, movies, and photos, via a content delivery network (CDN) edge servers. Usually, they are placed in prime global locations that host a disproportionate number of the CDN's users. Edge Cluster Node digital assets transferred across WAN lines or the Internet at the network's edge. ICN content server is specifically used to assess the cache system's performance in a hierarchical network. The 5G communication system is used by gNodeB (gNB) to manage communication systems with the 5G capable device. For quick user responses, all edge layer nodes can engage in caching. Our caching rule says that none of the core network's nodes will cache the material. The choice was made to preserve normal routing operations in the core layer. The edge layer nodes carry out content caching. Edge layer nodes are separated into clusters to deliver content and effectively reduce network traffic strain. The ability to communicate efficiently and reliably across many networks and hosts is why routing tables are so crucial. Typically kept in a router or a networked computer, routing tables provide the destination addresses of data packets traversing an IP network. By directing data flows to optimize available bandwidth without overwhelming the system, routing helps keep networks running smoothly. The primary function of a routing table is to aid routers in making efficient routing choices. The routing table is consulted by routers whenever packets are sent to be forwarded to hosts on other networks. The table contains the IP address of the target device and the optimal path to reach that device.

In other words, the only way to communicate is through the cluster head. An edge layer's size is always considered when deciding how many clusters to use. A straightforward edge layer clustering model is shown in Fig. 2. Multiple groups comprise the edge network, representing an Edge Cluster (EC). By updating the cache position information for its CH, each node part of a particular cluster works together to provide caching. With the help of the load-balancing algorithm, more requests have been answered, and fewer are still waiting to be processed. Since there are more than data sources that the following situations may define, this approach is also crucial for delivering requests.

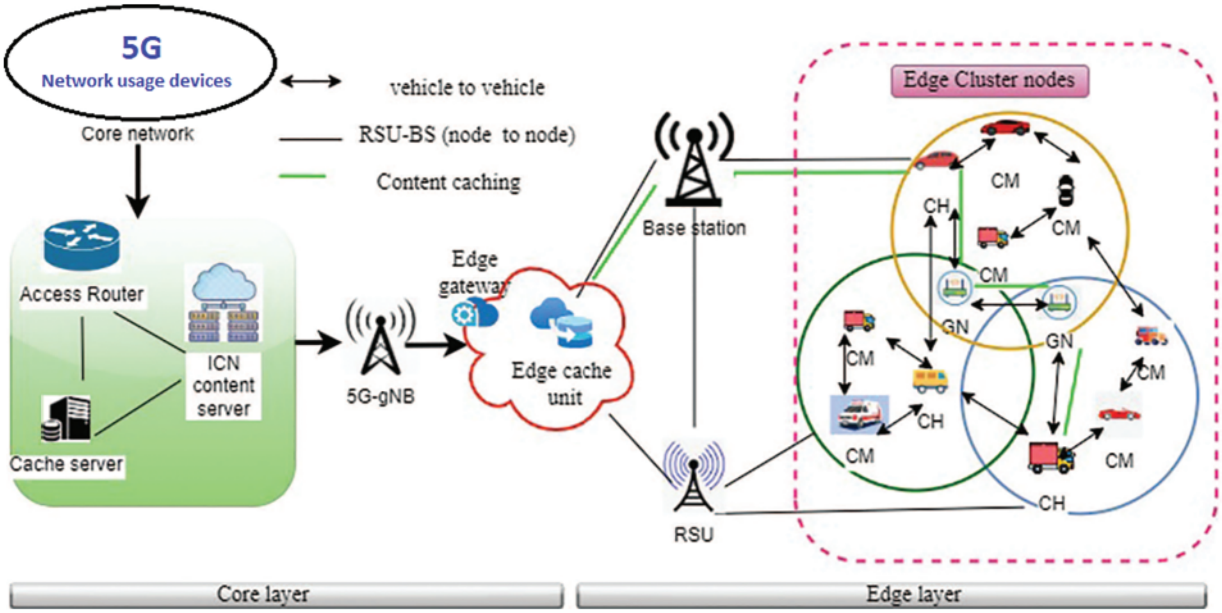

Figure 2: A basic edge layer clustering model in a hierarchical network

Requested nodes (RNs) in CCD may choose between two categories normal and priority, when submitting queries. The first case happens if the RN does not cache the data item locally. To determine whether the necessary data is accessible at nearby nodes, the RN consults its RPT or recent priority table. The information is obtained from the neighboring node, which may be found by consulting the RN's RPT. Priority queries will be sent to the cluster header (CH) if the RN's RPT does not include the cached data item information. The second scenario will occur if the requesting node's local cache and RPT do not contain the data item and its cached information. As a result, the RN gets the necessary data item by consulting its CH. When the data item cannot be located in the RN's local cache, its information cannot be indexed in the RN's RPT, and the query cannot be resolved at the RN's CH, the third scenario occurs. A cluster's cache hit occurs when the CH queries a node with the requested data item within the cluster's range. In the fourth case, the database server immediately responds to the RN when the RN initiates a priority query and the necessary data item is not stored locally or in the cluster. This happens because the RN's CH sends the query to the server. In case five, the RN started a typical query, but the cluster zone did not have the desired data item. When a necessary piece of information is missing at the CH level of a neighbouring CH and all CHs have been visited, the query, including the RN's address, is sent to the database server. Numerous node types, including Gateway Node (GN) and Cluster Member (CM), are further separated among each cluster. Depending on the coverage area of the CH and the cluster size limit, each cluster may contain a maximum of one CH and different GN and CM.

Additionally, it uses GNs to communicate to nodes within the cluster to cluster, and CH in nearby clusters. The GN of a cluster is its border node with a direct connection to a node in a nearby cluster. Transmission of intercluster packets is the primary duty of GN. In telecommunications, a gateway is a network node that bridges the gap between networks that employ distinct transmission protocols. All data must connect with or pass through a gateway before it can be routed, making it an entrance and departure point for the network. Computers that facilitate communication with other nodes in a cluster are known as edge nodes. Gateway nodes and edge communication nodes are alternative names for edge nodes. CMs can also act as access routers (AR), allowing transmission between servers and searchers.

3.2 Initial Parameter Set to Select CH Using IWCA

```
Initial setup = []
least[wt.] =
For every node $c$ in the cluster
Calculate $wt_c$ using IWCA Eq. (6).
```

The serving node CH, has the lowest overall score. IWCA method for our cluster formation and taking into account various elements to predict the cache position as shown in Eq. (6).

$$wt_c = wt_1 N_d + wt_2 [c_c + TT] + wt_3 HO_c + wt_4 c_c + wt_5 pr_c. \tag{6}$$

where weight factors $w1, w2, w3, w4,$ and $w5$ relate to various parameters and always give the sum of 1. The degree of the highest specific node weight is given as N_d. The average transmission time of the next preferred weighted content to a specific node in the further level of cache position is given as $c_c + TT$. If network congestion occurs average waiting of a particular node based on queueing delay is given as HO_c. The CH chosen metric with the cumulative power range for predicting the cache position of a node is given as pr_c. The cache space utilized for a node in a particular time is given as c_c.

The Improved Weight-Based Clustering Algorithm (IWCA) selects the CH method that combines all elements to construct a weight function for identifying the cache position in a communication scenario. Each node's mobility data, such as its direction of travel, road ID, average speed, amount of node connectedness, and mean distance to neighbours, is included in the metrics. A vehicle may identify its neighbours by sending periodic signals, including this mobility data. Before receiving and processing broadcast messages from surrounding vehicles, each vehicle must determine its direction of travel and road ID. As an alternative, criteria like average speed, node connection level, and average distance are to be computed to ascertain if a vehicle is suitable to be a CH. Each measure's real-value weight is assigned to reflect its significance.

```
If $wt_c <$ least[wt.]
update $CH_i = c$
```

else if $wt_c = \text{least}[\text{wt.}]$

$CH_i\, least\, [(cl - id)]$

Otherwise, repeat Eq. (6).

endif.

To calculate the cluster head among the cluster nodes, perform the condition for each c in a cluster and calculate its neighbor node by identifying (CH_i) with the condition $if\ \ c = 0\ \&\ size(C_i) < \beta$ updated the neighbor node with another ordinary cluster node in the cluster as a member and added it at the end of the cluster using the append function $append(c)$. It will make the cluster node with added weight a new member as one more. Make the cluster size a new one and update the count of the old cluster group.

Every CH begins by grouping nodes into clusters after choosing the cluster heads. A node threshold for each cluster to minimize network bandwidth inside the cluster nodes to limit the number of nodes in which each cluster may operate in a cache position. Its primary purpose is to provide setups that fulfil the following requirements: Each node cannot have more than 110 pods. Because it can provide high availability without sacrificing data resilience or redundancy or spending too much on hardware, a two-node software-defined storage system continues to be a popular choice among organizations. High availability, load balancing, and HPC are the three main motivations for server clustering. High availability is where clustering is most often used. Initial n = 0 indicates that node n currently does not exist. After choosing a CH, the above algorithm illustrates the grouping of nodes into clusters.

3.2.1 ICN-Based Cooperative Caching Technique (ICN-CoC)

The CH is in charge of the information stored in CM under one of the cooperative caching schemes that are being suggested. The cluster's members cache the data without duplicates. No two CMs are permitted to distribute data directly within the cluster, and the only means of communication within the cluster is via CH. Because a cluster of powerful nodes with sufficient cache storage can handle additional requests, this is the cause. Using this interconnection, CMs can respond to demands more routinely, perhaps immediately to their own CHs or partially through neighboring CHs. In general, there are two types of content management systems: enterprise and web content management systems. Web content management systems (WCM) are a subset of content management systems (CMSs). The content of websites may be easily created and managed using this system. In a nutshell, it makes working with digital media easier. Without knowing HTML, CSS, or any other coding language, a content management system (CMS) allows users to add, edit, and remove material from a website. Anytime a content chunk request is initiated, the CM first checks its CH for the requested content before caching it. Only a certain number of hops, as decided by HO_{value}, are permitted for each C_{ur}.

The content format id: MD shows the cluster communication mechanism for content identity(id) of each cluster with metadata information search in detail is encouraged to store the associated CM id and the metadata of such cached data from their CMs to address the cluster members. For each content chunk, CCH with the corresponding user with identity is registered in a coverage area of RSU. The provider configures each message's application zone or the time frame it is sent and received. It is instantly rejected by the recipient if the application message is delivered beyond the zone designated by the message source. Following is a method for calculating the position tolerance of an application message, which, combined with a verification mechanism, allows an OBU to ascertain whether or not a message is valid for the present location. Problems with data transmission fading and propagation are disregarded. The outgoing vehicle's onboard unit (OBU) may relay authenticated communications to cars on the road that are outside the communication range of the RSU if the safety application zone of a VANET does not have full network coverage via RSUs. For an OBU that is not inside an RSU's communication range, a "message-forwarder" OBU may relay the broadcast to them. Vehicles that accept signatures check their contents. A simple approach would be to return the same signature documents obtained from the RSU. The receiving OBU checks the signature as though it had already received the message from the matching RSU. However, because the forwarding OBU has already received and verified the message, repeating the complete verification procedure for each receiving OBU would be inefficient, especially when the traffic density is high. Furthermore, in a traffic dispute, it would be impossible to determine whether a message originated from an RSU or an intermediary OBU.

The node follows the CH instruction with the necessary update to the route for content requests and is stored in the corresponding cache position. Based on the probability matrix, the attractiveness of the content is shown in the user preference rating. The HO_{value} is reduced by 1 with each change of C_{ur}. Only when HO_{value} is greater than 0 content not in CH can be located from a neighboring CH'. By achieving data diversity, it offers adequate use of the limited disk space. Data splitting also promotes increased use of cache position space among CMs, eventually lowering network traffic and search expenses. One half of a two-part split is usually used for data evaluation or testing, while the other half is utilized for training the model. Data splitting is crucial in data science, especially when building models from data. The hold-out method is the quickest and easiest method for dividing the dataset into training and test sets. After training the model on the training set, it must be tested on "new" data (the test set) to see how well it can predict outcomes from data that are not part of it. The train-test split is one way to measure how well a machine-learning algorithm performs. It is compatible with all supervised learning algorithms and may be used for regression or classification issues. The time during which a particular node serves as the cluster head is typically a metric to estimate battery drain. The average moving velocity of a particular cluster to estimate the mobility percentage.

3.2.2 Workflow-for ICN-Based Cooperatively Cached Content Scheme

For every C_{ur} assign a new cluster member

Check *content in cache* then forward C_{ur} to corresponding CH

Update distance between CMs

Follow MD (C_{ur}) in cache-chunk (CH) and retrieve CM

Assign C_{ur} to *new CH* update cache chunk

Stop the cluster formation and update the corresponding user information

However, CH's data caching is difficult due to a lack of cache space for every organization member. To effectively manage storage capacity, the suggested caching system always caches the contents in the portions through each CS of both the CMs. Data consistency is guaranteed by distributed caching systems via many means. They start using a "write-through caching" method, in which writes are executed in the cache before being propagated to the underlying data store. The storage and the cache will always be up-to-date if this is done. Data and other information may be temporarily stored in a computer system's cache using software or hardware. It is used as a cheaper alternative to faster memory to improve the speed of data accessed regularly or often. CHs are instructed to save the CM id and metadata for the information stored by their CMs. If a CM cannot find the information needed within its internal cache, it uses metadata to perform the data search process. When searching for the necessary metadata inside CH, if the findings are unsuccessful, CH moves on to the neighboring CH and repeats the process.

3.2.3 Prediction of Content Rate Using CoC-NMF

The most common way consumers of almost any multimedia data provide comments is by rating them. The rating essentially represents the user's opinion of the accessed content. Eq. (7) can easily determine the rating for any content a user views, yet no ratings have been supplied.

$$C_{ur} = l_{ac}/C_{len} \tag{7}$$

The amount of time that the content was accessed is given as l_{ac}. The true length is represented as C_{len}. The higher the value of C_{ur}, the more interested the user is in the material. Users' future needs can be forecast by considering the historical attractiveness of videos and users' preferences, namely, the expected ratings, using the NMF method to anticipate the consumer's future ratings. The suggested caching strategy can cache media in the video layer in the core nodes in the network based on anticipated demand. Several key differences exist between online content caching and video streaming caching. Unlike conventional online objects like text and pictures, video objects are somewhat bigger. The second point is that online material has not followed the same pattern of popularity as streaming videos. Having only the beginning of a movie also enables a video app to start playing while the download is ongoing, which is not the case with web objects. The result was the widespread use of segment-based caching for multimedia streams, in which individual segments were created from data blocks belonging to video objects. For instance, to reduce start-up delays, these parts might be of varying sizes, for example, a shorter section at the beginning of the film. For such segmented videos, proposed cache admission and replacement policies consider the video's beginning and segment, the time when cached segments will be reused, and the eviction strategy's ability to predict the segment request that will happen farthest in the future. Assuming a single cache and never interrupting sessions, the authors fail to account for delivery durations across network nodes. We include session announcement data in our suggested method, which also takes advantage of the temporal structure of segmented video. Also, we use a cache structure that takes delivery delays between nodes into consideration and interruption models to mimic user churn. The CoC-NMF approach simplifies the forecasting of future ratings for any inaccessible resource. This cutting-edge machine learning method, which has also been employed in recommendation systems for rate forecasting, does not offer unfavorable predictions. A common reason for dividing accessible data into two halves is cross-validation; this process is known as data splitting. A prediction model is created using one subset of the data. Whereas the other is used to assess how well the model performs. Testing the model on new data aids in making an informed model selection and keeps overfitting at bay. Finally, data splitting is an important tool for accurately categorizing unbalanced datasets. Data segregation aims to allow for the application of separate access controls to distinct data sets by physically isolating them. The objective is to restrict access to certain data sets so only authorized users can see them. Each row of the rating matrix $RATE$ cell $ur(ur_n, a_i)$ represents how any user ur_n rated the content file a_i. The method takes into account both users and information when calculating ratings. The results of multiplying the consumer form $X(N \times Ft)$ evaluate the correlation relationship between the consumer, its capabilities, and the information matrix $Y(A \times Ft)$ represents a relationship between the content and features, and produce the results of a rating matrix $RATE(N \times A)$. Numerous features could influence how a person rates something represented as Ft. The characteristics of the user determine the user's interest rate in a certain piece of material depending on the characteristics such as films, journalism, games, etc. From Eq. (8), users can find the anticipated rating matrix $RATE$, which approximates the real matrix $RATE$, by using the equation

$$REAL\,(RATE) = usr - preference \times file\ chunk \tag{8}$$

Any file's single value rating from any user ur_n is calculated using Eq. (9).

$$x_n \times y^t_a = y_{n,ft} \times y_{ft,a} \tag{9}$$

where y_{ft} is the fth column vector of matrix X and x_n is the nth row vector of X and Y matrices, respectively.

3.2.4 Determining the Attractiveness of the Content

The content attractiveness is based on the information representing the second most crucial factor after rating in determining "what" to display among the chunks of information. Any further requests for content that is already available cause the previous request quantity to be updated to +0.01. The content attractiveness ranges from fresh content, access rate, and hop countcan be separated into different attractiveness classes given by CLS, assuming pc_0 and pc_1 The smallest and largest attractiveness counts are given in Eq. (10), respectively.

$$Edge\,(pc) = (fresh\ content + access\ rate + distance\ factor) * CLS \tag{10}$$

Here, the attractiveness of the content related to the rating is given in the above Eq. (10) as $Edge\,(pc)$. The core network should cache less attractive material, whereas the edge should store engaging content. Depending on the attractiveness and rating of the content, CMs are indeed the positions where real caching occurs. Cached data is stored in a high-speed data storage layer that allows subsequent requests for that data to be processed more quickly than if stored in its main location. This data is usually temporary. Memory temporarily stored by the computer's central processing unit (CPU) to facilitate faster processing of frequently used commands and data is known as cache memory. A computer's cache functions as an extension of its main memory, enhancing its performance. Software that facilitates some users' creation, editing, and publishing of material is known as a content management system (CMS). To speed up access times, files are often duplicated and stored in a temporary cache. This technique is known as caching. Although the phrase is often used in Internet technology, a cache may also apply to any temporary storage area for copies of files or data. The processor chip is the permanent home of a main cache. This little cache performs well with an access time similar to CPU registers. Matrix A is started. Matrix A, which consists of $RATE$ as rows and CLS as columns, describes the likelihood that any content is cached to correlate anticipated content rating and attributes. An element p_{yz} in matrix P where $y \in [1, RATE]$, $z \in [1, CLS]$ indicates a piece of content's likelihood of being cached depending on its score and the attractiveness category within which it depends is given in Eq. (11).

$$P_{RATE,CLS} = \begin{pmatrix} p_{q1,1}p_{q1,2}\cdots p_{q1,CLS} \\ p_{r2,1}p_{r2,2}\cdots\cdot\cdot p_{r2,CLS} \\ p_{R,1}p_{R,2}\cdots p_{R,CLS} \end{pmatrix} \tag{11}$$

There are representations $\theta_1 + \theta_2 = 1$ represented for the probability matrix P. The weight related to the rating and attractiveness class is the coefficient utilized in Eq. (12).

$$p_{yz} = \theta_1 \frac{1}{y} + \theta_2 \frac{1}{z} \tag{12}$$

Depending on how popular a piece of information is, it belongs to one of several classes. The CM uses combined attractiveness-rate prediction models to store the most frequently requested data, which content then demands direct access from individual factors, as follows via CHs, which has allowed for substantial advancements. Before putting money into the market, research its opportunities. Examining the market's attractiveness will help you characterize the many profitable prospects in a highly competitive sector. Facial signals connected to fitness have the same impact on the way people perceive them as other types of beauty. This similarity lends credence to the idea that our perceptions of physical attractiveness and unattractiveness are based on our responses to abnormal facial signals that indicate our health and fitness levels.

3.3 Evaluation Metrics

3.3.1 Cache Hit Ratio (CHR) for Effective Content Search

CHR is one of the most crucial metrics to consider when assessing the effectiveness of any caching solution. CHR is a metric for how many requests a caching node has fulfilled. Increased hit proportion in any cache management strategy always denotes effective content searches and impacts existing caching strategies. A popular statistic for assessing the effectiveness of a caching selection strategy is the hit proportion, which quantifies the proportion of demands that the enabled cache node can fulfil. The hit rate is often expressed as a percentage and is determined by dividing the total number of cache hits by the total number of memory requests made to the cache during a certain time frame. Analogously, the miss rate is calculated by dividing the total number of cache misses by the total number of memory requests made to the cache. The hit ratio may be determined by dividing the total number of cache hits by the sum of cache misses. To illustrate the point, divide 51 by 54 in the case of 51 cache hits and 3 misses over time. With this setup, the hit ratio would be 0.944. You may get the hit rate for the whole sales team, for each sales area, salesperson, or product category. It may be used to compare the sales team's performance across various periods and to see how the company stacks up against competitors in the same industry. The cache volume of edge nodes is defined as the maximum quantity of chunks that may be stored there, combining BSs and RSUs. When an intra-cluster node grants a request using its CH, a cache hit is experienced for the cluster-based cache management technique given in Eq. (13).

$$CHR\left(cl\right) = \frac{\sum_{u=1}^{cl} cachehit_u}{cl} \tag{13}$$

Here, the number of hits at node u in a cluster level is given by $cachehit_u$. A cluster level at nodes is given by cl. The component framework connection hit ratio is calculated. If the requested media are still not cached at the CH core network, it is necessary to fetch the requested videos from the content provider, and the CHR can indeed be enhanced no matter how big the edge node cache gets. With edge caching, data is stored as near to the user as feasible by combining classic caching techniques with edge computing infrastructure. By transferring data from a CDN that is geographically closer to the user, this method of content delivery decreases network latency. The cache hit ratio (CHR) is an essential measure for any system that uses a cache. At times, it stands alone as a measure. A metric for this is the number of times the cache is checked for potential content. The ratio of cache hits to total cache checks is determined by dividing the former by the latter. The hit ratio of a cache is defined as the percentage of content requests that are successfully filled relative to the total number of requests. Content delivery networks (CDNs) provide one kind of cache, and a high cache hit ratio indicates a well-performing CDN. The hit ratio is determined by calculating the number of cache hits and comparing them to the total number of content requests. Another way to look at it is as a mass ratio, which compares the total number of content requests with the cache misses. The amount of network infrastructure cache is part of the total video capacity that can be cached there.

3.3.2 Average Hop Minimization Ratio (AHMR) for Effective Content Caching

Based on the number of hops, a metric is used to gauge how effectively a caching strategy works. AHMR calculates the different hop counts necessary when a result is obtained from the origin in response to a query and the hop count saved when the node satisfies the demand for any caching strategy. The suggested method decreases the hops requested while boosting cache size to satisfy user content requests. Since actual providers deliver most of the content, comparison systems' AHMR performance is poor because responses to content requests must go further. For a network, AHMR is calculated using Eq. (14).

$$AHMR = 1 - \frac{HO(RQ, UR)}{|HO(RQ)|} \tag{14}$$

where RQ represents the average demand made by the consumer, and N is the total count of consumers. The quantity between node n and the node in hops satisfying RQ is given as HO, and the quantity of hopping between node n to the sender OS is given as HO_{nOS}. The faster content transmission with less network congestion is indicated by the higher AHMR. The underlying reality is that the content demanded may be present locally or among the member nodes.

3.3.3 Mean Access Latency (MAL)

Calculate the time taken to fulfil a content demand. Refers to the duration in units that a user must endure while waiting to respond to a content demand. Circumstances like heavy network traffic and a lack of diversity in material might impact it. The number of chunks depends on the edge unit that must determine if the requesting chunks are locally cached when the consumer resumes the stream. If so, it can immediately provide the user with the desired fragments. When a user changes to a different video, the edge node receives an interest packet for that new movie. The edge nodes will similarly determine whether there are locally cached copies of newly demanded multimedia chunks. To "chunk" anything implies dividing it into smaller, more manageable pieces that are easier to digest and retain. Web interfaces function better in every way when chunking is done wisely. Numerous multimedia file formats use chunks, which are bits of data. These formats include PNG, IFF, MP3, and AVI. Each chunk's header specifies several parameters, such as the chunk type, comments, size, and so on. A text file may store expressions and class definitions according to the chunk file format. The chunking process divides lengthy data strings into smaller, more manageable pieces. Compared to a lengthier, unbroken data string, the produced pieces are simpler to commit to working memory. It seems like chunking works with everything from text and audio to images and movies. Video content must also be continuously cached at the edge devices to minimize the fetching latency and the back-haul traffic because AV consumers move quickly and can only access a few portions out from edge nodes due to the short range of BSs/RSUs, edge devices cache video at the fragment level instead of the whole video level for core network nodes. As a result, storing complete films on the network edge will squander their cache space and reduce caching effectiveness. The MAL is calculated using the method given in Eq. (15). It can be utilized as a single statistic to determine the effectiveness of a caching scheme by roughly assuming that traffic and heterogeneity are equal.

$$MAL = \sum_{u=1}^{CH_{tot}} IPL_u + \sum_{u=1}^{CH_{tot}} DPL_u \tag{15}$$

where IPL stands for inter-segment packet latency for real content chunks CH, while DPL stands for the latency a data packet experiences as it travels from the content provider or cacheable node to the content requester.

4 Experimental Results

The observed findings demonstrate improved use of the cache, rapid data retrieval, and content cache dissemination. The following fields are found in all datasets: Start time: the number of seconds since the request was made. The request's end time is measured in seconds. Period: beginning and ending time contents: The quantity of packet content needed to fulfil this request is referred to in the data set. Rate is the number of packets divided by time in seconds. Meters: Sender stopping distance meters are the receiver stopping distance. Actual distance: the distance between the transmitter and the receiver in meters. Our random network has 100 nodes (vehicles), 30% as gateway nodes, 20% as core layer nodes, and the remaining as cluster nodes. The nodes that make up the core layer are those with a bandwidth of more than 100 Gbps. Each cluster can only have members within a 50-meter radius or roughly half the nodes' transmission range. As a result, vehicles in nearby clusters and local ones can communicate with each other. Only multimedia streaming services are available to vehicles. Media streaming refers to continuously sending video and audio data via an internet connection, whether direct or wireless. Streaming media is any material that can be sent over the Internet and played back in real-time on devices like computers and mobile phones, whether live or recorded. The three main types of protocols relevant to multimedia streaming are the transport protocol, the network-layer protocol, and the session control protocol. Basic network services like address resolution and network addressing are provided by the network-layer protocol, which is supported by IP. The cluster size is 4 (node count), and the vehicle's maximum speed is 50 km/h. The transmission range of the node is taken as 100 m. The number of contents depends on the server at a maximum of 10,000. The influence of cache size with varying lengths of chunks in terms of percentage is illustrated in Fig. 3 with multimedia attributes taken from. The time used for simulation is 120 s, and the information size is 0.2 to 3 GB. A default simulation tool SUMO represented highway vehicle scenarios for the implementation of the ICN-based cache scheme. This graph result discusses the impact on cache size using CHR.

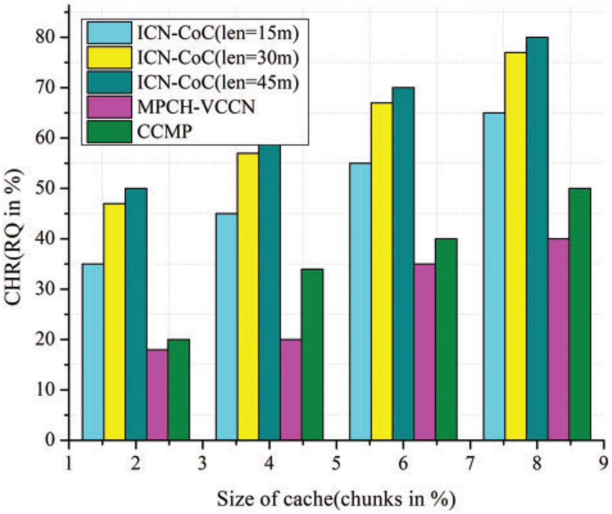

Figure 3: Impact of CHR on the size of the cache

Fig. 3 illustrates that different cache sizes influence the CHR compared to existing caching techniques; the hit ratio of the suggested scheme is always higher for whatever cache size. When the cluster length is 45 m, the highest growth rate of the suggested technique for cache hit is reported to be 84% for queue length holding 8% of all chunks. Additionally, because of the cooperative nature of the suggested technique, requests that a cluster cannot fulfil on its own may be fulfilled by neighboring clusters, increasing the CHR. The higher CHR indicates unmistakably effective caching with less network server demand. As shown, all methods can increase the CHR with an increase in the capacity of the core network cache. Interestingly, ICN-CoC techniques greatly improve the hit ratio in all circumstances with an increase in the size of the edge node cache, since larger edge node cache sizes allow edge nodes to store more chunks, thereby increasing the probability of a CHR.

Fig. 4 shows the impact of the hop count with varying cache sizes. The ICN-CoC technique significantly reduces the number of accessible hops while increasing cache size compared to the other two techniques, CCMP and MPCH-VCCN. To fulfil user content requests. The underlying reality is that the requested content may be locally or among cluster members. Since real providers provide most of the content, comparison schemes' AHMR performance is poor since the response to content requests must travel through more hops. Faster content delivery and lower network overhead are indicators of improved AHMR. An interconnected system of servers close to end consumers is known as a content delivery network (CDN). Media files like photos, videos, stylesheets, JavaScript scripts, and HTML pages may be quickly sent via a content delivery network (CDN). Page loads, transactions, and the overall online experience are all improved when websites and apps use a content delivery network (CDN). Because CDN technology operates invisibly behind the scenes, users may not notice they connect via one while reaping its advantages. Reducing the distance between the server and the user is the primary goal of a content delivery network (CDN), which may enhance the performance and speed of a website or application. In contrast, "cloud services" include a wide variety of offerings that are not physically housed on any one server but are made available over the Internet.

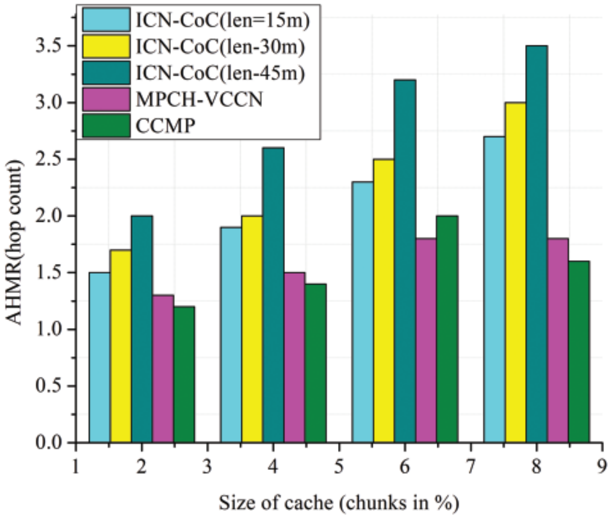

Figure 4: Comparison of algorithms that AHMR measure for different cache sizes

The MAL data in Fig. 5 shows that for each investigated method, the MAL value decreases as the buffer size grows in Fig. 5. Because of the node's huge cache capacity, the required information is mostly close to the requester. The IWCA method-based clustering strategy was utilized to choose the cluster head, which in turn chose the cache position. The average access delay gauges the requester's latency experience while retrieving the requested content. An object's average arrival rate to the access link (B) and the average time it takes to transmit an object over the link (A) is used to calculate the average access delay (A/(1-AB)). The HTTP request messages are probably so little that they do not cause any congestion on the network or the access connection. The term "latency" describes the time data packets get from one place to another. Delays in data transmission caused by high latency might impact the other two measures. Increased jitter and unpredictable arrival intervals are symptoms of a network with high latency. Delay is averaged across all cycles in the current buffer after each cycle is measured on a rising or falling pair of edges. The Measurements lozenge displays global statistics computed across all collected buffers, much as other measurements. The bottleneck is caused by the increase in the demands that are processed at any given moment. As a result, many requests will either not be processed or be processed by cachable nodes very slowly. The MAL that this entire phenomenon will greatly impact content requesters' experience. ICN-CoC with cluster members based on fragments of multimedia content taken for caching gives better results than the other two techniques.

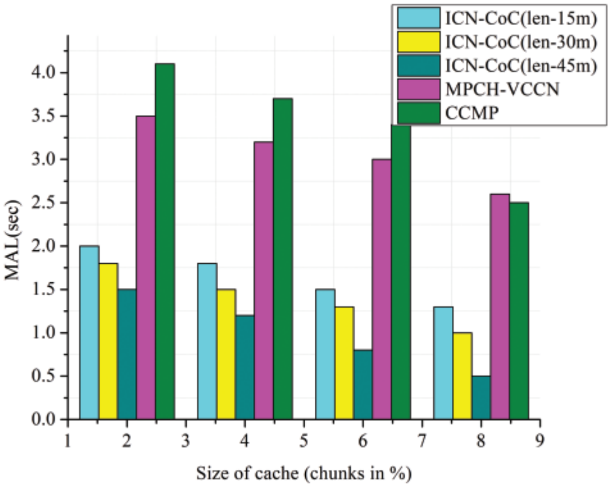

Figure 5: Impact of MAL on different cache sizes

In particular, the proposed method (len = 45 m) showed the greatest decrease in MAL for a cache size of 8% with a value of 0.4 s, representing improvements in CCMP and MPCH-VCCN.

Fig. 6 illustrates the simulation results for all metrics with an arrival rate of 20, 40, 60, 80, and 100 requests per second. Edge nodes' cache sizes are set to 100, and the core network's overall cache size is set to 10% due to the limited storage that includes edge units and core network access points. The results are displayed in Fig. 6 after the effect of the inter-arrival time on the CHR has been assessed. The user's transportation prediction module can predict the arrival time using the suggested path, speed, and new position of cached data. Predicting future traffic conditions, including the number of cars and travel time, in a given region or along a certain route is known as traffic prediction. Reducing traffic congestion and optimizing transportation networks are two significant goals of this endeavour. Historical and real-time traffic data gathered from various sensor sources—inductive loops, radars, cameras, mobile GPS (Global Position System), crowdsourcing, social media, etc.,—is crucial for traffic flow prediction. An often-used statistical method for making predictions about future behaviour is predictive modelling. One kind of data-mining technology, predictive modelling solutions, compiles past and present data to foretell future results.

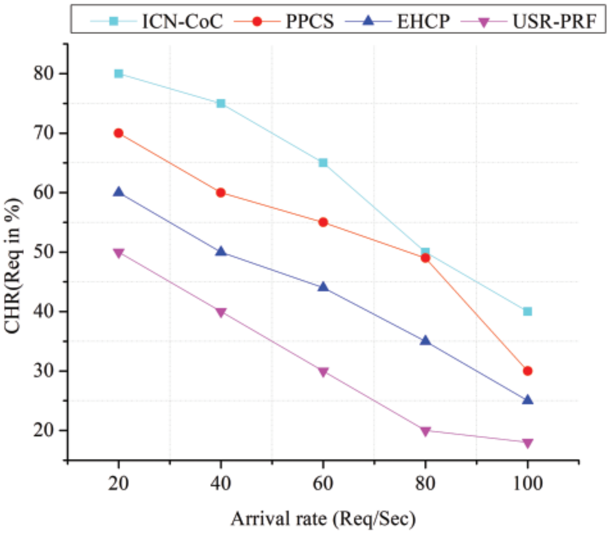

Figure 6: Impact of arrival rate on CHR

According to the initial observation, the CHR declines as the arrival time for approaches increases. Approaches that use caching other than CoC policy for edge nodes have a substantially smaller influence on arrival rate. According to the initial observation, the CHR declines as arrival rates on the approach rise. The rationale is that, because edge nodes have a small latency, a higher arrival rate will result in more requests from AV users, increasing unfulfilled requests. Edge computing improves online apps and Internet gadgets by moving processing closer to the data source. As a result, less bandwidth and delay are required for connections between the client and server. The choices made by a company on the manufacturing and distribution of its products constitute its operations strategy. An organization's operations strategy encompasses all choices about the many activities that go into making or delivering a product.

For approaches, the arrival rate provides far less effect based on the cache position in a network. The more favorable CHR in the high request percentage environment demonstrates the effectiveness of our suggested caching strategy in handling challenging network conditions with other algorithms such as PPCS, EHCP, and USR-PRF. For all the approaches tested, the hit ratio keeps dropping as the rate of arrival requests rises. The justification for this is that as the number of queries from content applicants rises, fewer will be satisfied since edge nodes' limited storage capacity will also rise. Despite this, the arrival rate has little impact on the existing literature schemes compared above.

From Fig. 7, the cluster members' use of combined attractiveness rating models to store the most frequently requested data, which content requesters then directly access from either factor, perhaps through cluster heads, has allowed for substantial advancements. As the vehicle population increases, so do improvements in the hit proportion of the ICN-CoC caching scheme. The mean latency keeps rising, in addition, to cache hit for all strategies, as the number of vehicles increases. Due to the extremely congested system, most demands are fulfilled by the nearest neighbors within the communication range of the content applicants, minimizing the number of hops traversed. The decreased average of travelled hops demonstrates that our suggested strategy provides higher cache efficiency by significantly decreasing network overhead. On the other hand, it is simple to observe how the expansion in vehicle numbers has a detrimental impact on accessibility latencies. Larger vehicles generate more demand. In a heavily populated area, there will be a delay in fulfilling these requests for the remaining algorithms, DL-IC, MPCH-VCCN, and VCCN.

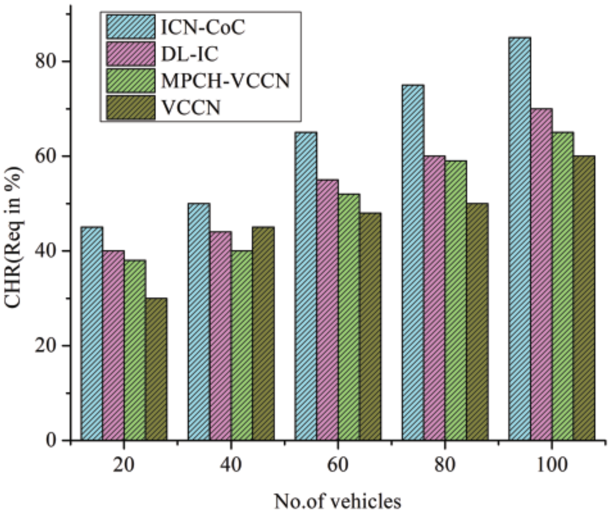

Figure 7: Impact of vehicle count on roads with CHR

5 Conclusions

Content delivery delay due to limited cache storage is a crucial issue in 5G networks and presents difficulties for the current network design. To overcome this, we analyzed the ICN-CoC plan in large-scale topology and contrasted it with other plans. Making use of a two-layer hierarchical network. The ICN-CoC approach considered caching content and location for network efficiency. The cluster methodology gives the modified IWCA utilized to choose the CH, which ultimately chooses the cache position. The value of the probability matrix, which considers the attractiveness and function of the content, was used to determine the content that should be cached. The advantage of the ICN-CoC method over various benchmark schemes is that it provides the highest CHR because it caches more content and content close to the requester; improving AHMR and MAL is shown in simulation findings, and discussions on results supported the effectiveness of the ICN-CoC approach. The ICN-CoC method’s limitation is that it does not consider the variable levels of user affinity for various content categories. As part of the future stage of the work, deep learning approaches are applied to forecast content attractiveness while accounting for user/content mobility in ICN.

Acknowledgement: The authors thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the New Brunswick Innovation Foundation (NBIF) for the financial support of the global project. These granting agencies did not contribute to the design of the study and the collection, analysis and interpretation of data.

Funding Statement: New Brunswick Innovation Foundation (NBIF) for the financial support of the global project.

Author Contributions: Conceptualization, Osamah Ibrahim Khalaf, Sameer Algburi; Data curation, R. Mahaveerakannan, Habib Hamam; Formal analysis, Thanarajan Tamilvizhi, Sonia Jenifer Rayen; Investigation, Habib Hamam; Project administration, Osamah Ibrahim Khalaf; Resources, Sameer

Algburi, Sonia Jenifer Rayen; Software, Sameer Algburi; Validation, R. Mahaveerakannan, Sonia Jenifer Rayen; Visualization, Thanarajan Tamilvizhi, Osamah Ibrahim Khalaf; Writing Thanarajan Tamilvizhi, Osamah Ibrahim Khalaf; Draft, Sameer Algburi, Sonia Jenifer Rayen. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Dataset hosted on Dataset: https://github.com/IhabMoha/datasets-for-VANET (accessed on 27 May 2024).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Boubakr, K. Sharif, F. Li, H. Moungla, and Y. Liu, "A unified hybrid information-centric naming scheme for IoT applications," *Comput. Commun.*, vol. 150, no. 1, pp. 103–114, 2020. doi: 10.1016/j.comcom.2019.11.020.

[2] R. Wajid, A. S. Hafid, and S. Cherkaoui, "SoftCaching: A framework for caching node selection and routing in software-defined information centric Internet of Things," *Comput. Netw.*, vol. 235, no. 109966, pp. 1–12, 2023.

[3] Y. Ying, Z. Zhou, and Q. Zhang, "Blockchain-based collaborative caching mechanism for information center IoT," *J. ICT Stand.*, vol. 11, no. 1, pp. 67–96, 2023. doi: 10.13052/jicts2245-800X.1114.

[4] W. U. Rehman, M. M. Al-Ezzi Sufyan, T. Salam, A. Al-Salehi, Q. E. Ali and A. H. Malik, "Cooperative distributed uplink cache over B5G small cell networks," *PLoS One*, vol. 19, no. 4, pp. 1–20, 2024. doi: 10.1371/journal.pone.0299690.

[5] S. Fayyaz, M. Atif Ur Rehman, M. Salah Ud Din, M. I. Biswas, A. K. Bashir and B. S. Kim, "Information-centric mobile networks: A survey, discussion, and future research directions," *IEEE Access*, vol. 11, no. 1, pp. 40328–40372, 2023. doi: 10.1109/ACCESS.2023.3268775.

[6] Y. Alotaibi, B. Rajasekar, R. Jayalakshmi, and S. Rajendran, "Falcon optimization algorithm-based energy efficient communication protocol for cluster-based vehicular networks," *Comput. Mater. Contin.*, vol. 78, no. 3, pp. 4243– 4262, 2024. doi: 10.32604/cmc.2024.047608.

[7] Z. Zhang, C. H. Lung, M. St-Hilaire, and I. Lambadaris, "Smart proactive caching: Empower the video delivery for autonomous vehicles in ICN-based networks," *IEEE Trans. Vehicular Technol.*, vol. 69, no. 7, pp. 7955–7965, 2020. doi: 10.1109/TVT.2020.2994181.

[8] D. Sharma *et al.*, "Entity-aware data management on mobile devices: Utilizing edge computing and centric information networking in the context of 5G and IoT," *Mobile Netw. Appl.*, pp. 1–12, 2023. doi: 10.1007/s11036-023-02224-5.

[9] O. I. Khalaf, K. A. Ogudo, and M. Singh, "A fuzzy-based optimization technique for the energy and spectrum efficiencies trade-off in cognitive radio-enabled 5G network," *Symmetry*, vol. 13, no. 1, pp. 1–47, 2020. doi: 10.3390/sym13010047.

[10] S. M. A. Kazmi *et al.*, "Infotainment enabled smart cars: A joint communication, caching, and computation approach," *IEEE Trans. Vehicular Technol.*, vol. 68, no. 9, pp. 8408–8420, 2019. doi: 10.1109/TVT.2019.2930601.

[11] D. T. Hoang, D. Niyato, D. N. Nguyen, E. Dutkiewicz, P. Wang and Z. Han, "A dynamic edge caching framework for mobile 5G networks," *IEEE Wirel. Commun.*, vol. 25, no. 5, pp. 95–103, 2018. doi: 10.1109/MWC.2018.1700360.

[12] M. Dehghan *et al.*, "On the complexity of optimal request routing and content caching in heterogeneous cache network," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1635–1648, 2017. doi: 10.1109/TNET.2016.2636843.

[13] P. Gopika, M. D. Francesco, and T. Taleb, "Edge computing for the Internet of Things: A case study," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1275–1284, 2018. doi: 10.1109/JIOT.2018.2805263.

[14] K. Kottursamy, A. U. R. Khan, B. Sadayappillai, and G. Raja, “Optimized D-RAN aware data retrieval for 5G information centric networks,” *Wirel. Pers. Commun.*, vol. 124, no. 2, pp. 1011–1032, 2022. doi: 10.1007/s11277-021-09392-1.

[15] M. Jeyabharathi, K. Jayanthi, D. Kumutha, and R. Surendran, “A compact meander infused (CMI) MIMO antenna for 5G wireless communication,” in *Proc. 4th Int. Conf. Invent. Res. Comput. Appl. (ICIRCA)*, Coimbatore, India, 2022, pp. 281–287.

[16] H. Li, O. Kaoru, and D. Mianxiong, “ECCN: Orchestration of edge-centric computing and content-centric networking in the 5G radio access network,” *IEEE Wirel. Commun.*, vol. 25, no. 3, pp. 88–93, 2018. doi: 10.1109/MWC.2018.1700315.

[17] N. Anselme, N. H. Tran, K. T. Kim, and C. S. Hong, “Deep learning-based caching for self-driving cars in multi-access edge computing,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 2862–2877, 2020.

[18] S. Siyang, C. Feng, T. Zhang, and Y. Liu, “A user interest preference based on-path caching strategy in named data networking,” in *2017 IEEE/CIC Int. Conf. Commun. China (ICCC)*, Qingdao, China, IEEE, 2017, pp. 1–6.

[19] L. Yao, C. Ailun, J. Deng, J. Wang, and G. Wu, “A cooperative caching scheme based on mobility prediction in vehicular content centric networks,” *IEEE Trans. Vehicular Technol.*, vol. 67, no. 6, pp. 5435–5444, 2017. doi: 10.1109/TVT.2017.2784562.

[20] M. Yahui, M. A. Naeem, R. Ali, and B. Kim, “EHCP: An efficient hybrid content placement strategy in named data network caching,” *IEEE Access*, vol. 7, no. 1, pp. 155601–155611, 2019. doi: 10.1109/ACCESS.2019.2946184.

[21] Q. N. Nguyen *et al.*, “PPCS: A progressive attractiveness-aware caching scheme for edge-based cache redundancy avoidance in information-centric networks,” *Sensors*, vol. 19, no. 3, pp. 1–12, 2019. doi: 10.3390/s19030694.

[22] Y. Lin, X. Xu, J. Deng, G. Wu, and Z. Li, “A cooperative caching scheme for VCCN with mobility prediction and consistent hashing,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20230–20242, 2022. doi: 10.1109/TITS.2022.3171071.

[23] A. A. El-Saleh *et al.*, “Measuring and assessing performance of mobile broadband networks and future 5G trends,” *Sustainability*, vol. 14, no. 2, pp. 1–20, 2022. doi: 10.3390/su14020829.

[24] M. V. S. S. Nagendranth, M. R. Khanna, N. Krishnaraj, M. Y. Sikkandar, M. A. Aboamer and R. Surendran, “Type II fuzzy-based clustering with improved ant colony optimization-based routing (T2FCATR) protocol for secured data transmission in manet,” *J. Supercomput.*, vol. 78, no. 7, pp. 9102–9120, 2022. doi: 10.1007/s11227-021-04262-w.