IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15824

Reducing Round-Trip Time in Cloud Environments Using Software-Defined Networking with Intelligent Prefetching and Caching: A Review

Chandu H M[1], Nagendra Swamy M S[2], Chandan Hegde[3]

Department of MCA, Surana College(Autonomous), Bengaluru[1]

Department of MCA, Surana College(Autonomous), Bengaluru[2]

Assistant Professor, Department of MCA, Surana College(Autonomous), Bengaluru[3]

Abstract: Round-Trip Time (RTT) is considered the main metric for measuring the performance of applications for end users in today's cloud environments. In order to improve the user experience, RTT has to be reduced by future network architectures. Current architectures are mainly designed to be reactive, meaning that they counteract congestion, user mobility and changes in service demand and resource requirements only after the quality of service that is being provided to users has already been affected. In this paper, we review works that have contributed in this regard and also present a completely integrated proactive network architecture design based on Software-Defined Networking (SDN), machine-learning-based prefetching and an edge cache manager. The architecture is especially suitable for scenarios with high user mobility as well as with high variability of access traffic. In our work, we first present notable technology progress supporting reduced round-trip times followed by a detailed description of the novel architecture. The work also describes a simulation-based approach for evaluating the performance of the architecture. In the end we outline the expected results as well as the challenges that have to be faced during the implementation of the architecture and how these challenges affect future cloud as well as edge networks.

Keywords: Software-Defined Networking (SDN); Predictive Prefetching; Edge-Centric Caching; Round-Trip Time (RTT); Cloud Computing; Quality of Service (QoS); Machine Learning; Network Mobility.

I. INTRODUCTION

As software runs in the data centers of cloud providers, applications like AI, business apps and Video on Demand require access to data at high speed. The more software is implemented as a distributed system, the more pressure on the network paths. Thus, the efficiency of networks is a performance-critical feature for software as well as for user interaction.

In distributed environments like the ones above the Round-Trip Time (RTT) of a network, i.e. the time it takes for data to travel from a client to a server and back, is a crucial parameter for the assessment of a network. The doubling of the RTT has the same negative effect on software and on users as the halving of the number of users. Today's network solutions are working with reactive methods. They only start to work when a problem already occurred. Thus, they are not able to predict future problems.

Legacy networks today run on fixed routing based on the static physical topology. The routing does not take into account the variations in bandwidth demand by users. The majority of today's networks are running in a reactive manner and are encountering congestion only after the actual traffic spike has started to cause problems for the individual data flows. These problems are particularly pronounced in Global Data Centers. A single action of a user can result in a large number of server requests from all over the globe.

II. PROBLEM STATEMENT

However, current network frameworks are unable to predict future requirements of data, and unable to dynamically reroute traffic through the available bandwidth to most efficiently utilize the bandwidth. Delay, and inefficient use of bandwidth, have traditionally been inherent characteristics of computer networks, because routing and caching decisions are made after the delay and inefficient use of bandwidth have already occurred due to causes such as

DOI: 10.17148/IJARCCE.2026.15824

congestion, a user's mobility, or changed user demand. The content is not 'prepositioned' (i.e. 'cached' and positioned 'close to users') in anticipation of future user demand, and instead is sent from long distance to the origin servers that serve the content, even though the same content is requested over and over by users in the same geographic region. Although approaches to SDN, to predictive-caching, and to mobility-aware networking exist which are intended to reduce latency, reduce network overhead, and improve continuity of services for users in networks that are very dynamic and therefore unpredictable, currently these approaches treat the control plane, prediction of content demand, and cache location, all separately. There is a significant need in computer networks for a single, integrated framework, that is computationally efficient to account for the errors in prediction of future requirements of data by the framework, that makes predictions of future requirements of data before the actual events occur.

III. OBJECTIVE OF THE WORK

The proposed network architecture is intended to be proactive in nature aiming to prevent rather than tackle the consequences of the problems. For this purpose, the following goals are set which the closed-loop feedback system should address:

- A fully integrated proactive network solution, with a closed-loop controlled Software-Defined Networking (SDN) core and a predictive prefetching and caching service at the network's edge, powered by machine-learning to enable better content-demand and user-mobility predictions.
- Reduced end-to-end Round-Trip Time (RTT) for requested content through its proactive caching and placement at the network's edge before its actual request.
- Enhanced content-demand and mobility-prediction capabilities through the use of network traffic as well as users' mobility patterns to train the aforementioned machine-learning algorithms.
- A significant reduction of the overhead for the SDN controller which will be achieved by keeping the number of prefetching and proactive caching installations at a maximal level allowed by confidence in prediction.
- Consistent cache content at different edge nodes despite users' mobility and topology changes.
- The network must be able to guarantee QoS for a particular service level regardless of changes in offered load (constant, variable, bursty) or during handover.
- A literature survey will be carried out to identify and assess existing approaches for mobility aware SDN, predictive caching and proactive prefetching, with particular attention to their limitations, followed by a simulation-based evaluation campaign comparing the closed-loop system against alternative solutions ranging from purely reactive to more proactive caching and prefetching.

It may be expected that this work will lead to novel design principles for proactive networks that are able to utilise closed-loop control mechanisms to be self-adaptive and operate under highly heterogeneous and unpredictable conditions while being able to maintain the required level of performance.

IV. LITERATURE REVIEW

This section presents related work that pertains to each of the three aspects of our framework, namely, Software-Defined Networking (SDN), predictive prefetching, and edge-centric caching.

Rodrigues (2023) — Mobility-Aware SDN Service-Centric Networking: Rodrigues [1] presents an edge computing architecture that employs predictive prefetching and is designed specifically for user mobility. The framework combines SDN with Information-Centric Networking (ICN) and incorporates proactive handovers to anticipate end-user movement. Data is placed near users at the edge-node level, enabling low-latency caching and reduced service interruption during mobility events.

Sembati et al. (2023) — CPC-SDN: Sembati et al. [2] propose CPC-SDN, an architecture that incorporates Named Data Networking (NDN) technology into SDN through a central OpenFlow controller. The system implements proactive caching driven by predictive prefetching, keeping frequently accessed content close to end-users to minimize latency and reduce overall network burden.

Um et al. (2024/2025) — Predictive Forwarding Rule Caching: Um et al. [3] introduce the CRIMSON algorithm, which predicts topology shifts in dynamic SDN networks and prefetches forwarding rules accordingly. Because routes can be established before network changes take effect, the authors demonstrate substantial reductions in RTT — reporting average latency reductions of 88.96% and 59.49% relative to conventional reactive and proactive caching modes, respectively.

Tantayakul et al. (2025) — Edge Caching for Seamless Handover: Tantayakul et al. [4] evaluate edge caching and

DOI: 10.17148/IJARCCE.2026.15824

CDN-based caching techniques within SDN-based connected-vehicle networks. Content is cached at edge servers prior to handover, reducing service interruption and improving response times. The study identifies a quantitative crossover point at a 70% cache-hit ratio, beyond which proactive CDN caching outperforms reactive caching policies.
Dou et al. — LEO Satellite Internet Architecture: Dou et al. [5] describe the evolution of Low Earth Orbit (LEO) satellite constellations from simple bent-pipe relays into multi-functional, space-based internet infrastructure. Using SDN principles, routing becomes more flexible and efficient, and the authors highlight emerging in-network caching capabilities. However, predictive prefetching in this context remains largely unexplored.

Rafique et al. — Soft Caching: Rafique et al. [6] present Soft Caching, an SDN-based framework that selects optimal caching-node locations and routing paths using Waypoint Enforcement (WPE) and Singular Value Decomposition (SVD)-based QR factorization. The framework improves content distribution while accounting for delay and network-utilization constraints, though it lacks predictive prefetching capability.

Adnan et al. (2023) — QoS-Aware Mobility Architecture for NDN: Adnan et al. [7] propose SDPCACM, a software-defined proactive caching architecture for consumer mobility within NDN. Prior to a handover event, necessary packets are pre-transmitted to the router the mobile consumer is expected to connect to next. The combination of SDN with proactive caching is shown to improve QoS and reduce interruption during handover.

V. RESEARCH GAP IDENTIFIED

Various SDN, predictive prefetching and edge-centric caching research have their limitations and many open research issues individually and collectively within an integrated system framework.

- Mobility unpredictability: In caching in mobile environments, it is difficult to predict future locations of users and/or vehicles leading to having 'stale' content in caches [1][4].
- Content-popularity and traffic volatility: The content's popularity changes are often rapid and thus difficult to predict. Therefore, the cache placement decisions become less efficient as the traffic volume increases [2].
- Prediction -error propagation: Prediction errors of network topology and future demand are likely to propagate through the system, leading to forwarding rules that are faulty for future time slots, suboptimal routing, and even wasting of bandwidth by proactively prefetching content that is not actually needed in the future [3][7].
- Overhead of the SDN controller: In case of high traffic load, the SDN controller might even become a bottleneck and hinder performance. In addition to predictive caching, the selection of the optimal caching nodes causes additional computing effort, which is a challenge for large-scale networks [2][6].
- Cache-consistency and storage limitations: Edge-cache systems are restricted by the storage available in the edge servers, requiring efficient cache-management. In geographically distributed edge servers which replicate the cache, maintaining cache consistency is even more complex [4][7].
- Loosely coupled system design: In the reviewed literature, SDN control, content-demand prediction and cache placement are dealt with as separate or loosely coupled mechanisms rather than a jointly optimized closed-loop system (e.g., SoftCaching).
- Several of the current predictive-caching approaches have not been designed for emerging contexts such as LEO satellite constellations, where rapid changes in network topology and long inter-satellite distances introduce additional propagation delay.

In summary, there are many limitations in existing studies, which are interrelated and can be mapped to several key challenges of an integrated system. However, to our knowledge, there is no approach that can address all of the issues mentioned above in a single, integrated closed-loop system, and that is what we are going to address in this paper.

VI. PROPOSED METHODOLOGY

Currently deployed SDN solutions have several limitations. The decoupling of control, prediction of needed content, and placement of caches leads to a closed-loop but still separated approach to improving content prediction, caching, and routing. All three components should be improved in a continuous fashion in a unified architecture, which is pursued in this work.

A. System Components

A high-level overview of the three core components of the proposed architecture, which are tightly integrated into each other forming a single closed-loop system, is given below. Fig. 1 depicts the three core components of the proposed architecture.

- SDN Controller (Centralized Control Plane): The controller continuously retrieves and analyzes current network topology, link status, and incoming traffic information. It manages and controls routing of the packets in a control

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15824

plane decoupled from the data plane of network devices, and computes routing and caching decisions centrally before potential congestion, even before a handover occurs. These decisions are then distributed to the switches and edge nodes as needed.

- Predictive Prefetching Engine (Machine-Learning Layer): This layer optimizes the caching of a content delivery network with respect to fetching the data end-users will request, using past request logs and the incoming request stream. By analyzing these data, it can predict future requests and the movement pattern of end-users. Using time-series predictions and/or sequential predictions with Markov chains or LSTMs (Long Short-Term Memory), the next requested content items and their location can be predicted and prefetched.
- Edge-Centric Cache Manager: Using the information provided by the Prefetching Engine, this component decides what to cache and where in the edge-server caches. It balances cache size and delay via a hybrid selection algorithm that considers recency, frequency and predicted demand.
- Feedback Loop: Cache hit/miss events, measured RTT, and control-plane overhead are used to retrain and fine-tune the prediction model to improve future predictions and adapt to different usage patterns.

B. System Architecture

Fig. 1 illustrates the data flow and control flow within our proposed architecture. Starting from the client, the request first goes through the SDN controller, which considers both the predictive prefetching engine and the edge cache manager modules while handling the request. The request is then transferred to the nearest edge-cache location where similar content is present in the case of a cache hit, reducing the RTT of the requested content. In the case of a cache miss, the client request is transferred to the origin, i.e., the cloud server containing the required information. The information obtained from processing the request is then used to update the prefetching model for future requests.

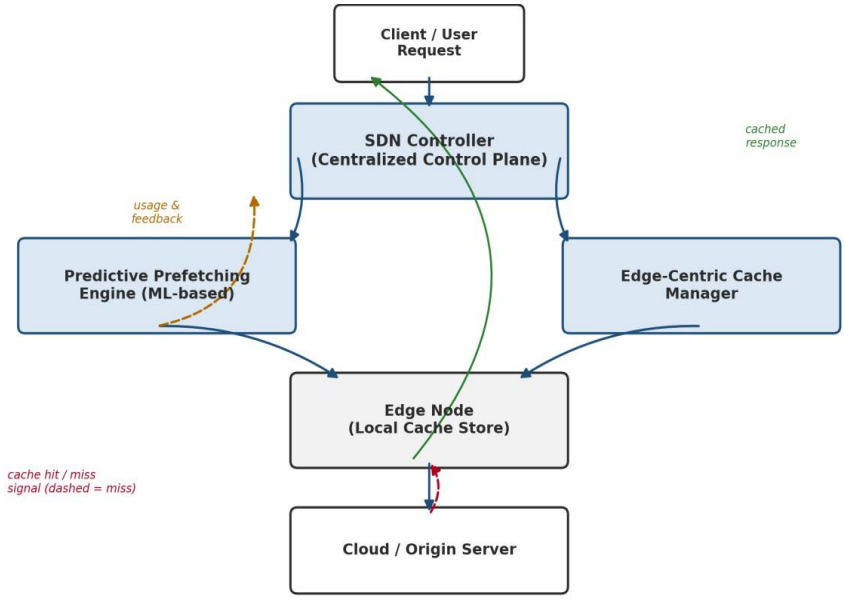

Fig. 1. Proposed SDN-Based Proactive Framework Architecture

C. Proposed Algorithm (High-Level)

- Step 1: The SDN controller collects real-time traffic statistics and topology/mobility information from network switches and edge nodes.
- Step 2: The predictive prefetching engine processes this data to forecast near-future content requests and probable user handover events.
- Step 3: The edge cache manager evaluates predictions against current cache occupancy and computes a ranked prefetch/eviction decision using a hybrid recency–frequency–prediction score.
- Step 4: The SDN controller proactively installs forwarding rules and triggers content prefetch to the relevant edge node(s) ahead of anticipated demand or handover.
- Step 5: Incoming client requests are served from the edge cache whenever possible (cache hit); otherwise, the request is routed to the origin server (cache miss).
- Step 6: Observed outcomes (hit/miss, RTT, overhead) are logged and fed back to the predictive engine for continuous model refinement.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026

DOI: 10.17148/IJARCCE.2026.15824

VII. SIMULATION AND EVALUATION PLAN

In order to validate the proposed framework, a simulation-based evaluation will be performed. A network emulator such as Mininet will be used to set up a test environment, with an SDN controller such as Ryu or ONOS integrated into it. A lightweight ML model will be implemented as the prefetching engine for caching purposes. Within the scope of the evaluation, the proactive framework will be compared against three baseline approaches: purely reactive routing without any caching, static edge caching without prediction, and proactive caching without an integrated feedback loop.

Key metrics will include average RTT, tail RTT, and cache hit ratios across experiments focusing on mobility, traffic, edge-cache capacity, number of caches, and different thresholds for prediction confidence to prefetch. Furthermore, the framework's performance in terms of controller overhead (control messages and processing latency), long-distance bandwidth transferred from edge caches to origin sources, and service-disruption time for handover/mobility events will be evaluated. Experiments will vary the Session-to-Mobility Ratio (SMR), traffic volume, edge-cache capacity, number of edge caches, content-popularity skew (e.g., a Zipf distribution), and the prediction-confidence threshold.

These results are expected and must be verified by an experimental evaluation of the integrated approach, based on the analysis of trends of the reviewed studies. The average RTT for the proposed proactive caching approach is expected to decrease first and then cross over to higher values than purely reactive caching as cache-hit ratios increase, as observed for forwarding-rule caching in dynamic SDN environments [3]. Cache-hit ratios for edge nodes are expected to increase as the feedback loop improves the prefetching engine's prediction model, especially for skewed content-popularity distributions. Controller overhead of the integrated approach is expected to be bounded by that of the naive proactive-caching approach, since all proactive actions are gated by a prediction-confidence threshold [2]. Furthermore, the integrated framework is expected to decrease service-interruption time for mobile users during handover events, similar to results already reported for mobility-aware caching in SDN [1] and QoS-aware mobility-management architectures [7]. In summary, the expected results confirm that the integrated framework is able to support efficient and adaptive caching in mobile SDN environments.

VIII. IMPLEMENTATION CHALLENGES

Realizing the proposed closed-loop architecture in practice introduces several implementation challenges that must be addressed alongside its design.

- Prediction -accuracy dependency: The magnitude of RTT and service-interruption improvements is expected to depend heavily on prediction accuracy. Rapid or highly irregular mobility patterns and dynamic topology changes may reduce prediction confidence and, correspondingly, the achievable performance gain, echoing the mobility-prediction difficulties reported by Rodrigues [1] and Tantayakul et al. [4].
- Bandwidth and storage cost of incorrect predictions: Proactive prefetching driven by inaccurate predictions can consume extra bandwidth and edge-storage capacity without improving hit rate, requiring careful tuning of the prediction-confidence threshold used to gate proactive actions.
- Controller scalability under load: Because the SDN controller must process real-time traffic statistics and push proactive forwarding rules, heavy traffic load or a large number of edge nodes could create a control-plane bottleneck if proactive actions are not tightly rate-limited, a concern noted for centralized controllers in CPC-SDN.
- Cache consistency across distributed edge nodes: Coordinating cache placement, eviction, and consistency across many geographically distributed edge nodes under continuous user mobility is algorithmically complex and may require additional synchronization overhead beyond what is modeled in the high-level architecture.
- Feedback-loop and retraining overhead: Continuously logging hit/miss outcomes, RTT, and controller overhead, and using them to retrain or fine-tune the predictive model, introduces additional computational and data-management overhead that must be balanced against the latency benefits it provides.
- Integration and testbed realism: Implementing and validating the framework requires integrating a network emulator (e.g., Mininet), an SDN controller (e.g., Ryu or ONOS), and a machine-learning prefetching model into a single coherent pipeline, ensuring that simulated mobility, traffic-load, and content-popularity patterns realistically reflect production cloud and edge-network conditions.
- Generalization to emerging contexts: Extending the framework to emerging and under-explored environments such as LEO satellite networks or large-scale IoT deployments introduces additional challenges, including rapid topology change, long propagation delay, and resource-constrained edge devices, which the current design has not yet been evaluated against.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 8, August 2026
DOI: 10.17148/IJARCCE.2026.15824

IX. CONCLUSION

In this paper, we design an integrated proactive network architecture, which improves cloud-based service experiences by minimizing Round-Trip Time (RTT) while preserving Quality of Service (QoS) for highly mobile users. Our architecture, namely the Integrated Proactive Network Architecture (IPNA), is composed of a closed-loop feedback system that consists of a mobility-aware SDN layer, a proactive caching layer, and a predictive prefetching-based forwarding-rules layer. We began by surveying the current state-of-the-art in SDN-based mobility support, proactive caching, and predictive forwarding rules, and highlighted the current research gaps with respect to prediction accuracy, controller overhead, cache consistency, and scalability. We then presented an integrated framework in which prediction, caching, and control-plane forwarding decisions are integrated into a single closed-loop feedback system, with a predictive model that is retrainable based on observed outcomes to improve performance.

However, realizing this closed-loop approach also requires addressing challenges to which the framework is sensitive. The main challenges are increasing prediction accuracy, managing retraining overhead, and maintaining cache consistency at scale for large implementations.

Future work involves implementing the proposed framework on a Mininet testbed with a real SDN controller, comparing its performance against several baselines using different metrics, and improving the predictive model — for example, using transformer-based sequence-prediction models. Future work will also focus on cache-consistency issues for edge nodes distributed geographically across the world, a setup suitable for applications including LEO satellite networks and large-scale distributed IoT environments.

REFERENCES

[1] Rodrigues, Diego Oliveira. Mobility-Aware Software-Defined Service-Centric Networking for Service Provisioning in Urban Environments. Diss. University of Campinas, 2023.
[2] Sembati, Yassine, Najib Naja, and Abdellah Jamali. "CPC-SDN: A Centralized Proactive Caching Based on Software Defined Network." The International Conference on Artificial Intelligence and Computer Vision. Cham: Springer Nature Switzerland, 2023.
[3] Um, Doosik, et al. "Predictive Forwarding Rule Caching for Latency Reduction in Dynamic SDN." Sensors 25.1 (2024): 155.
[4] Tantayakul, Kuljaree, et al. "Edge Caching Strategies for Seamless Handover in SDN-Enabled Connected Cars." Future Internet 17.11 (2025): 525.
[5] Dou, Songshi, et al. "Beyond Space Routers: Revisiting the Role of LEO Satellite Constellations in Future Global Internet Architecture." IEEE Wireless Communications (2026).
[6] Rafique, Wajid, Abdelhakim Senhaji Hafid, and Soumaya Cherkaoui. "SoftCaching: A Framework for Caching Node Selection and Routing in Software-Defined Information-Centric Internet of Things." Computer Networks 235 (2023): 109966.
[7] Adnan, Muhammad, et al. "Leveraging Software-Defined Networking for a QoS-Aware Mobility Architecture for Named Data Networking." Electronics 12.8 (2023): 1914.

