Postprint: Heterogeneous Cloud Radio Access Network Handover Method Using Machine Learning and BILP Model

Authors: Gong Zhuo, Tang Rujin, He Libo

Date: 2019-04-01T00:00:00+00:00

Abstract

In Heterogeneous Cloud Radio Access Networks (H-CRAN), mobile users may experience poor handover (HO) performance. To achieve optimal handover performance in H-CRAN networks, a heterogeneous cloud radio access network handover method based on machine learning and binary integer linear programming (BILP), termed SDHDE, is proposed. In the baseband pool, a radio controller is utilized to enable users to receive HO information through southbound API communications in the Radio Access Network (RAN); the controller provides this information to SDHDE via the northbound API, and SDHDE processes HO decisions for each user. Simulation results demonstrate that, compared with other recent schemes, the proposed scheme can effectively reduce the percentage of handover failures and improve network throughput.

Full Text

Preamble

Vol. 37 No. 5
Application Research of Computers
ChinaXiv Cooperative Journal

Heterogeneous Cloud Radio Access Network Handover Method Using Machine Learning and BILP Model

Gong Zhuo, Tang Rujin, He Libo
(Hunan Agricultural University, Changsha 410128, China)

Abstract: In heterogeneous cloud radio access networks (H-CRAN), mobile users may experience poor handover (HO) performance. To achieve optimal

HO performance in H-CRAN networks, this paper proposes a handover method for heterogeneous cloud radio access networks based on machine learning and binary integer linear programming (BILP), called SDHDE. In the baseband pool, a wireless controller enables users to receive HO information through the southbound API communicating with the radio access network (RAN). The controller provides this information to SDHDE via the northbound API, which processes HO decisions for each user. Simulation results demonstrate that compared with other recent schemes, the proposed scheme can effectively reduce the percentage of HO failures and improve network throughput.

Keywords: heterogeneous cloud radio access networks; software-defined wireless networking; switching decision; naive bayes; binary integer linear programming

0 Introduction

Heterogeneous cloud radio access network (H-CRAN) represents a new candidate architecture for next-generation mobile wireless networks (5G) [1]. H-CRAN combines the advantages of both heterogeneous networks (HetNet) and cloud radio access networks (cloud-RAN, C-RAN), enabling it to meet future data transmission demands [2]. The H-CRAN architecture employs centralized cloud-based load processing technology to deploy dense HetNets, thereby improving spectrum and energy efficiency through enhanced network optimization and inter-layer interference management.

Compared with traditional cell-based cellular networks, H-CRAN provides complete wireless coverage while guaranteeing prioritized data transmission rates. However, user equipment (UE) must frequently perform vertical handovers (VHO) within H-CRAN, which introduces a critical challenge: high-speed mobile users may trigger numerous handovers, requiring the processing of massive requests within short time intervals. Consequently, scalable mechanisms are needed to minimize the impact on user quality of service [3].

To avoid unnecessary handovers, researchers have proposed the concept of software-defined networking (SDN) [4], which transforms network operation layers into a software-coded layer with logically centralized control in wireless networks, also known as software-defined wireless networking (SDWN). SDWN can be applied to handover control, thereby implementing a software-defined handover optimization (SDHO) approach that demonstrates favorable performance in high-density networks [5].

Literature [6] analyzed the challenges and possibilities of using SDHO in all-IP networks. Literature [7] presented examples of how centralized SDHO controller architecture can dynamically adapt and manage radio resources in large-scale RAN networks. However, most existing schemes suffer from high latency, erroneous handover indications, and inappropriate access point selection, making them unsuitable for H-CRAN networks [8]. This paper proposes an SDHO solution where controllers are deployed near the BBU pool of H-CRAN networks, enabling centralized control over decisions across the entire control region. These controllers exhibit excellent scalability and support software-defined handover decision engines (SDHDE) for implementing handover management in H-CRAN networks. SDHDE can transform traditional UCHO handovers into a network-centric seamless handover. Candidate network selection for each UE is based on centralized decisions regarding the entire network state provided by the control plane, thereby converting the problem into an optimization framework.

1 Background and Related Work

Literature [9] presents research findings on general VHO problems. However, few studies specifically target H-CRAN. Therefore, this paper addresses two distinct research directions: handover criteria and the utilization of other network environments as potential solutions for future SDHO deployment in H-CRAN.

1.1 Handover Criteria

Cellular and IP networks have adopted multiple handover criteria to address vertical mobility support for user equipment. For instance, in IP networks, IPv4 and IPv6 [10,11] mobile internet protocols serve as standard communication protocols that enable location-independent routing of packets while maintaining permanent UE IP addresses.

Media independent handover (MIH) [12] protocol represents a standardized mechanism for executing vertical handovers between heterogeneous and layer-2 access technologies. MIH unifies all media-specific technologies under an abstract interface, thereby enabling interoperability between radio access technologies (RAT). However, this unified approach introduces numerous challenges. The method may be difficult to implement because a large number of RAT-specific standards (e.g., MIH for IEEE and LTE HO for 3GPP-based networks) must be extended to comply with the MIH protocol. Furthermore, current MIH standards lack a context-aware module to adapt to network dynamics and mechanisms for uniformly distributing load across the network. MIH is limited to context-agnostic scenarios and cannot guarantee network optimality to address SDHO requirements in H-CRAN.

1.2 SDWN-Based Handover Control

Recently, numerous handover control methods for HetNets have been proposed. Literature [13] introduced an SDN-based small cell offloading method that processes offloading of heterogeneous access points in the control region based on dissatisfaction parameters and end-user classification. Literature [14] proposed a seamless connectivity method between SDWNs that employs a handover control algorithm to reduce packet loss and latency, thereby maximizing quality of experience (QoE). Literature [15] utilized an ant colony algorithm (ACA) to achieve mobility robustness optimization (MRO) objectives in cellular mobile networks. The proposed method minimizes radio link failure (RLF) and unnecessary handovers by adopting a cost function that considers the number of handover failure events in ACA.

Literature [16] presented an interference mitigation and handover management method for heterogeneous cloud small cell networks. Coordinated transmission/reception techniques based on centralized signal processing provided by the baseband pool can mitigate co-channel interference between small cells for edge user equipment, thereby improving small cell spectrum efficiency. Literature [17] proposed a gradient method and cost function-based MRO scheme for LTE cellular networks. The first method detects RLF events and unnecessary handovers, while the second method iterates the previous algorithm using a gradient-based approach, aiming to converge to a minimized cost function.

All the aforementioned work originates from HetNets or traditional cellular networks and cannot leverage the full new capabilities offered by H-CRAN. This paper provides a novel approach that considers normalization of network parameters (such as link delay, signal-to-noise ratio, etc.) to ensure fair acquisition of global information for each UE. Moreover, the proposed real-time SDHDE solution makes the VHO problem scalable in self-organizing mobility management H-CRAN.

2 SDHDE in H-CRAN

In an SDWN-based H-CRAN, user equipment communicates with the RAN through different access points. Each access point connects to a BBU pool that processes wireless functions including signal processing, media access control (MAC), radio resource control (RRC), packet data convergence protocol (PDCP), and radio link control (RLC). These functions are controlled by the SDWN controller in the control plane using southbound APIs, which can execute various control functions including mapping H-CRAN topology, retrieving wireless resource information from access points, and monitoring access points. All control functions can be invoked from different routines in the application plane through northbound APIs. As user equipment moves across different H-CRAN access points, handover events are triggered. These events must be detected and coordinated by the SDWN controller in the control plane. Message processing occurs according to the logic implemented at the controller, which can be dynamically adjusted or changed by instances in the application layer. In this context, to alter controller logic, this paper designs an SDHDE in the application plane to execute a new decision technique that changes how the SDWN controller processes handovers. The SDHDE technology for optimizing H-CRAN handover is presented below, constructed through five phases.

2.1 Data Acquisition

When executing handover decisions, SDHDE must first obtain information from the forward layer to acquire the current operational state of the entire H-CRAN network. This paper employs a user-centric approach to receive data, utilizing Algorithm 1 to obtain access points within the coverage range of user equipment throughout the H-CRAN network.

Algorithm 1 Access Point Detection and Metric Collection Algorithm

When a user equipment executes Algorithm 1, it requires the current channel number (channelNumber), the maximum number of scanned channels (maxChannelNumber), and the MAC address of the source access point as input parameters. Specifically, since access points can operate on different channels, the UE needs to scan each channel from 1 to maxChannelNumber to send probe requests. When the loop allows user equipment to periodically collect network information from these access points within its transmission range, the user equipment first sets the physical layer channel number to 1 through the variable channelNumber. To collect network metrics such as signal-to-interference-plus-noise ratio (SINR) and packet error rate (PER), the next loop allows the user equipment to send probe requests (probeRequest) to actively search for network devices in the coverage area of the current channelNumber until the condition is satisfied. Subsequently, the user equipment prepares to accept incoming probe responses (probeResponse) received from access points through the WaitProbeResponse function. Once the probe request is received, the user equipment can retrieve network metrics such as RSSI, SINR, PER, MAC address of the scanned access point, and channel number of the scanned access point.

2.2 Data Management

Simply aggregating metric values from user equipment cannot provide sufficient information for user-centric handover decisions. Optimal handover decisions that improve the overall network state must consider numerous network parameters that cannot be obtained by a single user equipment alone. Therefore, this paper considers obtaining an enhanced global perspective from various network-related parameters at the control layer (such as bandwidth, delay, and packet loss rate). Modeling based on collected and received network information yields a directed dynamic graph.

Let the graph be defined as a triple $G(E, A)$, where E represents the set of nodes (i.e., access points, fixed and mobile nodes), and A represents the set of arcs between each pair of nodes. The graph can be represented as $E = [1 \cdots n]$ and $A = [1 \cdots a]$, where each edge (i, j) or $(j, i) \in A$ and $i, j \in E \cup \cdots \in$.

Figure 1 [Figure 1: see original paper] shows a directed graph with 5 nodes and 8 edges. Since entities are shipped together, this paper treats variable x as a binary value. By defining a binary integer programming problem, normalized parameters allow finding the optimal path that leads to the best cost for each user equipment. This paper assumes that the capacity of arcs in the network is the bandwidth of network connections, with associated costs determined by the cost matrix c. Additionally, other constraints include flow conservation and available link bandwidth.

For bidirectional edges, if arc (i, j) is a wired connection, then bandwidth is b_{ij}, delay is d_{ij}, and packet loss rate is p_{ij}. If arc (i, j) is a wireless connection, for each user equipment k (or i) within the coverage range of access point j (or i), in addition to the same parameters as wired connections, it also includes: RSSI r_{ij}, SINR s_{ij}, and PER per_{ij}.

Once the aforementioned data is received and stored by the SDWN controller, it becomes necessary to normalize all raw heterogeneous parameters and estimate them. In the SDWN controller, each parameter x can be normalized by calling function $x'()$:

$$x' = \begin{cases} \frac{x - x_{\min}}{x_{\max} - x_{\min}} & \text{if } x_{\max} \neq x_{\min} \\ 0 & \text{otherwise} \end{cases}$$

where $x_{\max}$ is the maximum value of the same type of parameter x currently received, $x_{\min}$ is the minimum value of the same type of parameter x received, and x' is the normalized parameter of x. After normalization, each parameter x is converted to x' with $0 \leq x' \leq 1$. Therefore, it can be considered that all parameters have been normalized. Next, for each connection (i, j), a cost function $c(i, j)$ can be defined:

$$c(i, j) = w_{\text{rssi}} \cdot \text{RSSI} + w_{\text{sinr}} \cdot \text{SINR} + w_{\text{per}} \cdot \text{PER} + w_d \cdot d + w_p \cdot p$$

where $w_{\text{rssi}} + w_{\text{sinr}} + w_{\text{per}} + w_d + w_p = 1$.

From the perspective of the entire network state, the controller can better manage QoS in a logically centralized manner, allowing flexible and adaptive adjustment of weights based on metric importance and traffic type. For example, in typical voice call applications, the weights of PER and SINR will be significantly increased by the controller due to retransmission effects. Furthermore, the SDWN controller constructs a cost matrix (CostMatrix) defined as follows:

$$\text{CostMatrix}_{ij} = \begin{cases} c(i, j) & \text{if connection is within coverage range of PoA} \\ +\infty & \text{otherwise} \end{cases}$$

Specifically, considering the heterogeneity of H-CRAN, for wired connections, this paper only considers delay and packet loss metrics, while for wireless connections, RSSI, SINR, PER, and delay are considered. At this point, since all metrics have been collected, stored, and normalized, the SDWN controller is ready to begin executing handover decisions.

2.3 Handover Optimization Problem Modeling

By constructing the cost matrix, the SDWN controller possesses all necessary information to model network selection as a binary (0 or 1) integer linear programming (BILP) problem. Specifically, network selection is formulated as a binary minimum-cost multi-commodity network flow problem (0-1 MCMCNF), whose primary objective is to find the best set of parallel flow paths for transporting commodities within the same underlying network structure. Since algorithms for minimum-cost flow problems implicitly solve shortest path problems and require treating k flows as a single entity, the problem can be defined as a triple (s_k, t_k, f_k) representing the set of commodities to be transmitted in the network, where s_k is the source node of the k-th commodity, t_k is the destination node of the k-th commodity, and f_k is its demand. Additionally, B_k represents the bandwidth occupied by the k-th commodity when transmitted from source node s_k to destination node t_k. For any set of workflows (x, y), it is impossible to have $x = s \wedge y = t$. That is, workflows with identical source and destination nodes cannot exist. Therefore, the 0-1 MCMCNF problem can be expressed as the following 0-1 integer programming model:

$$
\begin{aligned}
\min \quad & \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \\
\text{s.t.} \quad & \sum_{j:(i,j) \in A} x_{ij}^k - \sum_{j:(j,i) \in A} x_{ji}^k = \begin{cases} 1 & \text{if } i = s_k \\ -1 & \text{if } i = t_k \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in V, \forall k \in K \\
& \sum_{k \in K} B_k x_{ij}^k \le b_{ij} \quad \forall (i,j) \in A \\
& x_{ij}^k \in \{0,1\} \quad \forall (i,j) \in A, \forall k \in K
\end{aligned}
$$

Considering the problem scale, the number of variables is $|A| \times |K|$, where x_{ij}^k represents the decision variable for forwarding services, and c_{ij} represents the unit cost for transmitting commodity k on link (i, j). Therefore, there are $0-1$ variables, and the number of constraints is $|A| + |V| + |K|$, where $|A|$ is the number of arcs, $|V|$ is the number of nodes, and $|K|$ is the number of commodities. Clearly, the 0-1 integer programming problem is NP-complete [18]. To demonstrate this NP-complete 0-1 integer programming algorithm, consider a Boolean equation with constants $C_1, C_2, ..., C_m$, where for all i, assuming without loss of generality that $|C_i| \geq 1$, we can define:

This integer linear programming represents each Boolean value with a binary variable and each clause with a constraint requiring the sum of its literals to be at least 1. For this purpose, if the corresponding Boolean variable appears as a positive literal in the clause, the left side of the constraint includes x_j; if a negative literal appears, it includes $(1 - x_j)$. The above form is obtained by moving all constants to the right side of the equation.

2.4 Handover Execution Timing Strategy

This paper proposes a new algorithm based on received SINR values to solve the timing problem of the handover process, enhancing mobility robustness. After solving the 0-1 integer programming problem, all tasks can be divided into two cases:

Case 0: No user equipment requires handover.

Case 1: At least one user equipment requires handover.

In Case 1, the possibility of handover failure must be considered before triggering handover, and such phenomena should be minimized as much as possible. Specifically, SDHDE must make handover execution decisions for the following situations that may occur after triggering handover:

a) SDHDE decides to send a specific instruction to trigger handover for a user equipment, but the handover fails because the RSS level of the target access point is too low, meaning the handover was triggered too early. After detecting handover failure, the user equipment reconnects to the source access point. This situation can be defined as early handover.

b) SDHDE decides to send a specific instruction to trigger handover for a user equipment, but the handover fails because the device' s RSS level is too low, causing the user equipment to lose connection with the source access point before receiving the handover instruction, meaning the handover was triggered too late. After detecting handover failure, the user equipment reconnects to the source access point. This situation can be defined as late handover.

c) SDHDE decides to send a specific instruction to trigger handover for a user equipment, but the target user equipment has just experienced a handover within a short period (e.g., approximately 4-6 seconds), and the user equipment connects back to the source access point again. This handover failure is caused by the ping-pong effect, and thus this type of handover failure can be defined as ping-pong handover.

This paper considers the above situations as handover failure events, denoted by symbol failureHO. Since user equipment handover consumes network resources to retransmit user equipment access information to new access points, reducing the number of unnecessary user equipment handovers can also minimize network overhead. To this end, this paper introduces a handover trigger limit parameter $\delta(\tau)$ for adaptive handover operations, achieving a trade-off between optimal network handover and network performance.

Parameter $\delta(\tau)$ represents the limiting value after executing the handover process. It can be a deterministic value estimated based on H-CRAN characteristics before SDWN deployment, or a value continuously adjusted over time. After solving the 0-1 MCMCNF problem, the SDWN controller must check the difference between $\mathrm{SINR}_{\mathrm{targetPoA}}(\delta, \tau)$ and $\mathrm{SINR}_{\mathrm{sourcePoA}}(\delta, \tau)$, where δ represents the user equipment and τ represents the instantaneous moment when SINR values are collected. The difference between these two values should be greater than φ. The SINR difference for the same user equipment δ at the same moment τ for different access points can be defined as:

$$\Omega(\delta, \tau) = \mathrm{SINR}_{\mathrm{targetPoA}}(\delta, \tau) - \mathrm{SINR}_{\mathrm{sourcePoA}}(\delta, \tau)$$

If the state meets the requirement, the target access point becomes the best candidate for the specific user equipment under the current H-CRAN network state. Since wireless network environment characteristics are dynamically changing, network states may change in a rapid and unpredictable manner. Therefore, this paper proposes the following algorithm to minimize the percentage of handover failures.

As a general rule, the φ value needs to be set sufficiently large to avoid the ping-pong effect. However, if φ increases beyond a certain value, handover performance degrades. This is because the handover is not fast enough, preventing user equipment from timely establishing connections with optimal candidate access points. Therefore, when deploying SDHDE, if the goal is to balance the network aggressively, it can be assumed that $\varphi = 0$. Moreover, to minimize handover percentage and enhance mobility robustness in H-CRAN networks, this paper extends Algorithm 2 to enable adaptive adjustment of parameter φ values based on different handover failure causes. During the SDHDE lifecycle, the value of parameter φ is automatically and continuously updated, with its increase and decrease dynamically performed based on event occurrence until the number of handover failures is reduced. Assuming variable ψ contains unnecessary or failed handovers (i.e., failureHO), the adaptive adjustment algorithm for parameter φ is shown in Algorithm 2.

Algorithm 2 Handover Failure Algorithm

Algorithm 3 Adaptive Parameter φ Adjustment Algorithm

Premature handover is often caused by poor user equipment mobility or poorly defined motion models; delayed handover is often caused by scenarios where user equipment has high mobility models. Therefore, moderate adjustment of φ can achieve better handover timing in non-uniform scenarios.

2.5 Handover Execution

The handover decision is implemented using machine learning concepts. The machine learning model consists of four components: learning environment, learning unit, knowledge base, and learning execution unit, as shown in Figure 2 [Figure 2: see original paper]. The learning environment provides useful information for the learning process, which is processed by the learning unit to improve the knowledge base. The execution unit then processes corresponding tasks based on this information, transmitting feedback to the learning unit during processing or making changes to adapt to the system.

During operation, the system continuously learns and improves from existing examples using environmental information from heterogeneous cloud networks, adjusting relevant applications to perfect the learning system. The system accumulates rich knowledge about heterogeneous network handovers. This enables the decision-making system to adaptively complete handovers and decisions based on changes in environmental information and user requirements.

Naive Bayes is a commonly used machine learning algorithm. The Naive Bayes formula is:

$$P(C_k|X) = \frac{P(C_k)P(X|C_k)}{P(X)}$$

where C_k represents a class; X represents the metric values of software modules. The classifier marks modules with metric values X in the test dataset as dp or ndp, taking into account the distribution in the training dataset. Labeling is completed according to the class with the higher probability $P(C_k|X)$. Since the denominator in the formula is the same for all classes, it does not affect the final labeling result. Therefore, it can be removed from the formula and expressed as:

$$P(C_k|X) \propto P(C_k)P(X|C_k)$$

$P(C_k)$ and $P(X|C_k)$ must be obtained from the training dataset. However, software modules with metric values X may not exist in the training dataset. Due to this problem, Naive Bayes assumes feature independence. The probability of each metric is calculated separately. In this case, the Naive Bayes formula becomes:

$$P(C_k|X) \propto P(C_k)\prod_{i=1}^{n} P(x_i|C_k)$$

Let S represent the environmental state, including received signal strength, location information, speed, and other environmental information at the receiver. When a network handover occurs, a description of the current network state information S is generated. The system processes this description by first searching for similar examples in the knowledge base and comparing the matching degree with the state description S. If the match is successful, the handover example is added to the knowledge base; otherwise, the handover operation is executed. The specific process is shown in Figure 3 [Figure 3: see original paper].

3.1 Simulation Setup

The SDHDE scheme is deployed in an H-CRAN scenario consisting of a 2D rectangular grid area of size 1000×3000 mm. eNBs and small cells are randomly deployed in the grid and connected to four fully-connected OpenFlow switches. Three of these switches are connected to three different servers that generate data traffic constituting the scenario. All H-CRAN network parameter settings are shown in Table 1 .

A log-distance path loss model is configured for each access point in the experiment to predict signal propagation behavior in the used wireless channel. Additionally, a constant speed propagation delay model is employed to represent the total time consumed by signals from source to sink. The propagation speed is constant and equal to the speed of light in vacuum.

The received signal strength $RSSI_{i,j}$ for the i-th user equipment belonging to set N to the j-th access point belonging to set M can be expressed as:

$$RSSI_{i,j} = \text{txPower} - L_0 - 10n \log_{10}(d_{i,j})$$

where txPower is in dBm, n is the path loss distance exponent, L_0 represents the path loss at unit distance (i.e., 1m), $d_{i,j}$ represents the distance between user equipment and access point, and the value satisfies $0 \leq RSSI_{i,j} \leq 120$ dBm. Generally, data transmission applications require $RSSI_{i,j} \geq -80$ dBm.

The signal-to-interference-plus-noise ratio $SINR_{i,j}$ for the i-th user equipment is calculated using the Gaussian interference model. This paper defines the packet error percentage as the ratio of the number of erroneous received packets to the total number of received packets:

$$PER = 1 - (1 - BER)^L$$

where BER represents the bit error rate and L represents the packet length.

All OpenFlow switches are connected to the SDN controller. However, BBUs and all access points are conventionally connected to a modified Floodlight SDWN controller. Therefore, the control layer can collect wired or wireless network-related statistics. The scheme is deployed in the OpenNet simulation network, which is a software-defined wireless local area network based on Mininet and NS-3. Additionally, data traffic is generated by iPerf and VLC, where iPerf generates general data traffic and VLC generates data streams. Network throughput is collected by the bwm-ng tool. All data is repeatedly collected until its confidence level reaches at least 95%.

3.2.1 Handover Failure Occurrence

As handover failure events increase, the QoS and QoE of user equipment degrade. User equipment performing handover too early or too late wastes time and may even lose connectivity. To investigate situations that may cause handover failures, this paper attempts to evaluate handover failure events by randomly deploying 300 user equipment in the H-CRAN network. Specifically, regarding handover failure events (failureHO), this paper considers an additional factor (i.e., pingpongHO). If the time interval between two handovers executed by user equipment is less than 4 to 6 seconds, this factor increases. Furthermore, SDHDE detects earlyHO (target access point SINR less than -5dB) or lateHO events. If the SINR of the target access point (for earlyHO) or the SINR of the source access point (for lateHO) is less than -5dB, these two events are triggered.

The SDHDE is deployed in the H-CRAN scenario, and the experiment is repeated 20 times. These user equipment are divided into three categories based on their mobility speeds: pedestrian [1 m/s], car [10 m/s], and train [40 m/s]. As shown in Figure 4 [Figure 4: see original paper], slow-moving user equipment [1 m/s] with very low φ values (less than 3 dB) will face extremely high handover failure rates. If φ continues to decrease and eventually becomes negative, the impact of antenna power becomes negligible, which under other conditions might be an important factor in determining erroneous handovers. In contrast, high-speed user equipment, such as cars [10 m/s], exhibit stronger robustness to lower φ values, primarily because they only stay at handover boundaries or base station intersections for very short periods. Nevertheless, high-speed mobile user equipment still faces a high possibility of handover failure, even when φ values are large, and this rate often remains constant. The main reason for the constant handover failure rate is the ping-pong effect—because cars or trains move too fast, they leave the small cell coverage area before fully establishing connections, unable to utilize base station capabilities.

3.2.2 Throughput Evaluation

The proposed SDHDE scheme focuses on communication transmission scenarios for a single user equipment throughout its entire mobility path. In the experiment, user equipment communicates with available servers and receives a video stream while moving linearly through 8 base station cells. To verify the superiority of the proposed scheme, it is compared with the scheme proposed in literature [7]. The experiment evaluates different types of user equipment, including pedestrian [1 m/s], car [10 m/s], and train [40 m/s]. The comparison results for the three different types of user equipment are shown in Figure 5 [Figure 5: see original paper].

In the simulation, the algorithm calibrates the φ parameter in SDHDE, with calibration results of 3 dB for pedestrians, 1 dB for cars, and 2 dB for trains. In Figure 5, the x-axis represents the time elapsed in the experiment (in seconds), and the y-axis represents the cumulative amount of data received. The results show that for pedestrians (Figure 5(a)), SDHDE increases user equipment throughput by approximately 32%. Figure 5(b) demonstrates that user equipment throughput improves by about 11%. For high-speed trains, Figure

5(c) indicates that SDHDE enhances throughput by approximately 6%. In fact, high-speed mobile user equipment cannot continuously achieve high throughput in small areas, resulting in significantly slower throughput growth compared to other types of user equipment.

Table 2 shows the handover failure percentages for both schemes.

Table 2 Handover Failure Percentage

Scheme	Handover Failure Percentage
Literature [7] Scheme	
Proposed SDHDE Scheme	

Table 2 shows that the proposed SDHDE scheme achieves a smaller handover failure percentage compared to literature [7]. In the train scenario, the proposed scheme' s handover failure percentage is significantly lower than that of literature [7]. However, due to the excessive speed of trains, the user equipment throughput improvement shown in Figure 5(c) is obviously inferior to other scenarios, which directly leads to shortened simulation time for related experiments. Nevertheless, although the SDHDE scheme only brings limited throughput improvement (about 6%) in a short period (within 200 s), this cumulative improvement becomes quite significant in long-term usage.

3.2.3 Throughput Comparison Under High Load

The SDHDE scheme was designed considering antenna overload situations, where overloaded antennas cannot well support new user equipment. This paper evaluates user equipment throughput in H-CRAN when some antennas face user equipment overload. Except for several antennas in the mobility path that may be evaluated by SDHDE as having insufficient wireless resources, the throughput measurement method, experimental environment, and configuration are consistent with the previous experiments. Additionally, user equipment moves along a random path in the scenario at speeds of 1 m/s, 10 m/s, and 40 m/s.

Figure 6 [Figure 6: see original paper] shows the cumulative data received by user equipment after using the scheme from literature [7] and the proposed scheme. In this experiment, the threshold parameter ψ is set as load balancing. Then, a data stream is sent from a server to user equipment v currently connected to an RRH. After 45 s, data traffic is generated from a server to user equipment connected to the same RRH as v. The data stream transmitted to v lasts 30 s, so after the marked 45 s, the wireless link of this RRH is blocked. Subsequently, once data is collected from the environment, SDHDE sends an appropriate instruction to v to trigger handover to other access points with the lowest cost. Therefore, in this case, the cell will be the best handover choice.

Once the original RRH is no longer overloaded (marked at 75 s), SDHDE solves the 0-1 MCMCNF problem and sends a message to v to execute a new handover, switching back to its original access point and restoring the original settings. Meanwhile, in the scheme from literature [7], user equipment v cannot sense the blockage occurring in the wireless link, so handover is not triggered even when throughput drops. Considering the percentage difference between these two cases, the proposed SDHDE scheme improves data reception by approximately 17.64% compared to literature [7] during the simulation time.

4 Conclusion

SDHDE uses handover information collected by the SDN controller from user equipment and executes software-defined decisions through machine learning algorithms. This decision process consists of five phases: data acquisition, data management, handover optimization problem planning, handover timing strategy, and handover execution. The goal of these phases is to improve overall network performance. SDHDE decisions obtain optimal connection points for user equipment migration by solving a 0-1 integer programming problem. Subsequently, user equipment migrates to corresponding access points according to SDHDE decisions. Experimental results demonstrate that the SDHDE scheme achieves a smaller handover failure percentage compared to literature [7]. For user equipment connected to overloaded access points, the SDHDE scheme improves user equipment throughput by approximately 17.64% compared to literature [7].

References

[1] Du Xiuli, Dong Zelong, Chen Bo, et al. Heterogeneous network handoff algorithm based on pre-judgment and improved grey correlation [J]. Computer application research, 2017, 34(2): 587-590.

[2] Beyene Y D, Jantti R, Tirkkonen O, et al. NB-IoT Technology Overview and Experience from Cloud-RAN Implementation [J]. IEEE Wireless Communications, 2017, 24(3): 26-32.

[3] Ordonez-Lucena J, Ameigeiras P, Lopez D, et al. Network slicing for 5G with SDN/NFV: concepts, architectures, and challenges [J]. IEEE Communications Magazine, 2017, 55 (5): 80-87.

[4] Wang Mengmeng, Liu Jianwei, Chen Jie, et al. Software definition network: security model, mechanism and research progress [J]. Journal of Software, 2016, 27(4): 969-992.

[5] Thi M T, Huynh T, Hasegawa M, et al. A rate allocation framework for multiclass services in software-defined networks [J]. Journal of Network & Systems Management, 2017, 25 (1): 1-20.

[6] Yang Shunneng, Ho Shuwei, Lin Yibing, et al. A multi-RAT bandwidth aggregation mechanism with software-defined networking [J]. Journal of Network & Computer Applications, 2016, 61 (C): 189-198.

[7] Ceylan O, Caglar A, Tugrel H B, et al. Small satellites rock a software-defined radio modem and ground station design for cube satellite communication [J]. IEEE Microwave Magazine, 2016, 17 (3): 26-33.

[8] Hobiger T, Haas R, Löfgren J. Software-defined radio direct correlation GNSS reflectometry by means of GLONASS [J]. IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing, 2017, 9(10): 4834-4842.

[9] Louta M, Bellavista P. Bringing always best connectivity vision a step closer: challenges and perspectives [J]. IEEE Communications Magazine, 2013, 51(2): 158-166.

[10] Geng Daoqu, Zhu Dapeng, Yu Yanping, et al. Research and implementation of uNAT64 mechanism for 6LoWPAN access to the Internet [J]. Journal of Sensing Technology, 2016, 29(8): 1284-1289.

[11] Lu Huijuan, Liu Yaqing, Xia Haixia, et al. Research and simulation of RPL routing protocol in 6LoWPAN [J]. Journal of Chinese Computer Systems, 2016, 37(1): 83-87.

[12] Chiang Mengshu, Huang Chunming, Chau P B, et al. A forward fast media independent handover control scheme for proxy mobile IPv6 (FFMIH-PMIPv6) over heterogeneous wireless mobile network [J]. Telecommunication Systems, 2017, 65(4): 699-715.

[13] Arslan Z, Erel M, Özcevik Y, et al. SDoff: a software-defined offloading controller for heterogeneous networks [C]//Proc of Wireless Communications and Networking Conference. Piscataway, NJ: IEEE Press, 2014: 2827-2832.

[14] Ferrús R, Koumaras H, Sallent O, et al. SDN/NFV-enabled satellite communications networks: opportunities, scenarios and challenges [J]. Physical Communication, 2016, 18 (P2): 95-112.

[15] Chen Shaipei, Yang Ji, Li Yong, et al. Multiconstrained network intensive vehicle routing adaptive ant colony algorithm in the context of neural network analysis [J]. Complexity, 2017, 2017(1): 1-9.

[16] Zhang Haijun, Jiang Chunxiao, Cheng Julian. Cooperative interference mitigation and handover management for heterogeneous cloud small cell networks [J]. IEEE Wireless Communications, 2015, 22(3): 92-99.

[17] Huang Miaona, Chen Jun. A conflict avoidance scheme between mobility load balancing and mobility robustness optimization in self-organizing networks [J]. Wireless Networks, 2018, 24(1): 271-281.

[18] Zhang Zeqiang, Hu Yang, Chen Chong. An improved artificial bee colony algorithm for solving the disassembly line balance problem [J]. Journal of Southwest Jiaotong University, 2016, 51(5): 910-917.