



Photonic integrated circuits based optimization and enhancing data transmission for radio access networks using machine learning model

R. Asha Rajiv¹ · Shambhu Bhardwaj² · Vikram Singh³ · Dakshinamurthy V. Kolluru⁴ · Mohit Kumar Sharma⁵ · B. Ashwini⁶

Received: 30 August 2023 / Accepted: 10 November 2023 / Published online: 27 December 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

By using edge caching and edge computing, fog radio access networks (F-RANs) are viewed as suitable architectures to serve Internet of Things services. Current research on resource management in F-RANs, however, mostly takes into account a static method with a single communication mode. Resource management in F-RANs becomes highly difficult due to network dynamics, resource variety, and linkage of resource management with mode selection. This study suggests a unique method for data transmission and optimisation in radio access networks utilising a machine learning model. Here, photonic integrated circuits with dynamic optimisation are used to improve network optimisation. Then, data is sent using MASORL, a multi-agent self-organizing reinforcement learning system based on a fog edge model. The experimental study is done in terms of network efficiency, scalability, QoS, throughput, and energy consumption. We show how employing team learning rather than individual learning agents might improve network performance. Finally, in order to give scholars in the field a road map, we identify problems and unresolved concerns.

Keywords Radio access networks · Network optimization · Data transmission · Machine learning model · Fog edge model

✉ R. Asha Rajiv
Asha.rajiv@jainuniversity.ac.in

¹ Department of Physics, School of Sciences, JAIN (Deemed-to-be University), Bangalore, Karnataka 562112, India

² College of Computing Science and Information Technology, Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, India

³ Department of Computer Science and Engineering, Sanskriti University, Mathura, Uttar Pradesh, India

⁴ Corporate and Research Activities, UGDx School of Technology, ATLAS SkillTech University, Mumbai, India

⁵ Department of Electrical Engineering, Vivekananda Global University, Jaipur, India

⁶ Department of Electronics and Communication Engineering, Presidency University, Bangalore, India

1 Introduction

Future IoT applications may be served effectively by fog radio access networks (F-RANs) with the help of edge caching and edge computing. Health monitoring, low-latency services, analytics on large amounts of data from the Internet of Things, and so on are all examples of such offerings (Xiang et al. 2020). In an F-RAN, each piece of user equipment (UE) may switch between many different communication modes, such as device-to-device (D2D), fog radio access point (FAP), cloud radio access network (C-RAN), and so on. Performance analysis, radio resource allocation, collaborative design of cloud and edge processing, cache size effect, and many other areas of study have recently been conducted on F-RANs (Cao et al. 2021a). Network slicing, a novel technique being studied in the context of 5G, has the potential to accommodate a wide range of use cases and business models. The idea behind network slicing is to create customised services by orchestrating and chaining together network slice instances. Network slicing helps 5G networks in an economical way by offering flexible support of multiple applications (Iturria-Rivera et al. 2022). Network slicing in radio access networks (RANs) are researched as a crucial component of network slicing to further enhance end-to-end network performance. Although network slicing is an excellent way to address 5G service needs, it still faces enormous difficulties. Traditional core network slicing techniques are solely driven by business, ignoring RAN characteristics. Network slicing, however, varies depending on the network architecture, such as in heterogeneous networks or cloud RANs (CRANs) (Fang et al. 2022). It may be advantageous to take into account RAN characteristics and network slicing together. Second, developing applications have stricter performance requirements. With 4G, many services are offered on the same network but with varying requirements. However, it is inefficient to use the same network to deliver different services. For instance, Internet of Things services demand extremely large connections, yet a high data rate is not crucial. Furthermore, while enormous connections are not necessary for VR applications, they still require a high data rate. As a result, 5G employs a cutting-edge technology called network slicing (NS) to create networks that are appropriate for different services in slices. Slices are chosen based on criteria including throughput, latency, and dependability. In order to meet these needs, slices are also given access to network resources. To implement NS notion, different network resources must be separated into slices, with each slice receiving resources it needs. As opposed to when employing constrained radio resources, RAN is confronted with developing a technique that meets the slice requirements without lowering efficiency (Shi et al. 2020). To do this, it's crucial to take into account changes in the slice state, such as those related to traffic volume and the quantity of user equipment (UE) attached for control. Additionally, the number of slices processed by a base station (BS) fluctuates according to the service utilisation and UE entry and exit from the BS coverage area. Therefore, a mechanism that dynamically distributes radio resources in accordance with the slice status is required. One type of artificial intelligence method called deep learning (DL) may model the functioning of organic neural systems and provide patterns that can be used to decision-making tasks. Smallest component of this system, the neuron, is present in each layer in a predetermined number. The depth of the structure is determined by the quantity of concealed layers. While deep learning (DL) contains several hidden layers as opposed to shallow learning's single hidden layer, this is how the term "deep learning" was coined. The development of DL has gone hand in hand with development of technology that can organise vast

amounts of data. NLP, wireless networking, and computer vision are just a few of the disciplines where it has been successfully used broadly (Vimal et al. 2020).

2 Related works

Radio resources, which include RBs, are split into frequency and temporal domains in RAN slicing. The RB allocation to each slice is determined using a method that author (Shirmohamadi et al. 2022) described as an extension to the RB scheduling algorithm. Throughput was demonstrated to be higher with the extension approach than it was without one. When assigning RBs to a slice, the current technique does take the fulfilment of the slice criteria into account. It's critical to meet the slice criteria in NS. As a result, work (Murti et al. 2021) suggested a mechanism that distributes RBs to slices while taking the slice needs into account. By allocating RBs from slices without requirements, this approach satisfies slice requirements. The effectiveness of the RB distribution, however, was not assessed. It's possible that slices receive more RBs than are required. Author (Chen et al. 2018) suggested a solution that takes slice needs and RB consumption efficiency into account to overcome issue. With this technique, the slices are divided into four categories and given RB allocations. In accordance with the slice criteria, their analysis revealed a 12% increase in allocation efficiency over the scenario without abstraction. The issue is that because the slices are abstracted into four kinds and then given RBs, it was unable to entirely segregate RBs for each slice. In other words, the interference of RBs from other slices may result in a reduction in the extent to which the slice criteria were met (Shahjalal et al. 2023). As a result, it's crucial to only distribute the necessary amount of RBs to each slice, without allowing influence from other slices. In Du et al. (2020), we created an effective one-dimensional search method to identify the best solution to issue of delay optimum computing job offloading inside a MDP framework. The dependency on statistical data on channel quality variations and computing job arrivals presents a problem, though. In Yan et al. (2020), the author used a Lyapunov optimisation approach to study a dynamic compute offloading policy for a MEC method with wireless energy harvesting-capable mobile devices. The similar methodology was used by Chang et al. (2022) and Deka and Sharma (2022) to investigate power-delay tradeoff in context of compute job offloading. Only a roughly optimum solution can be created by the Lyapunov optimisation. The author created an algorithm that uses reinforcement learning and does not require previous knowledge of network data in order to discover the best compute offloading strategy in Zhou et al. (2023). Multiple BSs with various data transmission quality are available to offload a computing workload when MEC encounters an ultra-dense sliced RAN. The expansion of state space in this situation renders the traditional reinforcement learning techniques (Cao et al. 2021b) impractical. Based on Lu et al. (2019), suggested RL with MultiPointer networks (Mptr-Net) to address offloading issue in MEC, and results revealed that their method achieved greater than 98% optimality. To tackle placement problem for virtual network functions (VNF), authors in Lu et al. (2019) also developed a deep RL technique with a sequence-to-sequence method in an effort to reduce power consumption. A deep RL technique for dynamic computation as well as radio resource control in vRAN method was suggested in recent work (Filali et al. 2022) as the vrAIn framework. There is currently no prior effort to use such techniques for functional split optimisation in vRAN, despite the fact that they are promising for tackling complicated combinatorial issues for zero-touch optimisation in wireless networks (Koudouridis et al. 2022; Jiang et al. 2019).

3 System model

In this paper, we'll take into account an ultra-dense service area served by a virtualized RAN with a set $B = 1, B$ of BSs, as shown in Fig. 1. Over the same physical network architecture, both MEC services and conventional communication services are supported. At network's edge, a MEC server is placed into place, giving the MUs powerful computational resources. By carefully outsourcing the generated compute jobs via BSs to MEC server for execution, MUs can anticipate a significantly enhanced computing experience. Consider wireless radio resources are divided into slices for normal communication as well as MEC in order to provide inter-slice isolation.

Consider a typical single Base Station (BS) downlink cellular network system. Time is divided into TTI units of 1 ms, indexed by $t, 1, 2, \dots$. For each TTI, bandwidth is split up into a number of PRBs, designated as $F = 1, 2, \dots, F$. Consider cellular network is divided into a collection of N network slices, where $N = 1, 2, \dots, N$.

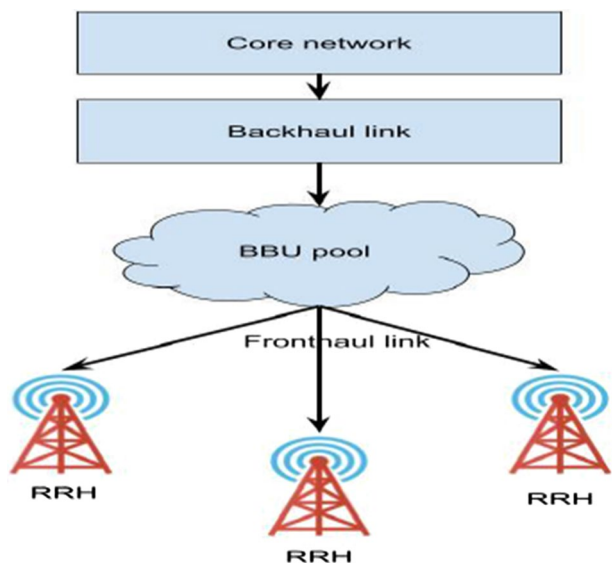
3.1 Signal transmission model

We may easily use Shannon's capacity formulation to evaluate data rate in traditional services, such eMBB with large transmitted packet sizes. The new uRLLC or MTC service, however, differs from typical services in that it transmits little packets (between 32 and 200 bytes in size). Shannon's capacity theory cannot adequately describe data rate in short packet transmissions. Instead, finite block length theory may be utilized to approximation possible data rate of short packet transmission by Eq. (1)

$$V_{i,j,t} = 1 - \left(1 + p_{i,j,t} |h_{i,j,t}|^2 / N_0 \right)^{-2} \tag{1}$$

Instantaneous data rate of UE $i \in U_n$ can therefore be expressed as Eq. (2)

Fig. 1 Three cells in the conventional C-RAN architecture, one RRH in every cell, and a centralised connection to the BBU pool



$$r_{i,t} = \sum_{j=1}^F s_{i,j,t} \cdot r_{i,j,t}, \text{ (bits per TTI)} \tag{2}$$

where the binary variable $s_{i,j,t}$ is set to 0 if the UE i is not assigned the j -th PRB and to 1 otherwise. Additionally, only one UE may simultaneously acquire a PRB. FCFS policy dictates that every UE have a data queue at BS where it can keep incoming packets before transmitting them. At t -th TTI, queue length of UE $i \in U_n$ is designated as $q_{i,t}$ and evolved as follows by Eq. (3)

$$q_{i,t+1} = \max\{q_{i,t} - r_{i,t}/Z_n, 0\} + A_{i,t}, \tag{3}$$

where $A_{i,t}$ is instantaneous packet arrival for UE i during t -th TTI and Z_n is overall packet size (in bits). Additionally, active UEs in slice n are defined as collection of UEs having a nonzero queue length. Transmission-related latency as well as scheduling-related latency make up majority of the packet delay. In formula (4), the data rate of the UE determines the transmission delay, but the scheduling strategy determines the queuing latency. The packet delay in our system model is calculated by adding transmission delay as well as queuing delay. In order to mimic delay of m -th ($m = 1, 2, \dots$) packet arriving at i -th UE's buffer

$$D_{i,m} = W_{i,m} + \delta_{i,m}, \text{ (in TTI)} \tag{4}$$

When UE's average packet arrival rate is low in slice n , queuing delay is almost nil, which causes transmission delay to predominate over packet delays. Applications classify a data packet as dropping out from the perspective of service provisioning when its latency exceeds a maximum tolerated packet delay that has been set. Typically, packet losses are what define how reliable a transmission is. As a result, the likelihood that packet delay is in excess of a predetermined maximum packet delay threshold is defined as PDR of m -th incoming packet at i -th UE's buffer in system model by Eq. (5)

$$\beta_{i,t} = \Pr\{D_{i,m} > D_n^{\max}\}, i \in U_n, \tag{5}$$

where, D_n^{\max} is maximum tolerant packet delay of every UE in slice n . Packet latency of UE, as stated in formula (5), should be taken into consideration as one crucial QoS parameter during service provisioning. In the meantime, PDR of UE, as stated in formula (6), measures communication dependability, which is another crucial QoS parameter. We will thus utilise packet delay as well as PDR as two important metrics to assess QoS performance of the service in the next section.

3.1.1 Photonic integrated circuits with dynamic optimization

By regulating light for lasing, switching, and optical filtering as well as for the trapping and emission of photons, ultra-small cavities play a crucial part in photonic integrated circuits. Ring resonators as well as photonic crystal micro-cavities are the most often used structures. One-dimensional (1D) micro-cavities are ideal for very dense packing since they have a very tiny footprint. Nowadays, Mach-Zehnder modulators have mostly been used to show high performance electrooptical modulation in silicon. These modulator devices are in the mm range in length. Recently, ring resonator-based electro-optic modulators have been shown. The tiny

modulators have modulation frequencies greater than 10 Gb/s and ring diameters as low as 12 m. If the rings are further shrunk to an order of magnitude smaller than for ring resonators. However, only a modulation frequency of 250 Mb/s has been proven thus far. Therefore, more research is necessary to boost modulation speed. The creation of such miniature electro-optic modulators is our key priority. For the arrangement of the modulators to be optimised for modulation frequency, loss reduction, and extinction ratio, we systematically iterated a number of design parameters. With a unique diode arrangement that decreases absorption and offers extremely low energy consumption per bit, goal was to attain 10 GHz modulation frequency. The cavity’s waveguide is a component of a p-i-n diode. Free carriers are injected into or drained from the cavity by providing a voltage to the diode. Through so-called free carrier plasma dispersion effect, voltage modifies silicon’s refractive index in waveguide. Spectral location of the transmission peak is shifted by the cavity’s refractive index change, which enables modulation of the transmitted light’s intensity. With this architecture, it has the lowest footprint yet demonstrated and could support a data throughput of 25 GBd.

The binary indication $x_{ij}(t)$ indicates if the UE i ’s request j is fulfilled at time t . Following are restrictions for allocating resources to network slices by Eq. (6)

$$\begin{aligned}
 \sum_{i,j} F_{ij}x_{ij}(t) &\leq F(t), (i,j) \in A(t), \\
 \sum_{i,j} P_{ij}^C x_{ij}(t) &\leq P^C(t), (i,j) \in A(t), \\
 \sum_{i,j} P_{ij}^T x_{ij}(t) &\leq P^T(t), (i,j) \in A(t),
 \end{aligned}
 \tag{6}$$

where $F(t)$, $P^C(t)$, and $P^T(t)$ represent, respectively, computing, transmit, and available communication (frequency-time blocks) resources at gNodeB at time t . certain of gNodeB’s resources could already is allotted to certain requests that haven’t finished processing, thus some resources might not be accessible. Picking $F_{ij}(t)$, $P_{ij}^C(t)$, and P_{ij}^T for resource allocation optimisation at time t is the narrow aim by Eq. (7)

$$\max \sum_{ij} w_{ij}x_{ij}(t), (i,j) \in A(t)
 \tag{7}$$

We then take into account the temporal horizon optimisation problem. From time t_1 to time t , the resources are updated as follows by Eq. (8)

$$\begin{aligned}
 F(t) &= F(t-1) + F_r(t-1) - F_a(t-1), \\
 P^C(t) &= P^C(t-1) + P_r^C(t-1) - P_a^C(t-1), \\
 P^T(t) &= P^T(t-1) + P_r^T(t-1) - P_a^T(t-1),
 \end{aligned}
 \tag{8}$$

where, At time t_1 , the resources related to frequency, CPU usage, and transmit power are freed. Every request has a lifespan of l_{ij} , and if service begins at time t to fulfil it, request will finish at time $t+l_{ij}$. Define $R(t)$ as collection of requests that have ended at time t . Released as well as allocated resources at time t are represented by Eq. (9)

$$\begin{aligned}
 F_r(t) &= \sum_{(i,j) \in R(t)} F_{ij}, \\
 P_r^C(t) &= \sum_{(i,j) \in R(t)} P_{ij}^C, \\
 P_r^T(t) &= \sum_{(i,j) \in R(t)} P_{ij}^T \\
 F_a(t) &= \sum_{ij} F_{ij}, \\
 P_a^C(t) &= \sum_{ij} P_{ij}^C, \\
 P_a^T(t) &= \sum_{ij} P_{ij}^T \\
 \max \sum_t \sum_{ij} w_{ij} x_{ij}(t), (i, j) \in A(t)
 \end{aligned}
 \tag{9}$$

This issue can be resolved offline on the unrealistic presumption that the gNodeB is aware of all upcoming requests.

3.1.2 Fog edge model based multi agent self-organizing reinforcement learning:

Figure 1 depicts the scenario that was taken into consideration for this work. It is predicated on an F-RAN architecture with three layers: a cloud computing layer, a network access layer, and a terminal layer. BBU pool enables centralised signal processing at the cloud computing layer. Additionally, there are single-antenna L_1 distributed RRHs connected to the BBU pool at network access layer. Additionally, there are M_0 F-APs set up with L_0 antennas. Fog computing allows for the execution of collaborative radio signal processing at dispersed F-APs in addition to the centralised BBU pool.

Figure 2 The terminal layer has K_1 single antenna F-UEs and K_0 single antenna conventional UEs, whose sets are designated as K_0 and K_1 . Traditional UEs, which aim for low power consumption as well as unpredictable bursty traffic arrivals, include industrial monitoring devices and sensors used in agricultural fields. F-UEs can be laptops or cell phones, both of which always have a sizable buffer. A network slice instance is built, consisting of several modes and related physical resources, to offer each F-UE a high data rate. In C-RAN mode, RRHs collaborate to receive uplink data, while BBU pool offers centralised baseband processing and signal detection. Additionally, F-APs are set up for local services to lessen load on fronthaul. Similar to this, network slice instance tailored for conventional UEs offers both CRAN mode as well as F-AP mode. However, goal is to keep classic UEs' transmission latency consistent and power consumption low. F-UEs can also help both network slice instances by using D2D mode. Particularly, F-UEs aggregate data in slice instance for conventional UEs to enable more traditional UEs to connect at once while relaying data traffic of other F-UEs to increase coverage of slice instance for F-UEs. There are N subchannels available for allocation, each having a bandwidth of W_0 . Both orthogonal as well as multiplexed subchannel techniques are taken into account in this article. In the former, strict isolation between slice instances is made possible by the allocation of subchannel n to a maximum of one standard UE i or F-UE j . with contrast, with the latter, numerous conventional UEs and F-UEs can share subchannel n . With this approach, a complex mode selection as well as resource allocation would ensure isolation

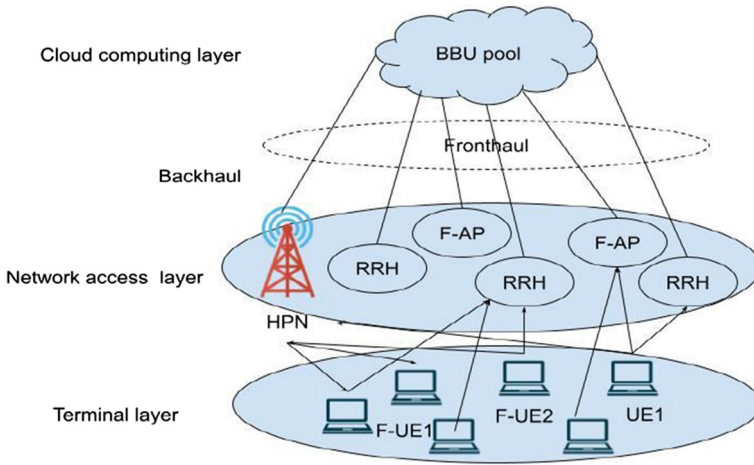


Fig. 2 The RAN slicing architecture’s single antenna system concept, which creates network slices for conventional UEs and F-UEs

between slice instances. Although a mechanism for allocating orthogonal subchannels primarily ensures slice isolation in present efforts. It is still important to look at a multiplexed subchannel allocation technique in order to increase spectrum utilisation by Eq. (10)

$$\mathbb{P}[S_{t+1} = s' | S_t = s] = \mathbb{P}[S_t = s' | S_{t-1} = s] \tag{10}$$

RL must make judgements throughout time in order to maximise the expected value of the return or to choose best course of action. Here are our explanations of our return and policy policies by Eq. (11)

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{11}$$

$$q_{\pi}(s, a) = \mathbb{E}_{\pi} [G_t | S_t = s, A_t = a]$$

Similarly, action-value function is decomposed as follows: $q_{\pi}(s, a) = \mathbb{E}_{\pi} [R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$. Additionally, we may observe the connection between $v_{\pi}(s)$ and $q_{\pi}(s, a)$:

$$v_{\pi}(s) = \sum_{\pi \in \Delta} \pi(a|s) q_{\pi}(s, a) \tag{12}$$

$$q_{\pi}(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{ss'} v_{\pi}(s') \tag{13}$$

$$v_{\pi}(s) = \sum_{a \in A} \pi(a|s) \left(\mathcal{R}_s^a + \gamma \sum_{s' \in S} \mathcal{P}_{ss'}^a v_{\pi}(s') \right) \tag{14}$$

The Bellman equation compares one state’s state-value function to those of other states. As demonstrated in Eq. (15), Bellman equation for $q_{\pi}(s, a)$,

$$q_\pi(s, a) = \mathbb{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'} \sum_{a' \in \mathcal{A}} \pi(a'|s') q_\pi(s', a'). \tag{15}$$

Using the theorem by Eq. (16), we can quickly get an ideal policy by increasing $q(s, a)$ over all actions.

$$\begin{aligned} v_*(s) &= \max_\pi v_\pi(s) \\ q_*(s, a) &= \max_\pi q_\pi(s, a) \\ \pi &\geq \pi' \quad \text{if } v_m(s) \geq v_m(s), \forall s \\ \tilde{\sigma}_\pi(a|s) &= \begin{cases} 1 & \text{if } a = \arg \max q_*(s, a) \\ 0 & a \in \cdot A \end{cases} \end{aligned} \tag{16}$$

Finding best value function is the last remaining problem. This issue is resolved using Bellman optimality equation. Equation (17) demonstrates that optimal state-value function and have a relationship that are found.

$$\begin{aligned} v_*(s) &= \max_a q_*(s, a) \\ q_*(s, a) &= \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'} v_*(s') \end{aligned} \tag{17}$$

In order to create a Bellman optimality equation for v and q , express $q(s, a)$ in terms of $v(s)$ using Eqs. (18).

$$\begin{aligned} v_*(s) &= \max_a \left(\mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'} v_*(s') \right) \\ q_*(s, a) &= \mathbb{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} P_{ss'}^a \max_{a'} q_*(s', a'). \end{aligned} \tag{18}$$

For the purpose of clarity, this research solely takes into account the deterministic case. Alternatively, the problem may be formulated stochastically, which would need thinking about anticipated returns for probabilistic transitions. Our method as well as conclusions are easily applied to stochastic MDPs if expectations are determined with accuracy. Controller selects actions to take in accordance with its policy $h: X \rightarrow U$ using $u_k = h(x_k)$. From any initial state (x_0) and current time ($k=0$), the controller’s goal is to find a method that maximises the discounted return by Eq. (19).

$$R = \sum_{k=0}^{\infty} \gamma^k r_{k+1} = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k) \tag{19}$$

Discounted return accurately captures reward accumulated by controller over time. Long-term performance improvement is the aim of learning, which merely makes use of feedback on immediate, one-step performance by Eq. (20)

$$\begin{aligned}
 Q^h(x, u) &= \rho(x, u) + \sum_{k=1}^{\infty} \gamma^k \rho(x_k, h(x_k)) \\
 Q^*(x, u) &= \rho(x, u) + \gamma \max_{u' \in U} Q^*(f(x, u), u') \\
 [T(Q)](x, u) &= \rho(x, u) + \gamma \max_{u' \in U} Q(f(x, u), u')
 \end{aligned}
 \tag{20}$$

Lipschitz continuity of Q is established utilizing Q-value iteration algorithm, called Q-iteration, which makes use of an a priori task model in form of transition as well as reward functions f. There is a finite L_Q that has the property by Eq. (21)

$$\begin{aligned}
 & \left| Q^*(x, u) - Q^*(\underline{x}, \underline{u}) \right| \leq L_Q (\|x - \underline{x}\| + \|u - \underline{u}\|) \\
 & \left| [T(Q_\ell)](x, u) - [T(Q_\ell)](\underline{x}, \underline{u}) \right| \\
 &= \left| \rho(x, u) + \gamma \max_{u'} Q_\ell(f(x, u), u') \right. \\
 & \quad \left. - \rho(\underline{x}, \underline{u}) - \gamma \max_{\underline{u}} Q_\ell(f(\underline{x}, \underline{u}), \underline{u}') \right| \\
 & \leq \left| \rho(x, u) - \rho(\underline{x}, \underline{u}) \right| \\
 & \quad + \gamma \left| \max_{u'} [Q_\ell(f(x, u), u') - Q_\ell(f(\underline{x}, \underline{u}), \underline{u}')] \right|
 \end{aligned}
 \tag{21}$$

$\left| \rho(x, u) - \rho(\underline{x}, \underline{u}) \right| \leq L_\rho (\|x - \underline{x}\| + \|u - \underline{u}\|)$ for second term by Eq. (22)

$$\begin{aligned}
 & \gamma \left| \max_{u'} [Q_\ell(f(x, u), u') - Q_\ell(f(\underline{x}, \underline{u}), \underline{u}')] \right| \\
 & \leq \gamma \max_{u'} L_{Q_i} \|f(x, u) - f(\underline{x}, \underline{u})\| \\
 & = \gamma L_{Q_i} \|f(x, u) - f(\underline{x}, \underline{u})\| \\
 & \leq \gamma L_{Q_i} L_f (\|x - \underline{x}\| + \|u - \underline{u}\|)
 \end{aligned}
 \tag{22}$$

Lipschitz continuity of Q and f. Therefore, $L_\rho \sum_{k=0}^{\ell+1} \gamma^k L_f L_{Q_{i+1}} = L_\rho + \gamma L_{Q_i} L_f = L_\rho + \gamma L_f L_\rho \sum_{k=0}^{\ell} \gamma^k$ and induction is complete. Taking limit as $\ell \rightarrow \infty$, it follows that $L_Q = L_\rho \sum_{k=0}^{\infty} \gamma^k L_f$.

Algorithm for MASORL

- 1: start replay memory \mathcal{O}^k with a size of U , mini-batch \mathcal{O}^k with a size of S , and the Q -function with two sets θ^k and $\hat{\theta}^k$ of random weights, for $k = 1$
 - 2: repeat
 - 3: After deploying \mathbf{y}^k , observe cost $p(\mathbf{x}^k, \mathbf{y}^k)$ and new network state $\mathbf{x}^{k+1} \in \mathcal{X}$
 - 5: Store $\mathbf{m}^k = (\mathbf{x}^k, \mathbf{y}^k, p(\mathbf{x}^k, \mathbf{y}^k), \mathbf{x}^{k+1})$ in \mathcal{O}^k
 - 7: Update θ^{k+1} with gradient given by (19)
 - 8: Regularly do $\hat{\theta}^{k+1} = \theta^k$
 9. Update epoch index by $k \leftarrow k + 1$
 10. until A predefined stopping condition is gratified
-

4 Experimental analysis

This section includes a summary of suggested system’s performance. Suggested system is implemented in Java. The Java platform’s physical setup consists of an Intel i5/i7 processor, 4 GB of RAM, and a 3.20 GHz CPU speed. A mathematical paradigm is suggested in the design concept to increase security of cloud storage. Security model works in tandem with the end user and the data owner in this strategy. Even when cloud storage is problematic, the owner’s data is safeguarded during data uploading as well as data transmission to the right user. Assume that there is no intercellular interference created. The 1Mbit file size is the default. Transmission rate in backhaul link is set to $R = 100$ Mbps for simplicity. The reward decay is set to 0.9 and learning rate is set to 0.001. Assuming nothing else, we set $U = 50$, $F = 500$, and $N = 5$. Benchmark schemes for simulations are standard method, as well as learning schemes. Operational states of a particular processor as well as UE are then altered, if necessary, based on a greedy scheme action selection. The controller then adjusts the precoding and cache state in line with the transition matrix for each D2D transmitter transit. HPN will assist those UEs with unsatisfactory QoS in D2D mode access C-RAN whenever it receives any QoS violation data from UEs, and controller will turn on all of processors. State change, action, and resulting decrease in system power consumption, which is reward, are then recorded in controller’s replay memory. In order to reduce MSE between goal Q values as well as predicted Q values of DQN, controller will update DQN after a number of interactions by training over a batch of interaction data randomly picked from replay memory. Additionally, controller will adjust the DQN weights to the intended DQN for every longer duration. A dense NN called the adopted DQN is built from an input layer, two hidden layers, and an output layer. Input layer has 14 neurons, whereas output layer has 96 neurons. Every hidden layer has 24 neurons, and ReLu is used as activation function. Table 1 contains a list of all other simulation-related parameters.

It is evident that a lower value of results in higher performance. This is due to the fact that a larger will result in about identical selection probabilities for various actions, even if difference between their Q values widens over the course of learning. Additionally, it is demonstrated that, when compared to $\tau = \tau_0 / \log(1 + \tau e^{\tau_i})$, a logarithmic decreasing $\tau = 0.1$ and $\tau = 0.5$ improves performance. This is due to the fact that logarithmic decrease tends to reduce its value as episode τe^{τ_i} rises, which causes a gradual selection of the best

Table 1 Simulation parameters

Parameter	Value	Parameter	Value
Steps to update DQN	480	Discounted factor	0.99
Capacity of re–play memory	5000	UE efficiency	$\frac{1}{20}$
Adam optimizer	0.0001	RRH power efficiency	$\frac{1}{40}$
Steps to update DQN	3	Noise	10^{-13} W
Every DQN batch size	32	To populate in action selection	1000
Linearly annealing	3000	Fronthaul transmission power for every UE	5w

solutions with a larger likelihood. As can be seen, value of goal power-minus-rate function is severely constrained when overall amount of computing resources is restricted. Power-minus-rate lowers dramatically when computing resource rises, which may be result of following factors. First, more conventional UEs/F-UEs may be supplied locally as the amount of computing resources grows. The power minus-rate may be reduced by increasing mode selection flexibility; secondly, with higher computational power, UEs/F-UEs that typically chose D2D mode can switch to FAP mode. Since the F-UE is not conveying data, it uses less energy, which causes the power-minus-rate to drop even lower.

4.1 Comparative analysis

Table 2 displays the results of a topological study of the network. In this case, we examine two network parameters: UE density and device density. Throughput, scalability, network efficiency, quality of service, and energy usage have all undergone parametric study.

Number of UE analysis is shown in Fig. 3. For comparison, existing Q-Learning achieved throughput of 91%, scalability of 45%, network efficiency of 81%, quality of service (QoS) of 41%, energy consumption of 36%, and MDP achieved throughput of 93%, scalability of 48%, network efficiency of 83%, quality of service (QoS) of 43%, energy consumption of 39%.

From above Fig. 4 analysis carried out based on number of devices. Proposed technique attained throughput of 96%, scalability of 59%, network efficiency of 91%, QoS of 51%, energy consumption of 49%, while existing Q-Learning attained throughput of 92%, scalability of 53%, network efficiency of 86%, QoS of 46%, energy consumption of 43%, MDP attained throughput of 94%, scalability of 55%, network efficiency of 89%, QoS of 49%, energy consumption of 45%.

Since the number of slices the agent controlled was fixed in previous techniques, retraining the model was required if the number of slices changed between training and evaluation. As a result, under the suggested technique, one agent allots rbs to a single slice, and the agent is called several times when there are numerous slices. Rb allocation that is independent of the number of slices is actualized by this design. Additionally, the agent learns to maximise the number of slices in which the criteria are fulfilled while improving the efficiency of rb utilisation. This is done by satisfying the slice requirement with the least amount of rb allocation necessary. Two different sorts of slice designs are presented here: one creates slices by loosely categorising services, while the other creates a slice for each type of service. In this study, we employ the design that establishes a slice for every service

Table 2 Analysis based on network parameters

Network parameters	Techniques	Throughput	scalability	Network efficiency	QoS	Energy consumption
Number of UE	Q-Learning	91	45	81	41	36
	MDP	93	48	83	43	39
	MASORL	95	51	85	45	42
Number of devices	Q-Learning	92	53	86	46	43
	MDP	94	55	89	49	45
	MASORL	96	59	91	51	49

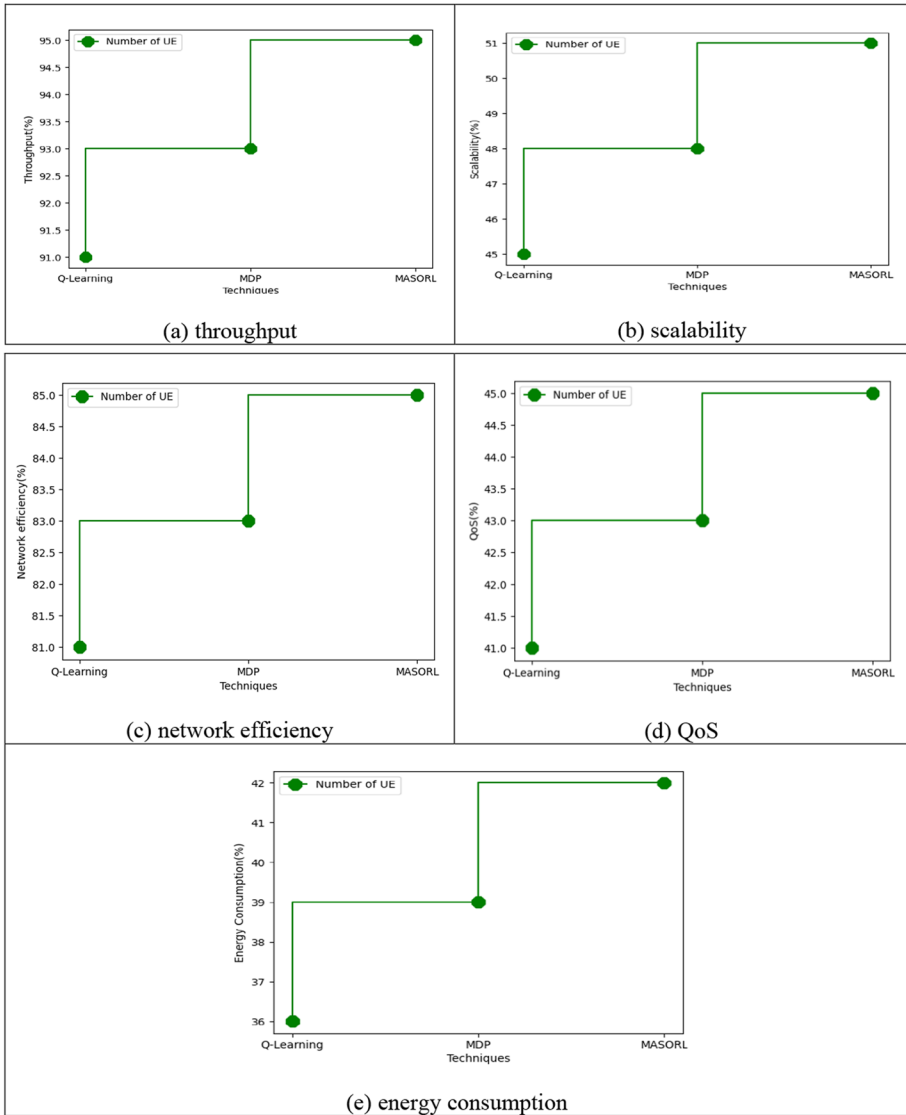


Fig. 3 a–e Analysis for number of UE

category. It is possible that numerous criteria are established for the same item in a slice when defining slices by broadly categorising services. By designating the tightest criteria for services in a slice as the slice requirements, the suggested solution may be used in this situation. Additionally, because each service has its own slice, there are fewer users per slice than there would be if the slices were determined by broadly categorising the services. When there are one or more ues in the slice, a slice is created, and when there are zero ues, the slice is terminated. This indicates that slices are created and terminated more often than when the slices are determined by approximately categorising services.

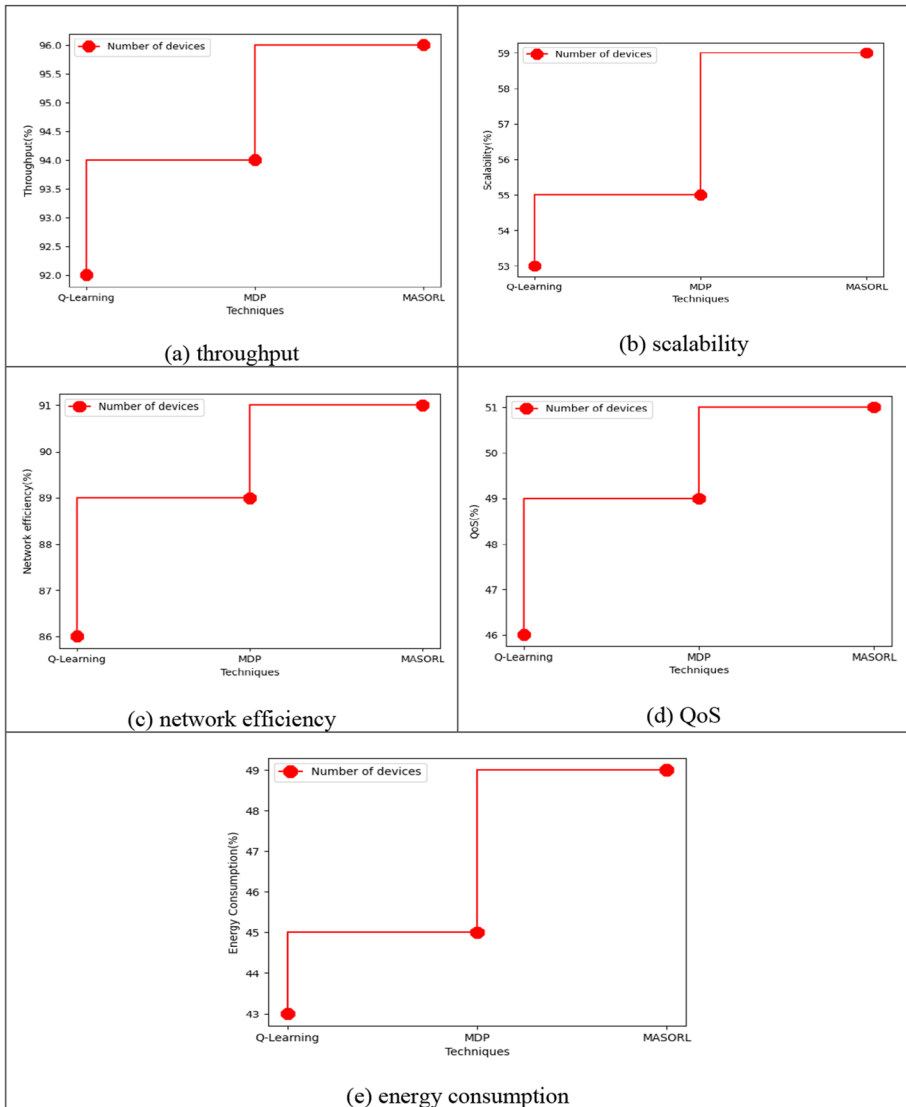


Fig. 4 a–e analysis for number of devices

5 Conclusion

This study suggests a unique technique for radio access network data transmission that uses Photonic Integrated Circuits (PICs) with dynamic optimisation and multi-agent self-organizing reinforcement learning (MASORL) based on the fog edge model. To discover the best course of action without having prior knowledge of network dynamics, we first suggest a double DQN based model computing offloading technique. A Q-function decomposition approach is then integrated with double DQN, which results in a unique learning method for solution of stochastic computing offloading. This is driven by additive nature

of utility function. Compared to the baseline rules, numerical studies demonstrate that our suggested learning techniques significantly enhance computation offloading performance. The connection between the concealed layer's size and processing speed has to be looked into. Investigating factors that influence training outcomes, such as learning rate and batch size, is also crucial. In this study, we demonstrated that the simulator is capable of performing the ideal RB allocation. The slice state could alter in ways that are unique to a real world, though, as the simulator cannot accurately replicate a genuine environment. However, because it takes a lot of time as well as money to train a model exclusively in a real environment, it is not practicable.

Author contributions RAR: Conceived and design the analysis. SB: Writing—Original draft preparation. Dr. VS: Collecting the Data, DVK: Contributed data and analysis stools. MKS: Performed and analysis, Wrote the Paper. BA: Editing and Figure Design.

Funding No funding.

Availability of data and materials All the data's available in the manuscript.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethical approval This article does not contain any studies with animals performed by any of the authors.

References

- Cao, Y., Lien, S.Y., Liang, Y.C., Chen, K.C., Shen, X.: User access control in open radio access networks: A federated deep reinforcement learning approach. *IEEE Trans. Wirel. Commun.* **21**(6), 3721–3736 (2021a)
- Cao, Y., Lien, S.Y., Liang, Y.C., Chen, K.C.: Federated deep reinforcement learning for user access control in open radio access networks. In: *ICC 2021-IEEE International Conference on Communications, IEEE*, pp. 1–6 (2021)
- Chang, Q., Jiang, Y., Zheng, F.C., Bennis, M., You, X.: Cooperative edge caching via multi agent reinforcement learning in fog radio access networks. In: *ICC 2022-IEEE International Conference on Communications, IEEE*, pp. 3641–3646 (2022)
- Chen, X., Zhang, H., Wu, C., Mao, S., Ji, Y., Bennis, M.: Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. *IEEE Internet Things J.* **6**(3), 4005–4018 (2018)
- Deka, K., Sharma, S.: Hybrid NOMA for future radio access: design, potentials and limitations. *Wireless Personal Communications*, pp. 1–16 (2022)
- Du, J., Yu, F.R., Lu, G., Wang, J., Jiang, J., Chu, X.: MEC-assisted immersive VR video streaming over terahertz wireless networks: a deep reinforcement learning approach. *IEEE Internet Things J.* **7**(10), 9517–9529 (2020)
- Fang, C., Xu, H., Yang, Y., Hu, Z., Tu, S., Ota, K., Liu, Y.: Deep-reinforcement-learning-based resource allocation for content distribution in fog radio access networks. *IEEE Internet Things J.* **9**(18), 16874–16883 (2022)
- Filali, A., Mlika, Z., Cherkaoui, S., Kobbane, A.: Dynamic SDN-based radio access network slicing with deep reinforcement learning for URLLC and eMBB services. *IEEE Trans. Netw. Sci. Eng.* **9**(4), 2174–2187 (2022)
- Iturria-Rivera, P.E., Zhang, H., Zhou, H., Mollahasani, S., Erol-Kantarci, M.: Multi-agent team learning in virtualized open radio access networks (o-ran). *Sensors* **22**(14), 5375 (2022)

- Jiang, N., Deng, Y., Simeone, O., Nallanathan, A.: Cooperative deep reinforcement learning for multiple-group NB-IoT networks optimization. In: ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 8424–8428 (2019)
- Koudouridis, G.P., He, Q., Dán, G.: An architecture and performance evaluation framework for artificial intelligence solutions in beyond 5G radio access networks. *EURASIP J. Wirel. Commun. Netw.* **2022**(1), 1–32 (2022)
- Lu, L., Jiang, Y., Bennis, M., Ding, Z., Zheng, F.C., You, X.: Distributed edge caching via reinforcement learning in fog radio access networks. In: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), IEEE, pp. 1–6 (2019)
- Murti, F.W., Ali, S., Latva-aho, M.: Deep reinforcement based optimization of function splitting in virtualized radio access networks. In: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, pp. 1–6 (2021)
- Shahjalal, M., Kim, W., Khalid, W., Moon, S., Khan, M., Liu, S., Jang, Y.M.: Enabling technologies for AI empowered 6G massive radio access networks. *ICT Express* **9**(3), 341–355 (2023)
- Shi, Y., Sagduyu, Y.E., Erpek, T.: Reinforcement learning for dynamic resource optimization in 5G radio access network slicing. In: 2020 IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), IEEE, pp. 1–6 (2020)
- Shirmohamadi, M., Bakhshi, H., Dosararian-Moghadam, M.: Optimizing resources allocation in a heterogeneous cloud radio access network using machine learning. *Trans. Emerg. Telecommun. Technol.* **33**(9), e4570 (2022)
- Vimal, S., Kalaivani, L., Kaliappan, M., Suresh, A., Gao, X.Z., Varatharajan, R.: Development of secured data transmission using machine learning-based discrete-time partially observed Markov model and energy optimization in cognitive radio networks. *Neural Comput. Appl.* **32**, 151–161 (2020)
- Xiang, H., Peng, M., Sun, Y., Yan, S.: Mode selection and resource allocation in sliced fog radio access networks: a reinforcement learning approach. *IEEE Trans. Veh. Technol.* **69**(4), 4271–4284 (2020)
- Yan, S., Jiao, M., Zhou, Y., Peng, M., Daneshmand, M.: Machine-learning approach for user association and content placement in fog radio access networks. *IEEE Internet Things J.* **7**(10), 9413–9425 (2020)
- Zhou, G., Zhao, L., Zheng, G., Xie, Z., Song, S., Chen, K.C.: Joint multi-objective optimization for radio access network slicing using multi-agent deep reinforcement learning. *IEEE Transactions on Vehicular Technology* (2023)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.