Reinforcement Learning-Based Multihop Relaying: A Decentralized Q-Learning Approach

Xiaowei Wang * and Xin Wang

Abstract: Conventional optimization-based relay selection for multihop networks cannot resolve the conflict between performance and cost. The optimal selection policy is centralized and requires local channel state information (CSI) of all hops, leading to high computational complexity and signaling overhead. Other optimization-based decentralized policies cause non-negligible performance loss. In this paper, we exploit the benefits of reinforcement learning in relay selection for multihop clustered networks and aim to achieve high performance with limited costs. Multihop relay selection problem is modeled as Markov decision process (MDP) and solved by a decentralized Q-learning scheme with rectified update function. Simulation results show that this scheme achieves near-optimal average end-to-end (E2E) rate. Cost analysis reveals that it also reduces computation complexity and signaling overhead compared with the optimal scheme.

Keywords: reinforcement learning; Q-learning; multihop network; relay selection

1. Introduction

Multihop relaying is believed to extend transmission range and to form the essential communication structure for many practical networks, such as ad hoc networks and vehicular networks. In these networks, candidate relays for each hop are often clustered. For example, in vehicular networks, a vehicle gets access to a roadside unit (RSU) with help of multiple relay vehicles which are often geographically clustered. Therefore, judiciously designed relay selection policies guarantee a stable and efficient communication path. The optimal selection policy searches for the best path based on maximization algorithm and inter-cluster channel state information (CSI) of all hops. Its computational complexity and signaling overhead are considerably high. Then, decentralized selection schemes have been proposed to reduce costs by compromising a proportion of performance [1–4]. Ref. [2] considered clustered multihop networks and proposed a decentralized relay selection scheme which selects a set of relays. This scheme explores multiuser diversity but causes interference, so the size of selected set should be very small. In [3], a decentralized selection scheme is proposed to choose the best relay for each cluster with the consideration of physical-layer security. Another way to design decentralized relay selection is to set a timer for each node within a cluster, which is the reciprocal of CSI. The node whose timer counts to 0 first is selected as the relay. In spite of these academic efforts, satisfactory tradeoff between performance and cost has not been achieved. Therefore, it is meaningful to investigate brand-new decentralized selection scheme which further narrows the performance gap to the optimal policy.

Recently, machine learning has found its extensive applications in optimizations for wireless communications such as antenna selection [5,6], relay selection [7–9], and power allocation [10,11]. Learning tools used to solve these problems include supervised learning [5–8], reinforcement learning [12], neural network (NN) [9], etc. The flourish of learning-based optimization inspires us to exploit new multihop relay selection schemes. Recently, more complex optimization problems have been solved by reinforcement learning.
in dualhop relay networks, combining relay selection with other techniques such as energy harvesting [13], buffer aided relays [14], device-to-device (D2D) communications [15], access control [16], etc. In [7,8], relay selection for dualhop networks is modeled as multiclass classification and solved by decision tree. However, multihop clustered relaying yields large number of possible paths, which makes classification inefficient. To solve multihop relay selection problem, we will design a novel learning-based scheme. In [17,18], relay selection schemes based on reinforcement learning are proposed for dualhop networks, but these schemes cannot be extended to multihop networks.

In this paper, multihop relay selection is modeled as Markov decision processes (MDP) and solved by reinforcement learning. We propose a Q-learning-based decentralized algorithm which allows each cluster to train its own Q-table and predict relay selection. We aim to reduce computational complexity and signaling overhead while keeping near-optimal average end-to-end (E2E) rate.

2. System Model and Optimization-Based Relay Selection

2.1. Communication Model

We consider a linear multihop network with a source node (S), a destination node (D) and M clusters of relays denoted by $C_1, \ldots, C_M$. $C_m$ consists of $K_m$ decode-and-forward (DF) relay nodes denoted by $R_{m}^{1}, R_{m}^{2}, \ldots, R_{m}^{K_m}$, $m = 1, \ldots, M$. For convenience, we let $C_0$ denote S and $C_{M+1}$ denote D. Then, $K_0 = K_{M+1} = 1$. Neither the relays nor S has direct link to D, except the nodes in $C_M$. The signal transmitted by S is delivered to D along a path composed of $M$ relays selected from the $M$ clusters. At the beginning of the multihop transmission, S broadcasts its data to $C_1$. One member of $C_1$ is selected to receive the data and broadcasts it to $C_2$. By this means, the data is relayed until it is received by D.

The wireless channels in the network are assumed to experience independent and identically distributed (i.i.d.) Rayleigh fading. The noise at each receiver is modeled as complex Gaussian random variable with zero mean and variance $\sigma^2$. $h_{m}^{k,k'}$ denotes inter-cluster complex channel coefficients from $R_{m}^{k-1}$ to $R_{m}^{k}$. When $m = M + 1$, $h_{M+1}^{k,k} = h_{D}^{k}$. Assuming that transmitting power of $R_{m}^{k}$ is $P_{m}^{k}$, received signal-to-noise ratio (SNR) of $R_{m}^{k}$ is given by

$$\Gamma_{m}^{k,k'} = \frac{P_{m-1}^{k} |h_{m}^{k,k'}|^2}{\sigma^2}, m = 1, \ldots, M + 1.$$ (1)

E2E SNR and E2E rate of the multihop communication are expressed as

$$\Gamma_E = \min_m \Gamma_{m}^{k,k'}, m = 1, \ldots, M + 1$$ (2)

and

$$R_{e2e} = \frac{1}{M+1} \log(1 + \Gamma_E).$$ (3)

2.2. Optimization-Based Relay Selection

2.2.1. Optimal Selection

In the considered network, there are $\prod_m K_m$ possible paths. The optimal selection is a centralized maximization-based scheme which chooses the best path. The central controller collects $h_{m}^{k,k'}$ for all $k, k', m$, and computes $\Gamma_{m}^{k,k'}$ and $\Gamma_E$ for each path. The optimal policy is to select the path (relay combination) yielding maximum $R_{e2e}$, as described by

$$\begin{align*}
(1^*, 2^*, \ldots, m^*, \ldots, M^*) &= \arg \max_{\text{all paths}} R_{e2e} \\
&= \arg \max_{\text{all paths}} \min_m \Gamma_{m}^{k,k'}.
\end{align*}$$ (4)

Here, $m^*$ denotes the index of selected relay of $C_m$ and $0^*$ represents S.
Selecting the maximum from $\prod_{m} K_m$ values is of $O(\prod_{m} K_m)$ complexity, and requires all inter-cluster CSI. Designing decentralized multihop relay selection schemes will reduce these costs.

2.2.2. Conventional Decentralized Selection

To distribute the computations, the $M$ relays are selected separately and successively at each cluster. For $1 \leq m \leq M - 1$, the conventional decentralized selection policy is described by

$$m^s = \arg \max_{1 \leq k \leq K_m} \Gamma^{m}_{(m-1)^s k}$$  \hspace{1cm} (5)

and for $m = M$,

$$M^s = \arg \max_{1 \leq k \leq K_M} \min\{\Gamma^{M}_{(M-1)^s k}, \Gamma^{M+1}_{kD}\}.$$  \hspace{1cm} (6)

3. Q-Learning-Based Multihop Relaying

In the considered network, a cluster only requires information from the previous and the next clusters. Hence, multihop transmission is naturally Markov. In this section, we model multihop relay selection as MDP and propose a Q-learning-based decentralized relay selection scheme. The scheme is composed of three phases: initialization, training, and prediction. The tasks of training and prediction are decentralized to each cluster. Hence, each cluster, including D, maintains a Q-table and a reward table; Training and prediction are completed in a successive manner from $C_1$ to $C_M$. The Q-tables are updated for multiple episodes until convergence is reached, and then are used to search for the best relays. First, we provide basic definitions for a standard Q-learning algorithm [19], taking the algorithm on $C_m$ for example.

State: $s_m$ represents the selected relay node of $C_{m-1}$, which broadcasts the data-carrying signal to $C_m$.

Action: $a_m$ represents the relay node selected from $C_m$ which will receive the signal broadcast by $s_m$. For state $s_m = R^{m-1}_k$, possible actions involve all relay nodes of $C_m$, i.e., $a_m \in \{R^m_k | k = 1, \ldots, K_m\}$.

Reward: $r_m(s_m, a_m)$ denotes the reward of state $s_m$ if action $a_m$ is taken, and is stored in the reward table of $C_m$. If $s_m = R^{m-1}_k$, $a_m = R^m_k$, $r_m(s_m, a_m)$ is defined by the SNR from $s_m$ to $a_m$, i.e.

$$r_m\left(R^{m-1}_k, R^m_k\right) = \Gamma^m_{kK}.$$  \hspace{1cm} (7)

Q-value: $Q_m(s_m, a_m)$ denotes the Q-value for a given state-action pair $(s_m, a_m)$, which is defined to evaluate the accumulated value of $s_m$. $Q_m(s_m, a_m)$ is stored in the Q-table of $C_m$ and is obtained by iterative updating.

System parameters include learning rate $\alpha$, discount factor $\gamma$, and error threshold of convergence $\epsilon$. A brief illustration of this scheme is given in Figure 1.
3.1. Initialization

The reward table and Q-table of \( C_m \) are initialized by \( K_{m-1} \times K_m \) tables, \( m = 0, 1, \ldots, M + 1 \). The reward table stores the rewards of all possible state-action pairs. To initialize the reward table, \( C_m \) estimates CSI of the channels from each node of \( C_{m-1} \) to each node of its own, and calculates reward values using (7). The Q-table stores Q-values of all \((s_m, a_m)\) pairs. Each row represents a node in \( C_{m-1} \) and each column represents a node in \( C_m \). All Q-values of tables from \( C_1 \) to \( C_M \) are initialized to be 0. The Q-table of \( D \) is fixed and duplicated from \( D \)’s reward table. The reward table and Q-table of \( C_m \) are described by Tables 1 and 2.

Table 1. Reward table of the \( m \)th cluster.

| \( s_m \) \( a_m \) | \( R^m_1 \) \( R^m_2 \) | \( \gamma^m_{11} \) \( \gamma^m_{12} \) | ... | \( \gamma^m_{1K_m} \) |
|---|---|---|---|---|
| \( R^m_{1-1} \) | \( \gamma^m_{21} \) | \( \gamma^m_{22} \) | ... | \( \gamma^m_{2K_m} \) |
| \( ... \) | \( ... \) | \( ... \) | ... | \( ... \) |
| \( R^m_{K_{m-1}} \) | \( \gamma^m_{K_{m-1}1} \) | \( \gamma^m_{K_{m-1}2} \) | ... | \( \gamma^m_{K_{m-1}K_m} \) |

Table 2. Q-table of the \( m \)th cluster.

| \( s_m \) \( a_m \) | \( R^m_1 \) \( R^m_2 \) | \( Q_m(R^m_{1-1}, R^m_1) \) \( Q_m(R^m_{1-1}, R^m_2) \) | ... | \( Q_m(R^m_{1-1}, R^m_{K_m}) \) |
|---|---|---|---|---|
| \( R^m_{1-1} \) | \( R^m_{2-1} \) | \( Q_m(R^m_{2-1}, R^m_1) \) \( Q_m(R^m_{2-1}, R^m_2) \) | ... | \( Q_m(R^m_{2-1}, R^m_{K_m}) \) |
| \( ... \) | \( ... \) | \( ... \) | ... | \( ... \) |
| \( R^m_{K_{m-1}} \) | \( R^m_{K_{m-1}} \) | \( Q_m(R^m_{K_{m-1}1}, R^m_1) \) \( Q_m(R^m_{K_{m-1}1}, R^m_2) \) | ... | \( Q_m(R^m_{K_{m-1}1}, R^m_{K_m}) \) |
3.2. Training

Training phase is iteratively updating the Q-tables of each cluster in a successive manner for multiple episodes until convergence is reached. The key issue of training is the update function. If we adopt the standard update function given by [19], only the reward of current state is involved, i.e., the rate of current hop. However, in DF multihop relaying, the rate of an individual hop cannot contribute to the calculation of end-to-end rate if the rate of another hop is even smaller. Therefore, the update function should take into account all hops, which is not economical. We predict that the standard update function will not yield high performance. Instead, we revise the standard update function and make it choose the larger one between the rewards of current hop and the next hop. This means that the update function can eventually maximize the end-to-end rate. The new update function is given by (8).

In the beginning of the \( n \)th updating episode, \( s_1 = S \) and \( a_1 \) is randomly selected from \( C_1 \). \( C_2 \) chooses the best action, \( a'_{2} \), by

\[
ad'_{m+1} = \arg \max_{a_{m+1}} Q_{m+1}(a_m, a_{m+1}), \tag{9}
\]

and computes the Q-value and reward of \( a'_{2} \) by

\[
Q_{\max}(m + 1) = Q_{m+1}(a_m, a'_{m+1}), \tag{10}
\]

\[
r_{m+1} = r_{m+1}(a_m, a'_{m+1}). \tag{11}
\]

Then, \( Q_{\max}(2) \) and \( r_2 \) are sent back to \( C_1 \), and are used to update \( Q_1(S, a_1) \) by (8).

For \( C_2 \), \( s_2 = a_1 \), and \( a_2 \) is randomly selected from \( C_2 \). The rest procedures are the same as \( C_1 \), and \( Q_2(s_2, a_2) \) is updated. The above updating procedure for a single cluster is repeated to each of the following clusters to finish this episode. If the error of E2E rate is lower than \( \varepsilon \), the Q-tables are converged and training is completed. Else, a new episode of updating begins.

3.3. Prediction

Successively from \( C_1 \) to \( C_m \), each cluster searches its Q-table and selects the best relay. First, \( C_1 \) searches its single-row Q-table for the action with maximum Q-value and selects it as the best relay. This selection policy is described by

\[
m^* = \arg \max_{a \in c_m} Q_m((m-1)^*, a), \ m = 1, \ldots, M. \tag{12}
\]

Notified of \( 1^* \), \( C_2 \) searches the corresponding row of \( 1^* \) from its Q-table and obtains \( 2^* \). After all clusters complete their relay selection, a multihop path is established through which the source data is delivered to \( D \).

The proposed Q-learning-based relay selection scheme for multihop clustered networks is summarized in Algorithm 1.
Algorithm 1. Q-learning-based multihop relay selection.

Phase 1. Initialization
for \( m = 1 : M + 1 \)
\( C_m \) evaluates and collects \( h_{k'}^m \) for all \( k' \) and \( k \)
\( C_m \) generates a \( K_{m-1} \times K_m \) reward table and initializes it by (7)
\( C_m \) generates a \( K_{m-1} \times K_m \) Q-table.
if
\( m = 1, \ldots, M \), all Q-values are set to be 0.
else
Q-values are duplicated from reward table.
end
Phase 2. Training
\( n = 1, R_0 = 0 \)
while (1)
\( s_1 = S \)
for \( m = 1 : M \)
\( C_m \) randomly selects an action \( a_m \) from \( C_m \)
Request \( Q_{\text{max}}(m+1) \) and \( r_{m+1} \) from \( C_{m+1} \)
Update the \( Q_m(s_m, a_m) \) using (8)
\( s_{m+1} = a_{m+1}^r \)
end
Compute end-to-end rate \( R_n \)
If \( R_n - R_{n-1} < \epsilon \)
break
else
\( n = n + 1 \)
end
end
Phase 3. Prediction
Let \( 0^* = S \)
for \( m = 1 : M \)
\( C_m \) searches in its Q-table and selects the best relay \( m^* \) using (12)
Notify \( m^* \) to \( C_{m+1} \)
\( R_{m^*}^m \) receives the signal transmitted from \( R_{(m-1)^*}^{m-1} \), and broadcasts it to \( C_{m+1} \)
end

4. Performance Evaluation

4.1. Simulation Results
We simulate a multihop network with \( M \) clusters and each cluster contains equally \( K \) relays. Average E2E rate is calculated as the performance metric. Simulation parameters are given in Table 3. We first examine the convergence of the Q-learning-based relay selection. Then, the scheme is compared with optimal scheme given by (4) and conventional decentralized scheme described by (5) and (6).

First, Figures 2 and 3 show the convergence of the proposed Q-learning scheme with respect to \( K \) and \( M \). It is observed from Figure 2 that more iterations are needed for convergence if \( K \) increases. When \( K \) is fixed, we observe from Figure 3 that \( M \) has little impact on the number of iterations for convergence. This means that the proposed scheme applies to long route.
Table 3. Simulation parameters.

| Parameter                        | Value          |
|----------------------------------|----------------|
| Transmit Power $P_m$             | 30 dB          |
| noise power $\sigma^2$          | 1              |
| learning rate $\alpha$          | 1              |
| discount factor $\gamma$        | 0.4            |
| fading parameter $\mathbb{E}[|h_{mk}|^2]$ | 1              |
| convergence threshold $\epsilon$| $10^{-3}$      |

Figure 2. Convergence vs. $K$ when $M = 3$.

Figure 3. Convergence vs. $M$ when $K = 5$.

Figure 4 shows average E2E rates of the three schemes with respect to $K$. The first observation is that the curve of Q-learning scheme is very close to the curve of the optimal scheme, especially when $K$ is small. The gap grows larger when $K$ increases, which indicates
that Q-learning scheme cannot consistently benefit from growing $K$. Without designed update function, the Q-learning scheme achieves the lowest E2E rate and cannot benefit from growing $K$. Another important observation is that when $K \leq 19$, Q-learning scheme clearly outperforms the conventional decentralized scheme. After $K = 19$, the conventional decentralized scheme yields better performance. This is because great $K$ yields large action space, and Q-learning cannot work well with large action space. This issue can be easily avoided because larger $K$ also increases computational complexity. So, normally cluster size should be controlled.

![Figure 4. Average E2E rate vs. $K$ when $M = 2$.](image)

Figure 4. Average E2E rate vs. $K$ when $M = 2$.

Figure 5 illustrates average E2E rates with respect to $M$. As $M$ grows larger, average E2E rates of all three schemes decrease because E2E rate is bounded by the worst hop. We also observe that the curve of Q-learning scheme is very close to optimal scheme. Moreover, Q-learning scheme clearly outperforms conventional decentralized scheme. The advantage of Q-learning scheme keeps unchanged when $M$ increases.

![Figure 5. Average E2E rate vs. $M$ when $K = 5$.](image)

Figure 5. Average E2E rate vs. $M$ when $K = 5$. 
From above figures, we summarize that the proposed Q-learning scheme achieves near-optimal E2E rate. To take the best advantage of it, the proposed scheme is better applied to multihop linear networks with moderate cluster size.

4.2. Cost Analysis

4.2.1. Computational Complexity

The optimal policy selects the path with maximum E2E rate among all $K^M$ paths, leading to computational complexity of $O(K^M)$. In the proposed Q-learning scheme, the complexities of initialization and prediction are $O(K^2)$ and $O(K)$. The main part of training phase is iterative updating of Q-tables, leading to complexity of $O(MK \log \frac{1}{\epsilon})$. In most practical networks, Q-learning scheme is superior to the optimal scheme.

4.2.2. CSI Amount

The optimal scheme is centralized and requires CSI of all $(M - 1)K^2 + 2K$ inter-cluster links reported to the central controller. In the proposed Q-learning scheme, CSI of $(M - 1)K^2 + 2K$ inter-cluster links is only estimated and collected locally between adjacent clusters. Thus, total energy consumed and interference to other transmissions caused by signaling are greatly reduced. In each iterative update, each cluster requires only two values, $Q_{\text{max}}(m+1)$ and $r_{m+1}$, from the next cluster to update its Q-table. This causes extra communication costs. Moreover, each cluster transmits the selected action $a_m$ to the next cluster, costing only $\log_2 K$ bits.

Signaling overhead of multihop networks is mainly due to CSI collection. To evaluate CSI amount, we propose a calculating method which takes into account both the number of channels to be estimated and the length of CSI transmission route. We suppose that the central controller is located at the middle cluster of the multihop path. In the optimal scheme, the CSI of faraway clusters is delivered to the central controller via multihop transmission. Thus, we calculate the weighted sum CSI amount, and let the weights be the numbers of hops needed to collect the CSI. The Q-learning-based scheme and conventional decentralized scheme only require CSI transmission between adjacent clusters, so all weights are 1. The weighted CSI amount of the optimal scheme is calculated by (13). It is not difficult to prove that $C(M, K)$ is always greater than $(M - 1)K^2 + 2K$, CSI amount of the Q-learning-based scheme, for all values of $M$ and $K$. The three schemes are compared in Table 4.

$$C(M, K) = \begin{cases} \frac{(M^2 - 1)K^2}{4} + (M + 1)K, & M = 3, 5, 7, \ldots \\ \frac{M^2K^2}{4} + (M + 1)K, & M = 2, 4, 6, \ldots \end{cases}$$

(13)

Table 4. Comparison of the Q-learning-based scheme with benchmark schemes.

|               | Complexity     | CSI Amount               | E2E Rate       |
|---------------|----------------|--------------------------|----------------|
| optimal       | $O(K^M)$       | $C(M, K)$                | optimal        |
| conventional decentralized | $O(MK)$ | $(M + 1)K$ | below optimal |
| Q-learning    | $O(MK \log \frac{1}{\epsilon})$ | $(M - 1)K^2 + 2K$ | near optimal |

5. Conclusions

In this paper, we have proposed a decentralized Q-learning-based relay selection scheme for multihop clustered networks. The scheme is composed of three phases: initialization, training, and prediction. A new update function for Q-values is designed to promote prediction performance. Simulation results show that the proposed Q-learning scheme achieves near-optimal performance and outperforms conventional decentralized
scheme in terms of average E2E rate. The advantages of Q-learning scheme also lie in lower computational complexity and smaller cost for collecting CSI.

**Author Contributions:** Conceptualization, investigation, methodology, software and writing—original draft preparation, X.W. (Xiaowei Wang); validation and writing—review and editing, X.W. (Xin Wang). All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is supported by National Natural Science Foundation of China under grant 61703264 and Major Program of Shanghai Municipal Education Commission under grant 2021-01-07-00-10-E00121.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Boddapati, H.K.; Bhatnagar, M.R.; Prakriya, S. Performance Analysis of Cluster-Based Multi-Hop Underlay CRNs Using Max-Link-Selection Protocol. *IEEE Trans. Cogn. Commun. Netw.* 2018, 4, 15–29. [CrossRef]
2. Senanayake, R.; Atapattu, S.; Evans, J.S.; Smith, P.J. Decentralized Relay Selection in Multi-User Multihop Decode-and-Forward Relay Networks. *IEEE Trans. Wirel. Commun.* 2018, 17, 3313–3326. [CrossRef]
3. Duy, T.; Kong, H. Secrecy Performance Analysis of Multihop Transmission Protocols in Cluster Networks. *Wirel. Pers. Commun.* 2015, 82, 2505–2518. [CrossRef]
4. Wang, X.; Su, Z.; Wang, G. Secure Connectivity Probability of Multi-hop Clustered Randomize-and-Forward Networks. *ETRI J.* 2017, 39, 729–736. [CrossRef]
5. Joung, J. Machine Learning-Based Antenna Selection in Wireless Communications. *IEEE Commun. Lett.* 2016, 20, 2241–2244. [CrossRef]
6. He, D.; Liu, C.; Quek, T.Q.S.; Wang, H. Transmit Antenna Selection in MIMO Wiretap Channels: A Machine Learning Approach. *IEEE Wirel. Commun. Lett.* 2018, 7, 634–637. [CrossRef]
7. Wang, X. Decision-Tree-Based Relay Selection in Dualhop Wireless Communications. *IEEE Trans. Veh. Technol.* 2019, 68, 6212–6216. [CrossRef]
8. Wang, X.; Liu, F. Data-Driven Relay Selection for Physical-Layer Security: A Decision Tree Approach. *IEEE Access* 2020, 8, 12105–12116. [CrossRef]
9. Nguyen, T.T.; Lee, J.H.; Nguyen, M.T.; Kim, Y.H. Machine Learning-Based Relay Selection for Secure Transmission in Multi-Hop DF Relay Networks. *Electronics* 2019, 8, 949. [CrossRef]
10. Fan, Z.; Gu, X.; Nie, S.; Chen, M. D2D power control based on supervised and unsupervised learning. In Proceedings of the 2017 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 558–563.
11. He, C.; Zhou, Y.; Qian, G.; Li, X.; Feng, D. Energy Efficient Power Allocation Based on Machine Learning Generated Clusters for Distributed Antenna Systems. *IEEE Access* 2019, 7, 59575–59584. [CrossRef]
12. Liang, L.; Ye, H.; Li, G.Y. Spectrum Sharing in Vehicular Networks Based on Multi-Agent Reinforcement Learning. *IEEE J. Sel. Areas Commun.* 2019, 37, 2282–2292. [CrossRef]
13. He, X.; Jiang, H.; Song, Y.; He, C.; Xiao, H. Routing Selection With Reinforcement Learning for Energy Harvesting Multi-Hop CRN. *IEEE Access* 2019, 7, 54435–54448. [CrossRef]
14. Huang, C.; Chen, G.; Gong, Y.; Xu, P. Deep reinforcement learning based relay selection in delay-constrained secure buffer-aided CRNs. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
15. Wang, X.; Jin, T.; Hu, L.; Qian, Z. Energy-Efficient Power Allocation and Q-Learning-Based Relay Selection for Relay-Aided D2D Communication. *IEEE Trans. Veh. Technol.* 2020, 69, 6452–6462. [CrossRef]
16. Zhao, B.; Ren, G.; Dong, X.; Zhang, H. Distributed Q-Learning Based Joint Relay Selection and Access Control Scheme for IoT-Oriented Satellite Terrestrial Relay Networks. *IEEE Commun. Lett.* 2021, 25, 1901–1905. [CrossRef]
17. Su, Y.; Lu, X.; Zhao, Y.; Huang, L.; Du, X. Cooperative Communications With Relay Selection Based on Deep Reinforcement Learning in Wireless Sensor Networks. *IEEE Sens. J.* 2019, 19, 9561–9569. [CrossRef]
18. Jadoon, M.A.; Kim, S. Relay selection algorithm for wireless cooperative networks: A learning-based approach. *IET Commun.* 2017, 11, 1061–1066. [CrossRef]
19. Watkins, C.; Dayan, P. Technical Note: Q-Learning. *Mach. Learn.* 1992, 8, 279–292. [CrossRef]