Enhancing 5G Network Slicing: Slice Isolation via Actor-Critic Reinforcement Learning with Optimal Graph Features

Amir Javadpour, Forough Ja’fari, Tarik Taleb, and Chafika Benzaid
Introduction

- Network slicing faces **two major challenges:**
  * covering the maximum number of requests
  * providing slice isolation

- **Slice isolation** is a mechanism for protecting the slices against DoS/DDoS attacks.

- None of the existing researches have considered slice isolation in the optimal mapping process.

- Our contributions:
  * introducing an actor-critic RL model that incorporates innovative features to isolate the slices effectively.
  * Devising a technique to decrease the environmental footprint of the RL model, thereby enhancing its efficiency.
  * Performing evaluations through a comparison with various RL models using four distinct evaluation metrics.
Schematic View of Our Solution

Considering both challenges

Dealing with none of the challenges

Dealing with the first challenge
Proposed Reinforcement Learning Model (SIRL)

- In our proposed model, Slice Isolation-based Reinforcement Learning (SIRL), we aim to map as many slice requests as possible while keeping their availability at an acceptable level.

- We have chosen actor-critic Reinforcement Learning as our learning model because the learning environment is too wide, and there is a need to limit the valid actions according to our constraints.

- The main strategies of SIRL:

  * Considering appropriate network features helping the agent find optimal mapping solutions that lead to both the maximum possible number of successfully mapped requests and the minimum possible DoS/DDoS damage.

  * Reducing the state size of the environment to improve the agent performance by ranking the features and make repeated states appear frequently.
Environment States (First Strategy)

- Our first strategy is to **represent the network with optimal features**.

- Five initial features are considered:

  * **F1**: The remained CPU capacity of each substrate node
  
  * **F2**: The sum of the adjacent links remained bandwidth of each substrate node
  
  * **F3**: The number of virtual nodes that are currently mapped on each substrate node
  
  * **F4**: A binary value for each substrate node that indicates whether it can be the host of the current virtual node
  
  * **F5**: The importance value of each substrate node
Environment States (Second Strategy)

- The second strategy of SIRL is to **optimize the state size**, or the number of possible states, to improve the learning performance. So that when a single SIRL model is trained, it can be used for different networks.

- Consider the beneath network with these CPUs: \{5, 4, 4, 10, 2, 3\}. Each substrate node may vary from 1 to 10, and hence, we may have 10^6 states. Our solution is to rank these values to reach: \{4, 2, 2, 5, 0, 1\}. Now, each substrate node may vary from 0 to 5, and we may have only 6^6 states.

```
Algorithm 1: The procedure of generating the set of final features in SIRL

Require: \( F_i \), the set of the values of the \( i^{th} \) initial feature
Ensure: \( F'_i \), the set of the values of the \( i^{th} \) final feature

\( S \leftarrow \) an empty array

for \( 1 \leq j \leq N \) do
    \( S \leftarrow S + \{j, F_i[j]\} \)

for \( 1 \leq j \leq N \) do
    \( \text{min} \leftarrow j \)

for \( j \leq k \leq N \) do
    if \( S[k][2] < S[\text{min}][2] \) then
        \( \text{min} \leftarrow k \)
    \( \text{temp} \leftarrow S[j] \)
    \( S[j] \leftarrow S[\text{min}] \)
    \( S[\text{min}] \leftarrow \text{temp} \)

\( F'_i \leftarrow \) an empty array

for \( 1 \leq j \leq N \) do
    \( F'_i \leftarrow F'_i + \{j - 1\} \)
    if \( j \neq 1 \) and \( S[j][2] = S[j - 1][2] \) then
        \( F'_i[j] \leftarrow F'_i[j - 1] \)

return \( F'_i \)
```
Reward Function

- A network's remaining resource capacity at time $t$:

$$\text{Re} = \sum_{i=1}^{N} \sum_{j=1}^{N} rlc(i, j)$$

- The maximum number of virtual nodes that are mapped on the substrate nodes:

$$\text{Ma} = \max_{1 \leq i \leq N} s_{mv}(i)$$

- The reward function is:

$$\text{Reward}(i, j) = \begin{cases} -\infty, & \text{If mapping is invalid} \\ \text{Re} - \text{Ma}, & \text{Else if } \alpha \\ \text{Re}, & \text{Otherwise} \end{cases}$$

---

**Algorithm 2** The procedure of training the agent in SIRL

**Require:** $S$, the substrate network
**Require:** $episodes$, the number of training episodes
**Ensure:** $ac$, the trained model

$ac \leftarrow$ initialize the actor-critic model

\[ \text{for } 1 \leq e \leq episodes \text{ do} \]

\[ \text{moves} \leftarrow \text{The number of virtual nodes} \]

\[ \text{move} \leftarrow 0 \]

\[ \text{while } \text{move} < \text{moves} \text{ do} \]

\[ \text{state} \leftarrow \text{the environment state from Algorithm 1} \]

\[ \text{action} \leftarrow \text{the optimal action from } ac \]

\[ s \leftarrow \text{action} \]

\[ v \leftarrow \text{move} \]

Map the $v^{th}$ virtual node on the $s^{th}$ substrate node

\[ \text{reward} \leftarrow \text{Reward}(s, v) \quad \triangleright \text{Equation 3} \]

Update $ac$ based on $state$, $action$, and $reward$

\[ \text{move} \leftarrow \text{move} + 1 \]

\[ \text{return } ac \]
Evaluation

- To evaluate the performance of SIRL, we have simulated different topologies in Python, and PyTorch is used for implementing the Reinforcement Learning models and training them.

- We have compared the performance of SIRL against nine of the existing Reinforcement Learning models.

- Eight different network scenarios are considered in the simulations, each of which has different graph features.

<table>
<thead>
<tr>
<th>Scenario name</th>
<th>Number of SNs</th>
<th>Number of SLs</th>
<th>Min. SN capacity</th>
<th>Max. SN capacity</th>
<th>Min. SL bandwidth</th>
<th>Max. SL bandwidth</th>
<th>Min. request VNs</th>
<th>Max. request VNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>4</td>
<td>6</td>
<td>9</td>
<td>15</td>
<td>5</td>
<td>8</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>5</td>
<td>7</td>
<td>2</td>
<td>5</td>
<td>12</td>
<td>20</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>6</td>
<td>9</td>
<td>2</td>
<td>6</td>
<td>12</td>
<td>41</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>Scenario 4</td>
<td>7</td>
<td>11</td>
<td>2</td>
<td>7</td>
<td>18</td>
<td>37</td>
<td>3</td>
<td>12</td>
</tr>
<tr>
<td>Scenario 5</td>
<td>8</td>
<td>14</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>36</td>
<td>5</td>
<td>11</td>
</tr>
<tr>
<td>Scenario 6</td>
<td>9</td>
<td>15</td>
<td>2</td>
<td>9</td>
<td>10</td>
<td>35</td>
<td>7</td>
<td>13</td>
</tr>
<tr>
<td>Scenario 7</td>
<td>10</td>
<td>16</td>
<td>2</td>
<td>10</td>
<td>9</td>
<td>36</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>Scenario 8</td>
<td>11</td>
<td>20</td>
<td>2</td>
<td>11</td>
<td>21</td>
<td>36</td>
<td>5</td>
<td>11</td>
</tr>
</tbody>
</table>
Requests Acceptance Ratio

- Acceptance ratio is the ratio of the number of successfully mapped requests to the total number of arrived requests.

- SIRL has increased the acceptance ratio by 70%.
Required Memory

- Required memory is the number of bits that are involved in presenting an environment state.

- SIRL has **reduced** the required memory by 95%.

![Graphs showing memory reduction across different scenarios](image-url)
DDoS Damage

- When the adversary performs a DoS/DDoS attack against the substrate nodes, all the mapped slices on that node will be affected.

- Since all the virtual nodes of a request must collaborate, losing one of them leads to the failure of the whole request.

- We have defined the DDoS damage as the ratio of the average number of requests that are affected by a DDoS attack to the total number of requests.

- SIRL has decreased the DDoS damage by 8%.
- The **solving time of SIRL** is 0.07 seconds higher than the average time of other models, and **35% lower** than MLRL.

- However, we can ignore it according to the satisfying results of the other metrics.
Conclusion

- This study presents a new method for efficiently managing 5G network requests while maintaining high acceptance rates and ensuring security.
- The SIRL actor-critic RL model uses innovative network features and simplifies the environment to improve performance compared to previous models.
- SIRL enhances acceptance rates and provides protection against DoS/DDoS attacks.

Future works

- To enhance the performance of mapping between virtual nodes on physical machine nodes, it is necessary to employ more intricate graphs such as triangular, square, and star graphs. This will lead to more resilient topological features.
- To achieve higher accuracy and speed in learning for intricate and vast networks, it is recommended to implement a multi-agent structure.
For further updates, visit us at
www.mosaic-lab.org