Mobile IPv4-in-UDP

Ian Hajra*
Brown University
Providence, Rhode Island, USA
ian_hajra@brown.edu

Wyatt Sieminski*
Brown University
Providence, Rhode Island, USA
wyatt_sieminski@brown.edu

ABSTRACT

Mobile IPv4-in-UDP is an extension of traditional IP-in-UDP systems that allows users to switch networks while maintaining their original IP address. This document details the development of a Mobile IPv4 system, built on the IP project from CSCI 1680, which developed a fully functional IP-in-UDP system. The features of Mobile IPv4-in-UDP include support for key components, including Mobile Nodes, Home Agents, Foreign Agents, and an IP-in-IP tunneling mechanism to facilitate mobility. Furthermore, the implementation provides *REPL* command support for the movement of Mobile Nodes within the network, registration with Care-of Addresses, and other essential infrastructure required to achieve RFC-compliant Mobile IPv4 while adhering to a IP-in-UDP link layer abstraction.

1 OVERVIEW

Mobile IPv4 is a protocol that provides support for nodes that move quickly between networks. It allows a device such as a mobile phone to quickly switch between WiFi and cellular data without needing to register a new IP address. It achieves this by providing the infrastructure for Mobile Nodes (`vhost`) to register with a Home Agent (`vrouter`). Upon moving networks, a Mobile Node will obtain a Care-of-Address (CoA), which will serve as a form of temporary IP address for its time registered with a Foreign Agent (different `vrouter`). Note that the Foreign Agent does not broadcast the Mobile Node using a protocol such as RIP. Based on **RFC 5944**, upon registration with the foreign agent, the Mobile Node will traditionally send a **Registration Request** to its Home Agent, indicating the CoA, its home IP address and other crucial information. The Home Agents will send a **Registration Response** in response, indicating whether the Mobile Node can register successfully. The Home Agent will maintain a table that stores CoA information from Registration Requests. To support interoperability with the IP-in-UDP system utilized in CSCI 1680, this report presents a slightly modified protocol through which the mobile node registration system was simplified into a sequence of *REPL* commands. We refer to this protocol as Mobile IPv4-In-UDP. Within the *REPL* commands, users specify the movement of Mobile Nodes, including new *UDP Address and Port* information to support the movement of Mobile Nodes within the IP-in-UDP framework. After the *REPL* command registration process, packets destined for a mobile node are encapsulated upon arrival with the Home Agent. These encapsulated packets will be tunneled using IP tunneling to the Foreign Agent, which can be found at the Mobile Node's Care-of Address. Packets received by the foreign agent with the *encapsulation protocol number* will be unencapsulated and sent locally to the mobile node. For a Mobile Node to return to the home network, another sequence of *REPL* commands must be executed to represent the change at the link layer level, in which case the network will go back to operating within the specifications of the IP/TCP project.

*Both authors contributed equally to this project.

2 LANGUAGE AND DEPENDENCIES

Mobile IPv4-in-UDP is a protocol that can be implemented in a wide variety of languages. This project opted for the most straightforward approach, based on our previous work.

2.1 Language

To build Mobile IPv4 into our custom IP/TCP stack, we used Rust. This was the programming language our IP and TCP projects were implemented with.

2.2 Libraries

In order to implement Mobile IPv4-in-UDP within our IP/TCP implementation, the following libraries were used, which follow directly from IP/TCP.

- **std:** used for general functions associated with rust programming, as well as synchronization and network structs that proved helpful.
- **etherparse:** used for serialization and creation of packet headers.
- **ipnet:** used for creation and functions available on the Ipv4Net struct.

3 PROGRAM DESIGN

Our system aims to emulate behavior of **RFC 5944**, and our program design follows the specification listed there, with minor adjustments to comply with the IP-in-UDP link layer abstraction.

3.1 Mobile Nodes

In the context of our IP and TCP projects, Mobile Nodes are restricted to `vhosts`. These hosts will have modified functionality, which still permits all behavior as specified in the previous projects. The new functionality includes a new REPL command:

- **Change Network (cn)**: All mobile node movement is triggered by the use of a *REPL* command, which follows the specification:
  - **cn** <mobile-udp-addr> <mobile-udp-port> <foreign-agent-udp-address> <foreign-agent-udp-port> <foreign-agent-vip>

  where the arguments are:
  - **mobile-udp-addr:** the the IP address the mobile node will use at the link layer level after moving networks (for the sake of this project 127.0.0.1).

  - **mobile-udp-port:** the udp port number the mobile node will use at the link layer level after moving networks (for the sake of this project, this is typically in the 5007-5015 range, depending on the network).
  - **foreign-agent-udp-address:** the IP address of the foreign agent's interface that the mobile node is connecting to at the link layer level.
  - **foreign-agent-udp-port:** the udp port of the foreign agent's interface that the mobile node is connecting to at the link layer level.
  - **foreign-agent-vip:** the virtual IP address of the foreign agent the mobile node is connecting to.

The execution of the **cn** REPL command serves to effectively move the host at the link layer level, and is a substitute for the **Agent Discovery** process outlined in **RFC 5944**, as well as the physical change in how the node communicates with the network. As a result, the Mobile Node's routing table will be changed to contain only the foreign agent and the default, which is now the foreign agent as well. However, rather than being listed as a *LOCAL* route within the table, the foreign agent will be recognized as a `MOBILE` route.

The list of interfaces and the list of neighbors for the node are also changed following movement to a new network. Interfaces associated with the `HOME` network are downed, and a new interface is created for connection with the foreign agent. The UDP information for this interface comes directly from the REPL command. The neighbors list is altered similarly to the interfaces list, with the addition of the foreign agent as a `MOBILE` neighbor.

The **cn** REPL command can also be used to move an already `MOBILE` node elsewhere within the Network. This can be to a new location or back to the home subnet. In this process, all previous `MOBILE` interfaces, neighbors, and routes are removed, and the default route is adjusted based on the new information specified within the REPL. As a result, returning to the home subnet requires the use of **cn** with all the UDP information from the original node and the home agent's original router connection information.

Sending from a `MOBILE` node functions similarly to when a node is `HOME`. The primary difference is that all packets get sent directly to the Foreign Agent and are forwarded around the network from there. This was done to reduce the scope of the **cn** REPL command. If local neighbors were to be included within the command for establishment on the linked layer, significant complication would be added for insufficient trade-off, as the foreign agent router will simply forward packets to neighbors on the subnet. It is important to note that packets sent from a `MOBILE` node contain the nodes `HOME` IP address as the source.

Receiving packets while a node is `MOBILE` is the exact same as when a node is `HOME` as the tunneling and packet unencapsulation behavior behavior is built into the Home and Foreign Agents.

3.2 Home Agents

In the context of our IP and TCP projects, the Home Agents are `vrouters`. These routers have modified functionality, which still permits all behavior as specified in the previous projects. The new functionality includes IP-in-IP encapsulation, and REPL commands that effectively serve as a substitute for the **Registration** and **De-Registration** process as outlined in **RFC 5944**:

- **Tunnel Packets (tp)**: When a mobile node changes network, the **CoA Registration** process involves the use of a REPL command, which follows the specification:
  - **tp** <virtual-ip> <care-of-address>

  where the arguments are:
  - **virtual-ip:** the IP address of the host changing networks.
  - **care-of-address:** the Care-of Address for the mobile node changing networks; this is the foreign agents virtual IP address (specifically the interface that the mobile node will be connecting to).

The execution of the **tp** REPL command logs the movement of a Mobile Node with a specified IP address and registers it with a CoA, which is also specified with the REPL. The introduction of this registration introduces a new data structure within the `Ip_Stack`, the `coa-table`. This is a hash map mapping virtual IP addresses to Care-of Addresses. Any packet the router receives or creates that is destined for a key within the `coa-table` is IP-in-IP encapsulated and sent within a new packet to the corresponding Care-of Address. The IP protocol number for the outermost packet is 4, which is specified as the IP-in-IP protocol number within `RFC 2003`. This process is the backbone of Mobile IPv4 and is a core component of the infrastructure for easy and quick movement of nodes between networks.

- **Delete Tunnel (dt)**: When a mobile node returns to the home network, or moves to a different mobile network, the **De-Registration** process involves the use of a REPL command, which follows the specification:
  - **tp** <virtual-ip>

  where the arguments are:
  - **virtual-ip:** the `HOME` IP address of the host changing networks.

The execution of the **dt** REPL command removes the Virtual IP - Care-of Address pair from the `coa-table` data structure. This removes the IP-in-IP encapsulation for packets destined for the specified virtual IP. This process is equivalent to the mobile node returning to the home network, and the router functions according to the IP/TCP project specification.

- **List Tunnels (lt)**: Throughout the lifetime of a `vrouter`, a `coa-table` is kept that contains the mappings of virtual IP addresses to Care-of Address for mobile nodes. The **lt** command prints this table to the terminal.

3.3 Foreign Agents

In the context of our IP and TCP projects, the Foreign Agents will be `vrouters`. All `vrouters` include functionality to be both a Home Agent and a Foreign Agent. For the Foreign Agent side, the functionality includes the unencapsolation and forwarding of packets with protocol number 4, and REPL commands that substitute for **Agent Discovery** and **Agent Solicitation** process which is specified in **RFC 5944**.

- **Add Hosts (ah)**: When a mobile node joins a Foreign Agent's network, a REPL command is used to substitute the

Foreign Agent Registration and **Authentication** system. It uses the following specification:
  - **ah** <virtual-ip> <mobile-udp-addr> <mobile-udp-port> <if-name>

where the arguments are:
  - **virtual-ip:** the virtual IP address of the mobile node.
  - **mobile-udp-addr:** the udp address for the mobile node joining the Foreign Agent's network at the link layer level (for the sake of this project 127.0.0.1).
  - **mobile-udp-port:** the udp port for the mobile node joining the Foreign Agent's newtork at the link layer level (for the sake of this project, this is typically in range 5007-5015, depending on the network).
  - **if-name:** the name of the interface for the `vrouter` that the mobile node will connect to.

The execution of the **ah** REPL command adds a host to the local network on the interface specified as an argument for the command. This effectively adds the mobile node to the subnet at the link layer level. This command also adds the mobile node as a neighbor with the *NodeLocation* `MOBILE`. Whenever a `vrouter` with Foreign Agent capabilities receives a packet with the IP-in-IP encapsulation protocol number 4, and its own IP as the destination IP address, the agent will unencapsulate the packet, and reconstruct the encapsulated header. Using the destination IP Address within the encapsulated header, the agent will look through its list of neighbors to find a `MOBILE` neighbor whose virtual IP address matches the destination in the header. The router will then forward the packet to the Mobile Agent. It is important to note that the Foreign Agents does not add the Mobile Agent to its routing/forwarding table. This is important because it will not advertise the Mobile Agents address using RIP, and any packet sent from the router will first be directed to the Home Agent, before being encapsulated and sent to the Foreign Agents address (CoA in this case).

- **Delete Host (dh)**: When a mobile node leaves a Foreign Agent's network, a REPL command informs the Agent of the node's movement. It follows the following specification:
  - **dh** <virtual-ip>

  where the arguments are:
  - **virtual-ip:** the virtual IP address of the mobile node.

The execution of the **dh** REPL command represents the movement of a mobile node off of the Foreign Agent's network. As a result the Foreign Agent will no longer have the Mobile Node as a neighbor. Foreign Agents will still unencapsulate packets with protocol number 4 and their own IP address, however, they will no longer forward them to the mobile node as it has moved locations.

4 FAULT TOLERANCE

As with many systems, achieving Fault Tolerance is a difficult prospect. The Mobile IPv4-In-UDP system is generally fault tolerant. It ensures that packets are well formed, REPL commands are well formed, and that things are only read from the network in appropriate circumstances. Throughout our testing, we observed 0 host, router, or network crashes as the result of poor input.

Note that our system does assume honest interactions (i.e. there is no exploitative behaviour from any other nodes). Additionally, our system assumes that REPL command values will be correct if they are well formed. Notably, this means that REPL commamnds must have viable UDP addresses and ports provided, as well as be run correctly in relation to one another for the system to function properly. The system shouldn't crash in circumstances where this isn't followed, however it may not perform to the specification due to error on the part of the user.

5 RUNNING THE NETWORK

Running the Mobile IPv4-In-UDP system can be done with all of the specified REPL commands above. For an example of the system running, please refer to our *Demo Video*. Please note that running the network requires well defined configuration files, and that the following information assumes such.

5.1 Starting the Network

To start running the network, you must first determine the number of hosts and routers that should be present. Afterwards, corresponding `.lnx` and `.json` files should be created. Please refer to the `linear_r2h2` folder for an example of this.

After all of the required config files have been created, all `vhosts` and `vrouters` should be run using them. The `util/vnet_run` command is particularly useful for doing this on more complicated networks.

5.2 Example Usage

This section on an Example Usage assumes that the `linear_r2h2` network is being used. This outline follows the exact behaviour that is demonstrated in the *Demo Video*. For other networks, inputs to the REPL commands would have different arguments, however the general command usage would still be the same.

We begin by bringing H1 from its home network to the far side of R2.

(1) `cn 127.0.0.1 5006 127.0.0.1 5004 10.2.0.1`: This command should be run on **H1**. By running this, **H1** now knows that it has switched to be in communication with **R2**, and also knows what UDP address and port to expect to send and receive from. **H1** has now switched its "physical" location.
(2) `tp 10.0.0.1 10.2.0.1`: This command should be run on **R1**. This informs **R1** that it is serving as the Home Agent, and should begin tunneling packets to **R2**.
(3) `ah 10.0.0.1 127.0.0.1 5006 if1`: This command should be run on **R2**. This informs **R2** that the Mobile Node has entered its network, and it will be prepared to unencapsulate and pass on any packets indented for **H1**.

At this stage, **H1** has now fully moved its "physical" location to a different subnet, and the rest of the network has adjusted accordingly. Packets can now be sent to and from **H1**, and they will follow the procedure outlined in **RFC 5944**. This can be done by using the `send` REPL command from any of the members of the network. Note that other REPL commands, such as `lr`, `ln`, and `lt` can be used to observe the various states of members of the network.

At this stage, a similar procedure can be followed to send **H1** to any other location on the network. Note that each old tunnel should

be removed via the `dt` REPL command as the Mobile Node changes location. To bring **H1** back to its home network, the following steps are followed:

(1) `cn 127.0.0.1 5000 127.0.0.1 5001 10.0.0.2`: This command should be run on **H1**. By running this, **H1** knows it has switched its "physical" location back to the home network.
(2) `dt 10.0.0.1`: This command should be run on **R1**. This informs **R1** that the Mobile Node **H1** has returned home.
(3) `rh 10.0.0.1`: This command should be run on **R2**. This informs **R2** that the Mobile Node has left its network, and the Foreign Agent then removes information associated with **H1**.

At this stage, the network is now reset to its initial condition. All functions, such as IP packets and TCP connections will function as though nothing had ever changed.

6 KNOWN ISSUES

As it stands, the Mobile IPv4-in-UDP system will properly forward any and all IP packets. The primary drawback of the current system is that the TCP functionality is not maintained.

6.1 Fixing TCP

When the `TCP_Stack` is initialized, a copy of the `IP_Stack` is passed in. With changes happening throughout the `IP_Stack` during the Mobile IP process (such as UDP addresses and neighbors changing), updates are not properly reflected within the `TCP_Stack` due to its copied nature.

To fix this issue, a direct reference would have to be replaced inside of the `TCP_Stack` in order to obtain access to the `IP_Stack`. This change is not conceptually significant, however requires significant refactoring of previous work and at the time of submission has not been addressed. We feel that it is outside of the scope of goals for this project.

7 FUTURE WORK

This project has served to address the portion of RFC 5944 that incorporates IP-in-IP encapsulation and forwarding. However, there are significant portions of RFC 5944 that have been abstracted into REPL commands. Continued work on this project should make strides towards the completion of an RFC 5944 implementation. This work can be divided into 3 major sections.

7.1 Registration Requests and Responses

The initial proposal for this project included the implementation of a **Registration Request** and **Registration Response** protocol for mobile nodes to register with a Home Agent. While this portion of **RFC 5944** would have fit well within the framework of Mobile IPv4 in UDP, we elected to forgo this portion of **RFC 5944** in place of *REPL* commands to better scope the project after receiving feedback on our proposal, and after our finals periods quickly became incredibly busy. This section serves as an overview of the way a **Registration Request** and **Registration Response** protocol implementation could replace the **tp** and **dp** *REPL* commands.

Upon movement to the Foreign Agent's network, the Mobile Node will send a **Registration Request** to its Home Agent, indicating the CoA, its home IP address. The Home Agents will send a **Registration Response** in response, indicating whether the Mobile Node can register successfully within the CoA table. Upon successful registration, the Home Agent will have all the information necessary to encapsulate and forward packets to the Mobile Agents, removing the need for a **tp** *REPL* command. Additionally, upon movement to a different network, the Mobile Node will initiate another **Registration Request/Response** handshake which will indicate the Mobile Node is no longer registered with that particular CoA. This will remove the need for the **dt** *REPL* command and conform more accurately to **RFC 5944**.

7.2 Agent Discovery

Agent Discovery was initially listed as one of the stretch goals for the project, and was not implemented as part of this work. An overview of the ideas and importance of Agent Discovery is given here.

Agent Discovery is an important part of the seamless transition between networks that is enabled by the implementation of **RFC 5944**. It provides support for Mobile Nodes to interpret their current location within the network. This is supported by Home Agents and Foreign Agents by the periodic broadcasting of **Agent Advertisements**. These messages include information that allows the Mobile Agents to determine whether or not the are on their home network, and the information necessary needed to register with both the Foreign and Home Agents. If Mobile Agents have not received an **Agent Advertisement** within a specified time period, it can actively request the information using an **Agent Solicitation** message, which will illicit an **Agent Advertisement** from nearby Host or Foreign Agents. The implementation of Agent Discovery is a crucial part of **RFC 5944** and Mobile IP. This would also remove the need for the **cn** *REPL* command in its current form, as the Mobile Node would be able to understand its movement within the network through **Agent Advertisements**. However, this process would be hard to implement within the IP-in-UDP framework given the necessity for link layer address and port numbers.

7.3 Foreign Agent Authentication

Foreign Agent Authentication is yet another aspect of **RFC 5944** that was listed as a stretch goal within the project proposal. Foreign Agent authentication requires that a Mobile Node clear an Authentication procedure with the Foreign Agent before being able to Register with their Home Agent using the Foreign Agent's CoA. Foreign Agent Authentication is an important part of ensuring that Mobile IP is a safe and secure protocol. The authentication of Mobile Nodes with a Foreign Agent's subnet is essential to real-world deployment and security, however, did not see any implementation within this project.

8 CONCLUSION

By implementing Mobile IPv4-In-UDP, this created a meaningful extension to our work in earlier projects for CSCI 1680. The unique

abstraction of IP-in-UDP allows for the network to be run on individual computers, while still acting as a fully functional network. Mobile IPv4-In-UDP presented a fascinating challenge, while allowing our network to be significantly more portable.