**Вежба 6:**

**Симулација на мрежен рутер**

* **Тежина: 5/10**

**Преглед**: Во оваа вежба, на поедноставен начин ќе ja симулирате работата на мрежен рутер. Фокусот е на манипулацијата со JSON, со акцент на правилно рутирање и препраќање на датаграмитие. Дури и да не ја разбирате во целост работата на еден мрежен рутер, задачата, сепак, преку детална низа од чекори, ќе се обиде на едноставен начин да ја објасни неговата функционалност.

**Инструкции**:

1. **Router-Settings.json:**

* Дадена ви е "router-settings.json" датотека.
* JSON структурата вклучува:
* Име на рутерот
* Влезен интерфејс со неговата IP адреса
* IP адреси на други интерфејси
* Табела за рутирање (мапирање на конкретни IP адреси во соодветна next hop IP адреса и соодветниот интерфејс од рутерот)
* ARP табела (мапирање на IP адреси во MAC адреси)

1. **Datagram.json**

* Покрај "router-settings.json" датотека, дадена ви е и datagram.json датотека
* "datagram.json" датотеката содржи низа на поедноставени IP датаграми.
* Секој датаграм содржи полиња како верзија, должина на заглавието, вкупна должина, време на живот (TTL), протокол, checksum, изворна IP адреса, дестинациска IP адреса и payload (содржина).

1. **Визуелна репрезентација:**

* Дадена ви е HTML страница која визуелно ги прикажува интерфејсите на рутерот. Влезниот интерфејс е истакнат со црвена боја.
* Потребно е да доделите IP адреси на сите интерфејси (редоследот не е битен, но влезниот интерфејс треба да биде оној којшто е обележан со црвена боја и се наоѓа лево од самиот рутер).
* Исто така, треба визуелно да се прикаже табелата за рутирање и ARP табелата, со нивните соодветни записи.

1. **Прикажување на Датаграми:**

* HTML страницата визуелно ги прикажува датаграмите поставени во влезниот интерфејс.
* Корисникот треба да може да ги набљудуваат редоследот и деталите за секој датаграм.

1. **Копче за симулирање на рутирање:**

* Копчето "Simulate Routing" предизвикува симулација на рутирање.
* По кликањето, еден датаграм во еден момент започнува да се рутира. Најпрво неговото време на живот (TTL) се намалува за еден.
* Потоа датаграмот се рутират според неговата дестинациска IP адреса, според табелата за рутирање на рутерот.
* Откако ќе се определи next hop IP адресата, се пребарува во ARP табелата да се најде соодветната MAC адреса на next hop IP адресата.

1. **Логика на рутирање:**

* За секој датаграм, дестинациската IP адреса се користи за да се најде next hop IP адресата во табелата за рутирање.
* Потоа се користи ARP табелата за да се добие MAC адресата на next hop IP адресата.
* Датаграмите се преместуваат на соодветниот излезен интерфејс со дополнителни информации за MAC адресата на next hop интерфејсот.

1. **Отфрлање на датаграми:**

* Ако времето на живот (ТТL) на датаграмот стане нула откако е намалено, тој се отфрла и се преместува во "Discarded Datagrams" таб-от.
* Отфрлените датаграми не треба да подлежат на понатамошно рутирање.

1. **FIFO Ред:**

* При рутирањето на датаграмите, доколку повеќе датаграми се рутираат на еден ист излезен интерфејс, тие се ставаат во ред според правилото прв влезен, прв излезен (FIFO).
* Симулацијата завршува кога сите датаграми се препратени (рутирани).

***ПОГЛЕДНЕТЕ ГО ПРИМЕР СЦЕНАРИОТО ЗА ДОПОЛНИТЕЛНО РАЗЈАСНУВАЊЕ!***

**Пример сценарио**:

Замислете сценарио каде рутерот со име "RouterX" со повеќе интерфејси треба да обработи влезни IP датаграми. Рутерот има интерфејси како "FastEthernet0/1", "FastEthernet0/2", "FastEthernet0/3" и " FastEthernet0/4." Влезниот интерфејс е " FastEthernet0/1", а неговата IP адреса е "192.168.1.1." Табелата за рутирање на рутерот и ARP табелата содржат соодветни записи потребни за процесот на рутирање.

|  |  |
| --- | --- |
| **Router-Settings.json:** | **Datagrams.json:** |
|  |  |

**Чекори на симулацијата**:

1. **Визуелна Репрезентација:**

* HTML страницата визуелно ги прикажува интерфејсите на рутерот.
* "FastEthernet0/1" интерфејсот е влезен, па е означен со црвена боја.
* IP адреси се доделуваат и на останатите интерфејси.
* Табелата за рутирање и ARP табелата се пополнуваат со нивните записи.

1. **Приказ на датаграми:**

* Датаграмите се прикажани визуелно подредени во влезниот интерфејс "FastEthernet0/1."

1. **Симулација на рутирање:**

* Кликањето на копчето "Simulate Routing" го намалува TTL-то и рутира датаграми според табелата за рутирање на рутерот и ARP табелата.
* MAC адресите се земаат од ARP табелата за секоја next hop IP адреса.
* Датаграмите се преместуваат на соодветниот интерфејс со дополнителни информации за MAC адресата.

1. **Отфрлање на датаграми:**

* Ако TTL на датаграм стигне до нула, тој се отфрла и се става во табот "Отфрлени."

1. **FIFO ред:**

* Ако постојат повеќе датаграми на излезниот интерфејс, тие се подредуваат според правилото FIFO.

1. **Заклучок:**

* Симулацијата завршува кога сите датаграми успешно се рутирани.

Ќе дадеме специфичен пример како рутирањето точно се прави за еден специфичен датаграм користејќи го дадениот сценарио и JSON фајлови. Земете го во предвид датаграмот:

Изворна IP адреса: 192.168.30.10

Дестинациска IP адреса: 192.168.1.10

TTL: 5

**Чекори на рутирање**:

1. **Пребаруваме го табелата за рутирање:**

* Користејќи ја дестинациската IP адреса (192.168.1.10), табелата за рутирање на RouterX, покажува дека next hop IP адреса за 192.168.1.10 e 192.168.2.2, преку интерфејсот " FastEthernet0/2."

1. **Пребарување во ARP табелата:**

* Ја бараме MAC адресата на next hop интерфејсот (192.168.2.2) во ARP табелата.
* Од записите во ARP табелата може да заклучиме дека MAC адреса на next hop интерфејсот (192.168.2.2) е " 00:1A:2B:3C:4D:5E".

1. **Рутирање на датаграмот:**

* Датаграмот сега се рутира на интерфејсот " FastEthernet0/2" со MAC адресата "00:1A:2B:3C:4D:5E."

1. **Намалување на TTL:**

* TTL на датаграмот се намалува за еден.

1. **Заклучок:**

* Датаграмот е успешно рутиран до својата цел преку интерфејсот " FastEthernet0/2" со соодветната MAC адреса. Овој процес се повторува за секој датаграм.

**ЗАБЕЛЕШКА**: Оваа вежба користи поедноставен пристап за образовни цели. Во реални сценарии, табелите за рутирање обично вклучуваат мрежни IP адреси, а не специфични IP адреси на хост. Мрежните IP адреси им овозможуваат на рутерите да рутираат сообраќај поефикасно со групирање на повеќе хостови под заеднички идентификатор на мрежа.

**Fetch From:**

**router-settings**: https://raw.githubusercontent.com/Itonkdong/JSON/main/router-settings.json

**datagrams.json**: https://raw.githubusercontent.com/Itonkdong/JSON/main/datagrams.json

A screenshot of a computer

Description automatically generated

Слика 1. Успешно доделени IP адреси на сите интерфејси, успешно вчитани табели и успешно подредени датаграми на влезниот интерфејс од рутерот.

A screenshot of a computer

Description automatically generated

Слика 2. Успешно рутирање на првиот датаграм од влезниот интерфејс.

A screenshot of a computer

Description automatically generated

Слика 3. Успешно завршена симулацијата на рутирање со правилно подредување на датаграмите на соодвените излезни интерфејси. Датаграмите со TTL = 0 се отфрлени

**Exercise 6:**

**Network Router Simulation**

* **Difficulty Level: 5/10**

**Overview:** In this exercise, you will engage in a simulated network router scenario to explore the intricacies of packet routing. The focus is on JSON manipulation, emphasizing the routing decisions and packet forwarding processes. Even if you lack a deep understanding of networking, this exercise aims to provide a comprehensive experience.

**Exercise Steps**:

1. **Router-Settings.json:**
   * You will need to fetch information from JSON file called "router-settings.json."
   * The JSON structure includes:
     + Router hostname
     + Entering interface with its IP
     + IPs of other interfaces
     + Routing table (mapping concrete IP addresses to next hop IP addresses and interface identifications)
     + ARP table (mapping IP addresses to MAC addresses)
2. **Understanding Datagram.json:**
   * You then need to examine a JSON file named "datagram.json" containing an array of simplified IP datagrams.
   * Each datagram object includes fields such as version, header length, total length, time to live, protocol, header checksum, source IP, destination IP, and payload.
3. **Visual Representation:**
   * A provided HTML page visually represents router interfaces. The entering interface is highlighted in red.
   * You need to assign IP addresses to all interfaces (order is not crucial, but the entering interface should be the red one, on the left).
   * You also need to visually show the routing table and the arp table, with their respective entries.
4. **Displaying Datagrams:**
   * The HTML page displays datagrams visually queued up in the entering interface.
   * Students can observe the order and details of each datagram.
5. **Simulate Routing Button:**
   * A "Simulate Routing" button triggers the routing simulation.
   * Upon clicking, one datagram at a time is being routed. His TTL is decreased by one.
   * Datagrams is then routed based on destination IP addresses, following the router's routing table.
   * The ARP table is consulted to find MAC addresses for the next hop IP addresses.
6. **Routing Logic:**
   * For each datagram, the destination IP is used to find the next hop IP in the routing table.
   * The ARP table is then used to obtain the MAC address of the next hop IP.
   * Datagrams are moved to the appropriate interface with additional information about the MAC address.
7. **Discarding Datagrams:**
   * If a datagram's TTL becomes zero after decrementation, it is discarded and moved to the "Discarded" tab.
   * Discarded datagrams should not undergo further routing.
8. **FIFO Queue:**
   * After successfully routing all datagrams, if there are more datagrams on an exiting interface, they are queued up following the First In, First Out (FIFO) rule.
   * The simulation concludes when all datagrams are routed and queued if necessary.

***SEE THE EXAMPLE SCENARIO DOWN BELOW FOR MORE CLARIFICATION!***

**Example Scenario: Router Simulation Exercise**

Consider a scenario where a router named "RouterX" with multiple interfaces needs to process incoming IP datagrams. The router has interfaces like "FastEthernet0/1", "FastEthernet0/2", "FastEthernet0/3" и " FastEthernet0/4." The entering interface is " FastEthernet0/1," and its IP address is "192.168.1.1." The router's routing table and ARP table contain entries necessary for routing decisions.

|  |  |
| --- | --- |
| **Router-Settings.json:** | **Datagrams.json:** |
|  |  |

**Simulation Steps:**

1. **Visual Representation:**
   * The HTML page displays the router's interfaces visually, with " FastEthernet0/1" highlighted in red.
   * IP addresses are assigned to all interfaces.
   * Routing table and the arp table are filled with their entries.
2. **Datagram Display:**
   * Datagrams are shown visually queued up in the " FastEthernet0/1" entering interface.
3. **Routing Simulation:**
   * Clicking the "Simulate Routing" button decreases TTLs and routes datagrams based on the router's routing table and ARP table.
   * MAC addresses are fetched from the ARP table for each next hop IP.
   * Datagrams are moved to the appropriate interface with MAC address information.
4. **Discarding Datagrams:**
   * If a datagram's TTL reaches zero, it is discarded and placed in the "Discarded" tab.
5. **FIFO Queue:**
   * If more datagrams exist on an exiting interface, they are queued up following the FIFO rule.
6. **Conclusion:**
   * The simulation concludes when all datagrams are successfully routed, discarded if necessary, and queued if additional datagrams are present.

Let's walk through a specific example of how the routing is exactly done for one specific datagram using the provided scenario and JSON files.

Consider the datagram:

Source IP: 192.168.30.10

Destination IP: 192.168.1.10

TTL: 5

**Routing Steps:**

1. **Routing Table Lookup:**
   * Look up the destination IP (192.168.1.10) in the routing table of RouterX.
   * The routing table entry indicates that the next hop for 192.168.1.10 is 192.168.2.2, through " FastEthernet0/2."
2. **ARP Table Lookup:**
   * Look up the MAC address of the next hop IP (192.168.2.2) in the ARP table.
   * The ARP table entry provides the MAC address "00:1A:2B:3C:4D:5E" for the next hop IP.
3. **Routing the Datagram:**
   * The datagram is now routed to the interface " FastEthernet0/2" with the MAC address "00:1A:2B:3C:4D:5E."
4. **Decrement TTL:**
   * The TTL of the datagram is decreased by one.
5. **Conclusion:**
   * The example datagram has been successfully routed to its destination through the " FastEthernet0/2" interface with the corresponding MAC address.

This process is repeated for each datagram, ensuring they are correctly routed based on the router's routing table and ARP table entries. If a datagram's TTL reaches zero, it is discarded and not further routed. Additionally, any remaining datagrams on an exiting interface are queued up following the FIFO rule after successful routing.

**NOTE:** *This excerise uses a simplified approach for educational purposes. In real-world scenarios, routing tables typically include network IPs rather than specific host IPs. Network IPs allow routers to route traffic more efficiently by grouping multiple hosts under a common network identifier.*

**Fetch From:**

**router-settings**: https://raw.githubusercontent.com/Itonkdong/JSON/main/router-settings.json

**datagrams.json**: https://raw.githubusercontent.com/Itonkdong/JSON/main/datagrams.json