網路總整 HW2

Hosts' function

```python
def update_arp(self, dstMac, dstIp):
    # update ARP table with a new entry
    if not dstMac in self.arp_table:
        self.arp_table[dstIp] = dstMac
```

當 destination 不在 ARP table 中，就會在 ARP table 中新增此 host 的 ip 跟 mac (在 handle packet 收到 ARP request 或 reply 才會進來這個 function)

```python
def handle_packet(self, srcPort, message):
    # handle incoming packets
    if message[0] == "ARP request":
        if message[3] == sel
            # print(self.nam
            self.update_arp(message[2], message[4])
            message = ["ARP reply", message[2], self.mac, message[4], self.ip]
            self.send(message)
    elif message[0] == "ARP reply":
        # print(self.name + ": this is an ARP reply message")
        self.update_arp(message[2], message[4])
        message = ["ICMP request", message[2], self.mac,  message[4], self.ip]
        self.send(message)
    elif message[0] == "ICMP request":
        # print(self.name + ": this is an ICMP request message")
        message = ["ICMP reply", message[2], self.mac,  message[4], self.ip]
        self.send(message)
    # else:
        # print(self.name + ": this is an ICMP reply message")
```

(parameter) message: Any

message 中包含四個 variable，**封包種類**、**destination MAC address**、**source MAC address**、**source IP address**

- 當封包種類為 ARP request 時，更新 ARP table 並回傳 ARP reply(message[0] 為 ARP reply)
- 當封包種類為 ARP reply 時，更新 ARP table 並回傳 ICMP request，
- 當封包種類為 ICMP request 時，回傳 ICMP reply
- 當封包種類為 ICMP reply 時，不回傳東西

```
def ping(self, dst_ip):
    # handle a ping request
    if dst_ip in self.arp_table:
        for host in self.arp_table:
            if dst_ip == host:
                message = ["ICMP request", self.arp_table[host], self.mac, dst_ip, self.ip]
    else:
        message = ["ARP request", "ffff", self.mac, dst_ip, self.ip]

    self.send(message)
```

當一個 host 要 ping 另一個 host 時，先查詢 ARP table 是否有 dst host IP 對 MAC 的資訊

- 若有，則直接發送 ICMP request
- 反之，則發送 ARP request

```
def send(self, message):
    # determine the destination MAC here
    '''
        Hint :
            if the packet is the type of arp request, destination MAC would be 'ffff'.
            else, check up the arp table.
    '''
    node = self.port_to # get node connected to this host
    node.handle_packet(self, message) # send packet to the connected node
```

在相對應的 port 送出封包(也只有一個 port)

Switches' function

```
def update_mac(self, srcPort, macAddr):
    if not macAddr in self.mac_table:
        for i in range(self.port_n):
            if self.port_to[i] == srcPort:
                self.mac_table[macAddr] = i
```

如果 MAC address 沒有在 mac table 中，就會更新

```
def send(self, idx, message):
    node = self.port_to[idx]
    node.handle_packet(self, message)
```

在指定的 port 送出封包，封包內容都一樣

```python
def handle_packet(self, srcPort, message):
    # handle incoming packets
    self.update_mac(srcPort, message[2])
    dstMac = message[1]
    if dstMac == "ffff":
        for i in range(self.port_n):
            if self.port_to[i] != srcPort:
                self.send(i, message)
    else:
        if dstMac in self.mac_table:
                self.send(self.mac_table[dstMac], message)
        else:
            for i in range(self.port_n):
                if self.port_to[i] != srcPort:
                    self.send(i, message)
```

- 收到封包時先更新 MAC table
- 若收到的 MAC address 是 ffff，則此封包為 ARP request，直接網所有 port (source port 以外) 送出
- 若不是 ffff，則查看 MAC table，沒有查到對應的 port 則網所有 port 送