CS-542 PROJECT

(Fall 2017)

ARP Protocol Design

Submitted to:

Dr. Edward Chlebus

Submitted by:-

Animesh Patni (A20403240)

Ashish Ryot (A20405230)

Tapishwar Singh Wazir (A20410804)

1. The ARP protocol uses the next-hop address and the interface number, found in the routing table, to find the physical address of the next hop. Consider a certain router R_i whose routing table has only three columns: mask, network address, and next-hop address, i.e. the column with the interface numbers has been deleted. **(Note that only the column with the info on the interfaces has been removed not the interfaces themselves, i.e. the network configuration has not changed.)** Does the ARP protocol still work? If not, how to modify this protocol to make it work? Does your modification affect the network performance?

Solution:

The ARP protocol uses the next-hop address and the interface number found in the routing table to find the physical address of the next hop.
We consider a certain router R_i whose routing table has only three columns, i.e. mask, network address and next-hop address, but no column with the interface number.
Since the ARP protocol needs the IP address and the "Interface Number" through which a particular network is connected to a multi-homed host such as a router, we would need to modify the ARP protocol a bit to make it work when the information about the interface is not provided.

Instead of broadcasting the ARP request message through a particular Interface Number to all the hosts in that network in a regular setting of the ARP protocol, we could modify the protocol into broadcasting the request message to all the networks to which the router R_i is connected by iterating through all the interfaces of the router. The destination IP address recognizes its logical address in the ARP request message and sends back a unicast reply to the router R_i. The router then recognizes the interface from which the reply to the broadcast for a certain IP address was received from and uses that interface number to forward the packet.
This modification can slow down the networks by increasing unnecessary traffic in the (n-1) networks connected to the router (n is the total number of networks to which a router is connected through interfaces). This modification can also slow down the working of the ARP protocol because it has to iterate through all the networks to which the router is connected.

Time Complexity, T(n), of the ARP Protocol when the interface number is given:

T(n)=O(1)	*It takes constant time for the ARP protocol when the interface number is given, because in that case we only need to broadcast the request to a single network via a single interface number.*

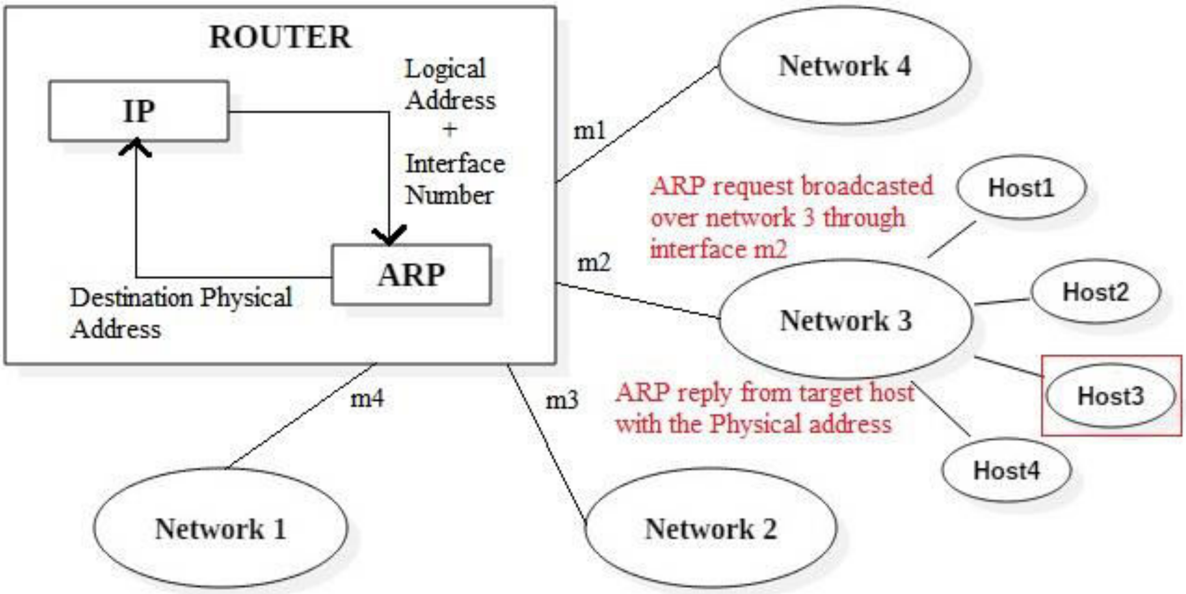

Figure 1: *ARP Protocol when interface number is given*

Time Complexity, T(n), of the ARP Protocol when the interface number is not given:

$T(n)=O(n)$ | *It takes time proportional to the number of interfaces that the router uses to connect to the various networks, because it has to iterate through all the interfaces to broadcast the ARP request to find the required destination physical address.*

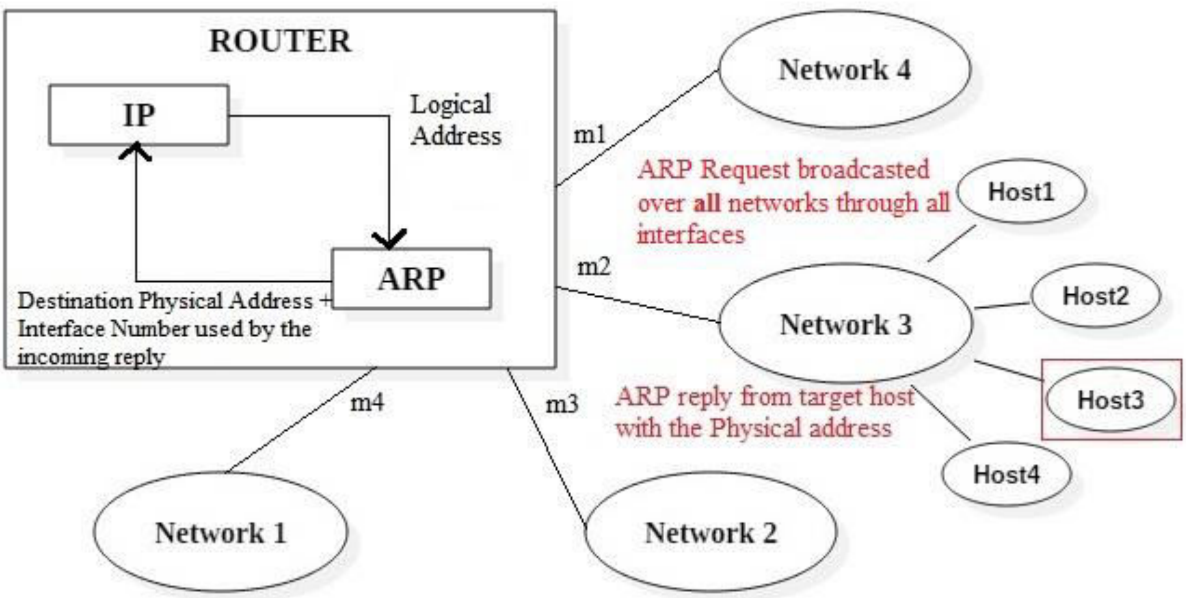

Figure 2: *Modified ARP Protocol when interface number is not given*

However, the performance of our modified network could be improved by introducing a new improvised routing table that we start to fill after receiving the first reply from the iterations.
In this way, we could update our new table with an extra column of the interface number which we can update by checking through which interface was the reply received for an IP address.
This new routing table could be filled completely after initial few runs of the modified ARP protocol and after that the running time of the protocol would be reduced from *O(n)* to *O(1), (Since we would refer this new table for the interface number corresponding to an IP address).*

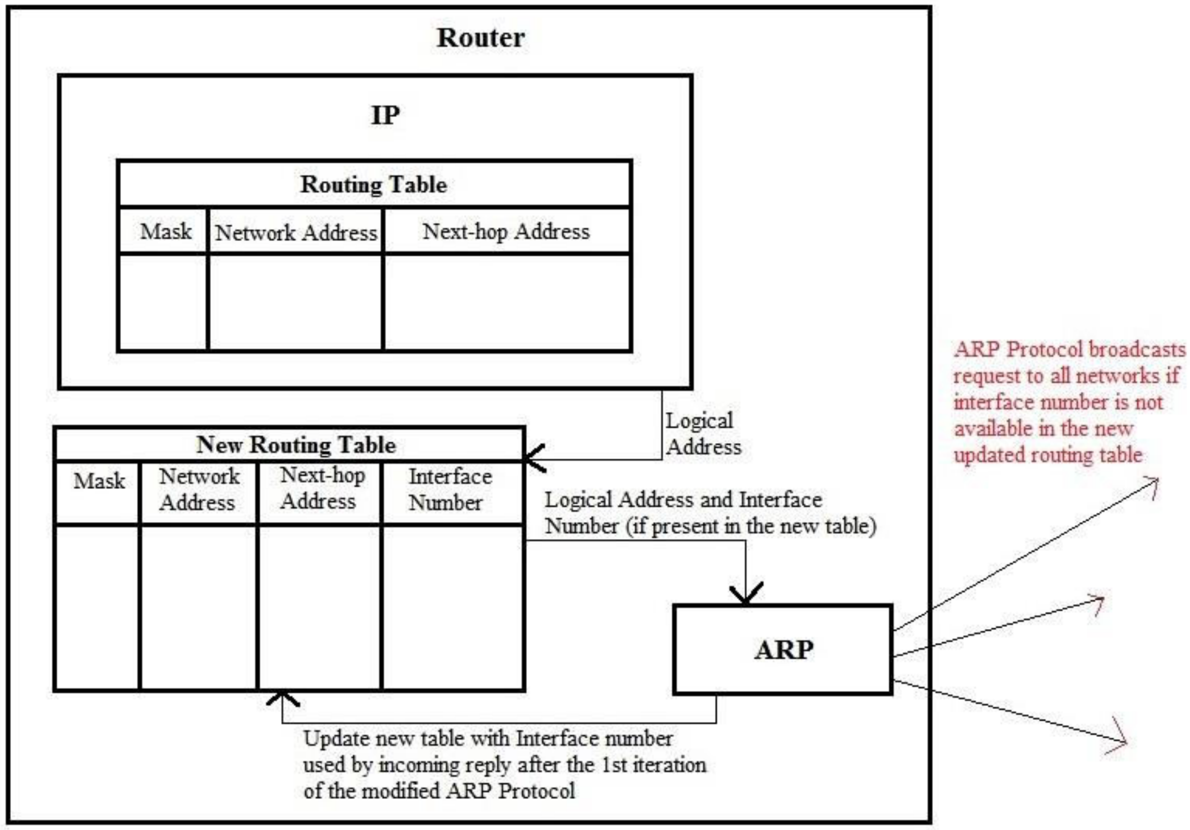

Figure 3: *Modified ARP Protocol with an additional Routing table with the column for interface number*

2. The ARP reply is used to update the ARP cache table. Can the ARP request be also used to update this table? If so, how?

Solution:

Yes, the ARP request can also be used to update the ARP cache table.

Method 1:

The host machine sends out an ARP broadcast request to get the physical address of the destination IP address to all the hosts connected to that network. Normally, these requests are discarded by the hosts connected to the network whose IP address does not match the target IP address of the ARP request. But we can modify the ARP protocol, so that the devices, when they get an ARP request broadcast message can update their ARP cache table with the protocol address and the physical address of the sender. i.e. on receiving the broadcasted ARP request, all the devices in that network or connected to that interface can update their cache table with the Protocol address and the Physical address of the broadcaster.

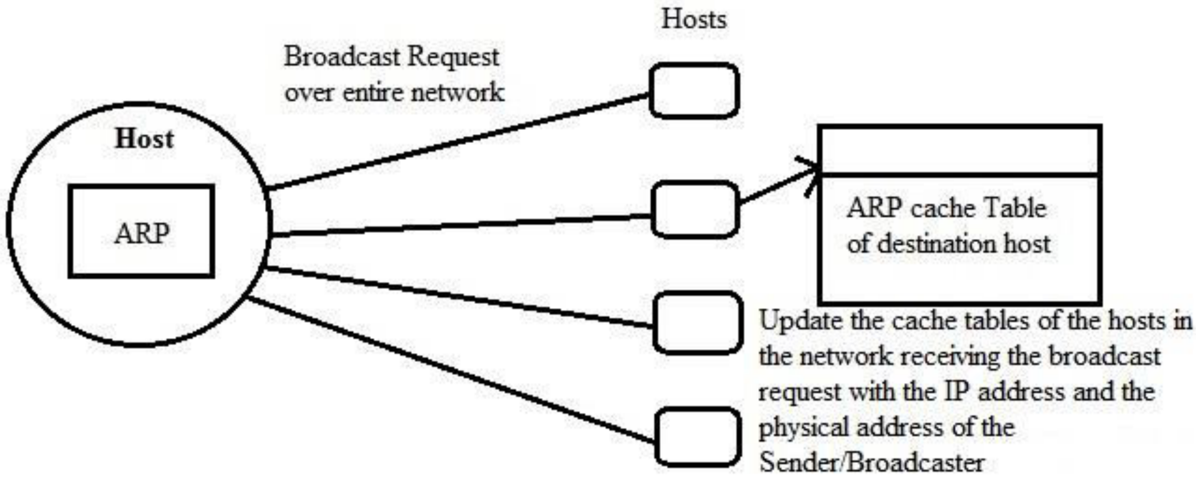

Figure 4: *Updating cache table by using requests as stated in Method 1*

Method 2:

Another way to update the ARP cache table with the requests is: All the hosts connected to a network can broadcast their ARP request every few minutes and the receivers of these requests then can update their ARP cache tables in order to keep them updated. But there is a down side to this implementation of updating the cache table with the requests broadcasted by all the hosts in the network in every few minutes. This implementation can unnecessarily increase the traffic on the network during some intervals. To reduce that traffic, we could set the time out value for the cache table entries in such a way that we do not have to resend broadcast requests very often.

For example: time out could be set to 30 minutes, and so the devices would only need to send out ARP requests every 30 minutes instead of broadcasting more frequently and increasing traffic.

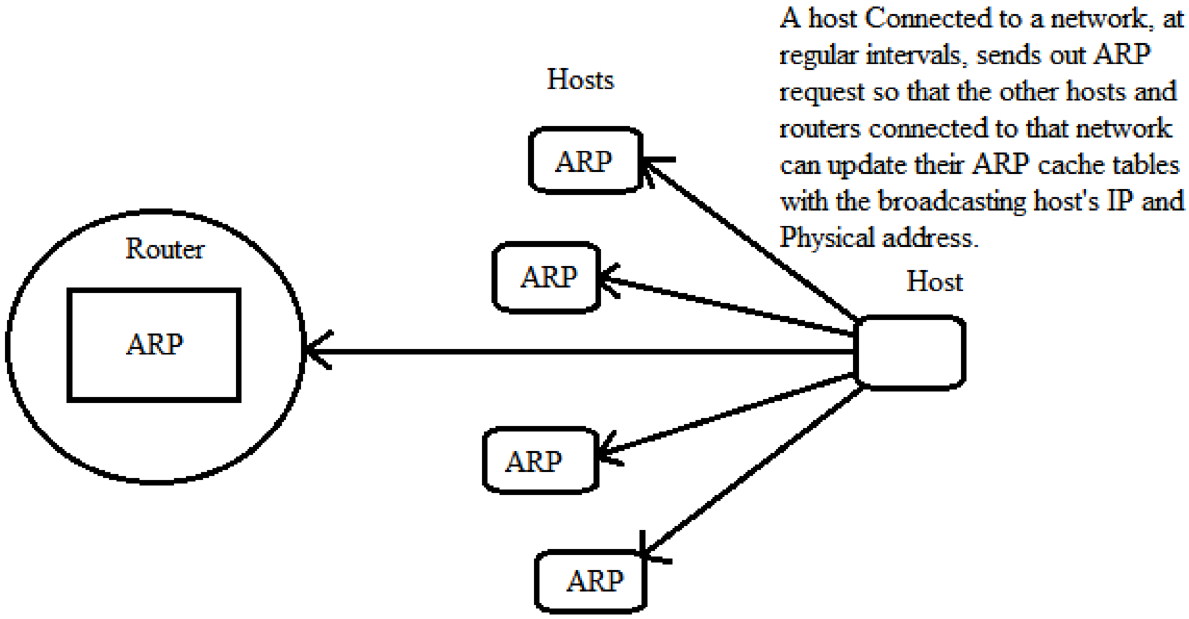

Figure 5: *Updating ARP cache table by using requests as stated in Method 2*

3. Two network administrators argue about setting an optimal time-out *T* for the ARP cache table of a certain router. The first administrator thinks that *T=t1* is satisfactory, the other one claims that *T=t2* is much better. Propose a set of criteria to evaluate which of them is right. (Maybe both are wrong?) Which of these criteria is the most important from your point of view?

<u>Solution:</u>

To set an optimal time-out value T, for the ARP cache table of a certain router, we need to consider the following cases:

Case 1:

The rate at which the ARP output module receives the IP datagrams with different destination addresses is higher than the rate at which the states in the ARP cache table are getting "FREE".

The ARP output module receives large number of IP datagrams with different destination addresses as compared to the number of "Free" states available in the ARP cache table.

i.e.

Rate of ARP module receiving IP datagrams with different destination addresses	>	*Rate of states changing from "Resolved" or "Pending" to "FREE"*

In this case the network administrator should not set the time-out value for the ARP cache table very high because then it would delay the IP datagrams from reaching their destinations because of the lack of "free" states available in the ARP cache table.

But also, the time out value should not be so less that the resolved state times out before all the IP datagrams present in the queue for a certain IP address have been sent.

Case 2:

The rate at which the ARP output module receives the IP datagrams with different destination addresses is much less than the rate at which the states are getting "FREE" in the ARP cache table.

i.e.

Rate of ARP module receiving IP datagrams with different destination addresses	<	*Rate of states changing from "Resolved" or "Pending" to "FREE"*

In this case, the network administrators should set the time put value higher than that of the previous case in order to avoid unnecessary rework and traffic in the network.

Case 3:
If the rate of the states getting FREE in the ARP table is nearly similar to the ARP output module receiving datagrams, then the administrators could set the timeout value somewhere between the timeout values for case 1 and case 2.

Case 4:
Another criterion that we can consider to set the time-out value for the ARP cache table is the number of entries (IP datagrams) present in the queue allotted to a certain IP address belonging to the RESOLVED state.
Depending on the number of entries in the queue, the network administrator can set the timeout value to change dynamically.
For example:
(i) If the number of entries in the Queue is more, then the time-out value for that entry in the ARP cache table should be set to a longer duration so that that entry does not time out before all the IP datagrams have been sent to the corresponding physical address.
(ii) If the number of entries in the Queue is very less, then the time-out value for that destination IP address in the cache table should be reduced to a smaller duration, so that the state of that row in the ARP cache table could be changed from RESOLVED to FREE so that it becomes available for newer entries.

Based on the above cases/criteria, the network administrators can set the time-out value to suit the particular scenario of their router.

According to us, the 4th case, where the number of entries in the queue is considered to set the time-out value for a particular entry in the cache table is important, because it does not fix the time-out value to a fixed duration but can alter that duration by checking the size of the queue corresponding to a target IP address. This way, an IP address that needs the connection for longer duration to receive the large number of datagrams is not timed out early, gets longer time-out duration. Whereas the IP address to which only a few datagrams are to be sent does not unnecessarily keep other entries waiting for their turn, gets shorter time-out duration.