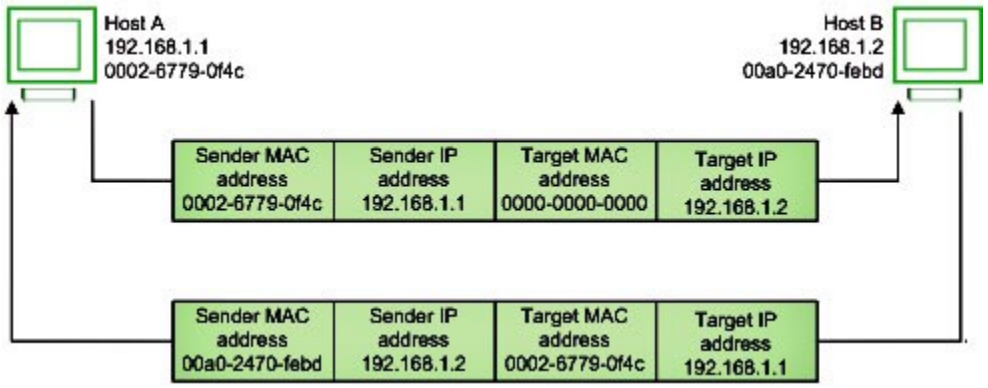
Address Resolution Protocol

By Michael Almandeel

Address resolution protocol, also known as ARP, is a set of data and behaviors that supports the mapping of logical(ip) addresses to physical(mac) addresses. In order to physically transmit data , regardless of the medium, a physical address for the recipient of that data is required(unless said transmission is a broadcast). Since applications tend to favor the use of ip addresses for sending data, we need to have a way to identify the physical addresses associated with said ip addresses in order to physically send the data to the target host. One of the solutions to this problem is ARP.

The set of data and behaviors which comprise ARP are the ARP cache, the ARP request, and the ARP reply. The ARP cache stores the ip to mac address mappings, as well as an age associated with each mapping. Entries are added to a machine's ARP cache when they receive an ARP request aimed at them(in which case they add the sender to the cache and send back an Arp reply) , or when they receive an ARP reply aimed at them. When adding a mapping to the cache, the age of that mapping is set to zero, and is incremented as time increases. When the age of a mapping is greater than some predefined maximum age, it will be removed from the cache. It is important to note that if a machine sees either an ARP request or reply, regardless of who its aimed at, and they have the mapping of the mac and ip addresses in their ARP cache, they will reset the age of that mapping to zero.

Both ARP requests and ARP replies are structured as an ARP message, which is typically 28 bytes in length. ARP messages contain an operation field, which indicates whether the message is a request or a reply. They also have fields for both the sender's and target's hardware(mac) addresses and ip addresses. These addresses, as noted above, are used to populate the ARP cache when the messages are received.



Typical use of the ARP protocol is as follows. Host A wishes to send data to another host with the ip address 192.168.1.2 . Host A generates an ARP message, and sets the sender hardware address to its own mac address, the sender ip address to its own ip address, and the target ip address to 192.168.1.2 . It also sets the operation field to request. It then sends the message on its network interface , and a short time later, Host B receives the message. Host B inspects the message and determines that it is the target, because the target ip address matches its own ip address. If it currently does not have a mapping of the sender addresses in its ARP cache, it adds the mapping to the cache and sets the age to zero. If it does already have the mapping, it sets the age to zero. Either way, Host B generates its own ARP message, but

this time with the operation field set to reply. It sets the target address fields equal to the sender address fields of the previous corresponding request, and it sets the sender address fields to its own mac and ip address. It then sends this message on its network interface, and a short time later Host A receives it. Host A sees that its own addresses match the target addresses, and adds the mapping of B's addresses to the ARP cache. As usual, when a new entry is added to the cache, its age is set to zero. Now that Host A has both addresses for Host B, A is free to send ip packets to B using the physical network, and vice versa.