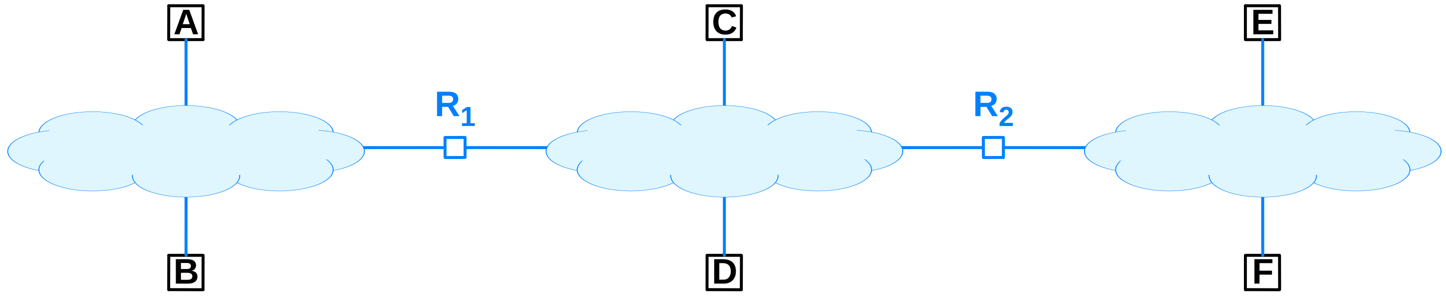
ADDRESS RESOLUTION PROTOCOL (ARP)

Protocol Addresses and Packet Delivery

- IP addresses are understood and maintained by software only
- Neither LAN nor WAN hardware understands IP addressing scheme; in another words, network hardware does not know how to locate a computer from its IP address
- At the same time, a frame transmitted across a physical network must contain the hardware address of the destination
- Consequently, the protocol software must translate the IP address of the destination computer into equivalent hardware address before it can send a packet across a physical network

Address Resolution

- The translation from a computer's IP address to an equivalent hardware address is known as *address resolution*
- Address resolution is local to a network;
  in another words, a host, or a router, uses address resolution when it needs to send a packet to another host, or router, on the same physical network
- A computer never resolves the address of a computer on a remote network
- If the destination is not located within the local network, the sender's software
  - Determines the next-hop address
  - Resolves the address of this next-hop



Address Resolution Techniques

- Address resolution algorithms can be grouped into three basic categories:
  - Table lookup
  - Closed-form computation
  - Message exchange
- The selection between these algorithms depends on the protocol and the hardware addressing schemes
- A computer that attaches to multiple networks, e.g., a router or a multi-homed host, may need more than one address translation module

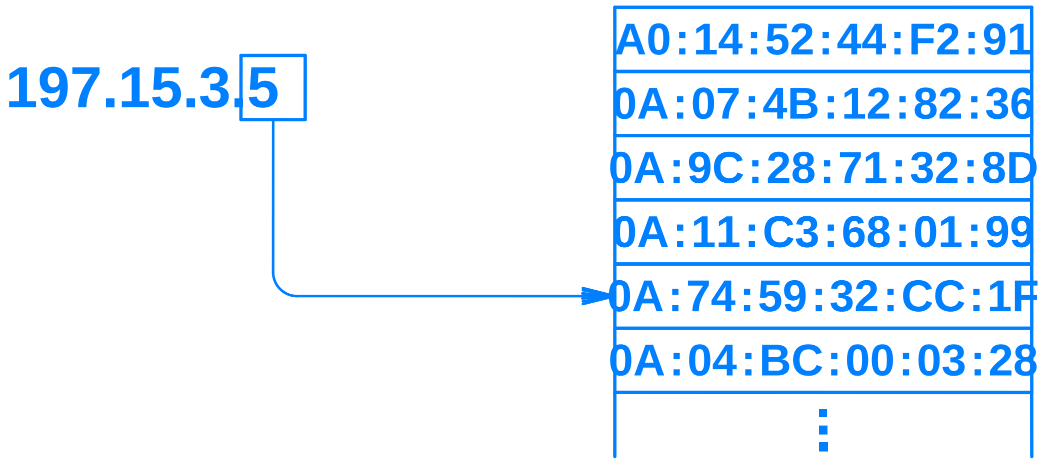
Address Resolution With Table Lookup

- The approach requires a data structure that contains information about address resolving
- The table consists of an array, where each array entry contains a pair of
  - A protocol address, and
  - Its equivalent hardware address

IP Address	Hardware Address
197.15.3.2	0A:07:4B:12:82:36
197.15.3.3	0A:9C:71:28:32:8D
197.15.3.4	0A:11:68:C3:01:99
197.15.3.5	0A:74:32:59:CC:1F
197.15.3.6	0A:04:00:BC:03:28
197.15.3.7	0A:77:0E:81:52:FA

- Since all entries inside a binding table are related to one physical network, all IP addresses in a given table have the same address prefix

- Implementations can save space by omitting the prefix from table entries
- The searching scheme can be
  - Sequential search, usually used with small networks
  - Hashing
  - Direct indexing, the IP address suffix is used as an index into the array



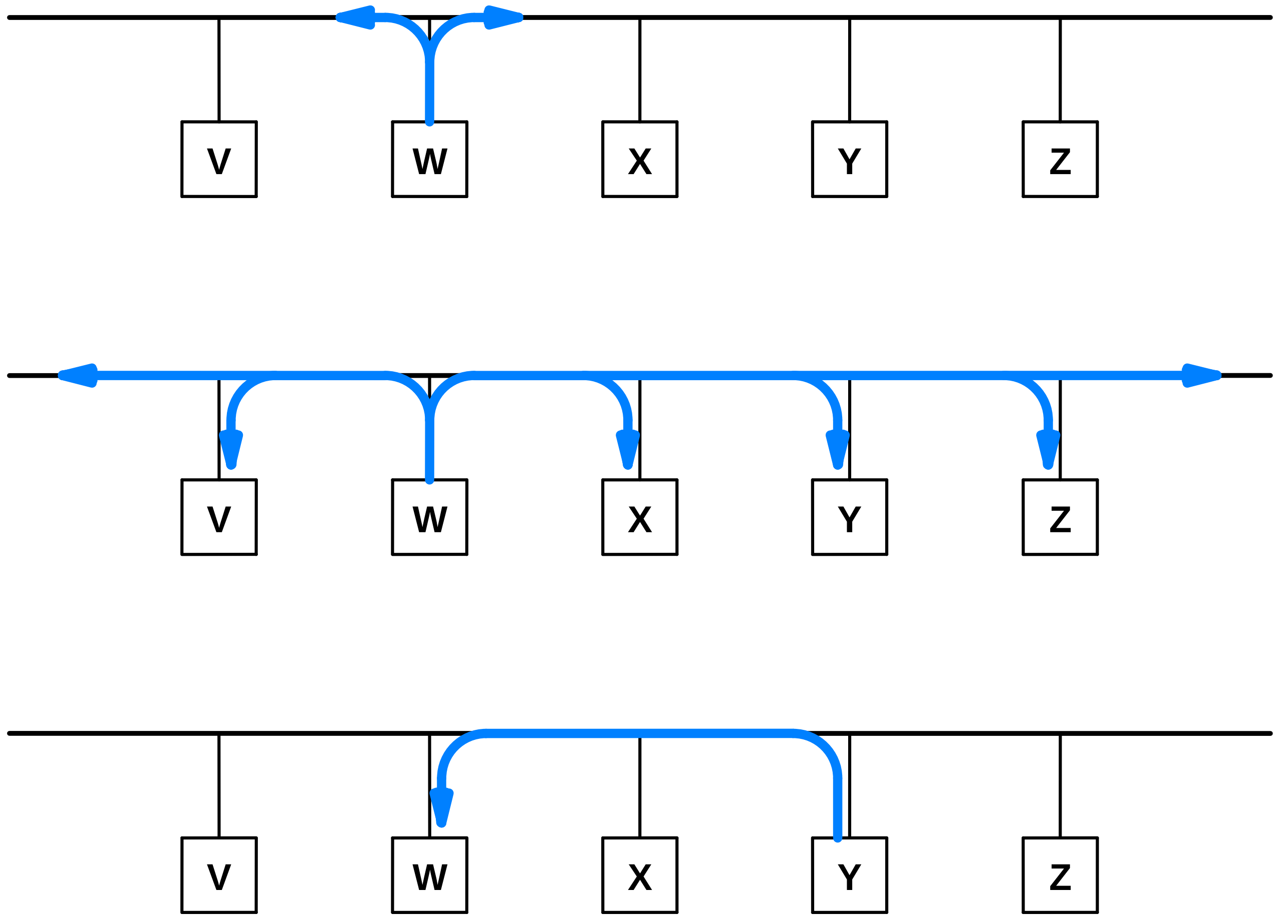
- Advantages
  - The table can store the address bindings for an arbitrary set of computers on a given network
  - Straightforward
  - Easy to program

Address Resolution With Closed-form Computation

- Suitable for networks that use configurable addressing, where the network administrator must choose a hardware address as well as an IP address for each host
- In this case, the two values can be chosen to make address resolution trivial
- Example: In a class C network, the hardware address can be selected to match the IP address suffix
- Advantages
  - Trivial to program
  - Computationally efficient
  - Does not require a table of values to be maintained

Address Resolution With Message Exchange

- A computer that needs to resolve an address
  - Sends a message across a network, i.e, an IP address
  - Receives a reply, i.e., the corresponding hardware address
- But, where should an address resolution *request* be sent?
  - To a one, or more, of the special address resolution servers until it finds an active server and receives a reply *(centralized solution)*
  - To all computers in the network simultaneously, i.e., broadcasting the request *(distributed solution)*
    - No special address resolution servers are needed
    - Each computer on the network participates in address resolution, by agreeing to answer resolution requests for its address
      - All machines receive the request and examine the requested address
      - If an incoming request matches a computer's IP address, the computer responds to the sender only



- Usually used on LAN hardware that has static addressing

Comparison of Address Resolution Schemes

- Table lookup
  - Useful with any hardware
  - Any address change affects all hosts
  - IP address is independent from hardware address
  - Produces resolution with minimum delay
- Closed-form computation
  - Hardware address must be smaller than IP address
  - IP address is determined by hardware address
  - Produces resolution with minimum delay
- Message exchange
  - IP address is independent from hardware address
  - May need hardware broadcast
  - Adds traffic to a network
  - Implementation is more difficult

The Usage of Address Resolution Schemes

- TCP/IP can use any of the three resolution methods
- The scheme chosen for a particular network depends on the addressing scheme used by the underlying hardware
  - *Table lookup*: usually employed to resolve IP addresses across a WAN
  - *Closed-form computation*: is used with configurable networks
  - *Message exchange*: is used on LAN hardware that has static addressing

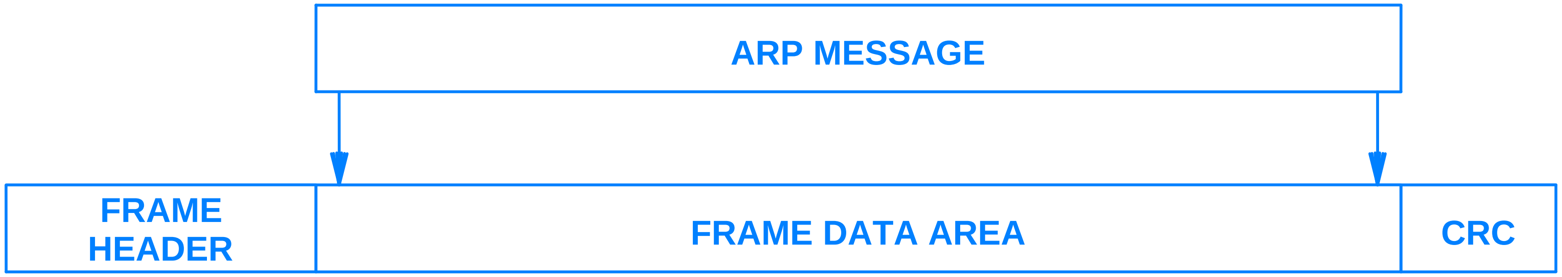
ARP Message Format

0	8	16	24 31
HARDWARE ADDRESS TYPE		PROTOCOL ADDRESS TYPE	
HADDR LEN	PADDR LEN	OPERATION	
SENDER HADDR (first 4 octets)			
SENDER HADDR (last 2 octets)		SENDER PADDR (first 2 octets)	
SENDER PADDR (last 2 octets)		TARGET HADDR (first 2 octets)	
TARGET HADDR (last 4 octets)			
TARGET PADDR (all 4 octets)			

The format for an ARP message when used to bind IP addresses to Ethernet hardware address

- Variable size hardware/protocol address fields
  - New network technology might be invented that have addresses with different sizes

Sending an ARP Message



An ARP message encapsulated in an Ethernet frame

- An ARP message travels inside a hardware frame (*encapsulation*)
- The ARP message is treated as data being transported
- The network hardware
  - Does not know about the ARP message format
  - Does not examine the contents of individual fields

Identifying ARP Frames

- The *type field* in the frame header specifies that the frame contains an ARP message
- A sender must assign the appropriate value to the type field before transmitting the frame
- The receiver must examine the type field in each incoming frame

Dest. Address	Source Address	Frame Type	Data In Frame
		806	complete ARP message

The *type field* in an Ethernet header used to specify the frame content
A value of 0x806 informs the receiver that the frame contains an ARP message

- Because Ethernet assigns a single type value to ARP, a receiver must examine the operation field in the message to determine whether an incoming message is a *request* or a *response*

A Problem With Message Exchange Scheme

- When computer A has a packet to deliver to computer B:
  - Computer A first broadcasts an ARP *request* to find computer B's hardware address
  - Computer B sends a *reply* to computer A with its hardware address
  - Computer A delivers the original packet to computer B
- Since most computer communication involves a sequence of packets, computer A is likely to repeat the exchange many time
- Sending a request for each binding is hopelessly inefficient

A Solution for the Message Exchange Traffic Overhead

- To reduce network traffic, ARP software extracts and saves/updates the information from ARP response/request messages
- These information is maintained in a small table in memory
- ARP manages this table as a *cache*
  - An entry is replaced whenever a response arrive
  - The oldest entry is removed whenever
    * The table runs out of space, or
    * After an entry has been updated for a long period of time
- Whenever ARP performs address binding, it searches the cache before sending an ARP *request*

Caching Message Exchange Information

- When an ARP message arrives, the receiver
  - Extracts the sender's address binding
  - Checks to see if a binding is present in the cache
    - If so, the receiver uses the binding in the incoming ARP message to update the previously stored binding
  - Examines the *operation* field of the message to determine whether the message is a *request* or a *response*
    - If it is a *response*, the receiver must have previously issued a *request* and is waiting for a binding
      - The receiver adds the binding to its cache for later use
    - If it is a *request*, the receiver compares field *target protocol address* with the local protocol address
      - If the two are identical, the computer must send an ARP response

Why Caching Message Exchange Information?

- Most computer communication involves two-way traffic
  - When a computer sends an ARP request, it must have a message to deliver
  - It is likely that once the packet has been delivered, a reply packet will be sent back
  - If the receiver does not have an address binding for the sender, it will need to broadcast an ARP request
  - Having a receiver to extract the sender's binding from the incoming ARP request eliminates the need for a later ARP request
- Maintaining a comprehensive list for every binding address in the network is not always a good idea; WHY
  - Waste of CPU time and memory
  - It is unlikely that every pairs of computers will communicate

Address Resolution Features

- Address resolution software
  - Hides the details of physical addressing
  - Allows software in higher layers to use protocol addressing
- Note that: address resolution is a function associated with the network interface layer
- The ARP scheme might represent a security hole, *HOW?*

