```java
import java.util.HashMap;
import java.util.Map;

class ARPCache {
    private Map<String, String> cache;

    public ARPCache() {
        this.cache = new HashMap<>();
    }

    public void addToCache(String ipAddress, String macAddress) {
        cache.put(ipAddress, macAddress);
    }

    public String getMacAddress(String ipAddress) {
        return cache.get(ipAddress);
    }
}

class ARPSimulator {
    private ARPCache arpCache;

    public ARPSimulator(ARPCache arpCache) {
        this.arpCache = arpCache;
    }

    public void simulateARP(String ipAddress) {
        String macAddress = getMacAddressFromARP(ipAddress);
        System.out.println("MAC Address for IP " + ipAddress + ": " + macAddress);
    }

    private String getMacAddressFromARP(String ipAddress) {
        String macAddress = arpCache.getMacAddress(ipAddress);

        if (macAddress == null) {
            macAddress = generateRandomMAC();
            arpCache.addToCache(ipAddress, macAddress);
        }
        return macAddress;
    }

    private String generateRandomMAC() {
        return "00:1A:2B:3C:4D:5E";
    }
}

public class ARPSimulation {
    public static void main(String[] args) {
        ARPCache arpCache = new ARPCache();
        ARPSimulator arpSimulator = new ARPSimulator(arpCache);

        arpSimulator.simulateARP("192.168.1.1");
        arpSimulator.simulateARP("192.168.1.2");
        arpSimulator.simulateARP("192.168.1.1");
    }
}
```

```java
import java.util.HashMap;
import java.util.Map;

class RARPCache {
  private Map<String, String> cache;

  public RARPCache() {
    this.cache = new HashMap<>();
  }

  public void addToCache(String macAddress, String ipAddress) {
    cache.put(macAddress, ipAddress);
  }

  public String getIPAddress(String macAddress) {
    return cache.get(macAddress);
  }
}

class RARPSimulator {
  private RARPCache rarpCache;

  public RARPSimulator(RARPCache rarpCache) {
    this.rarpCache = rarpCache;
  }

  public void simulateRARP(String macAddress) {
    String ipAddress = getIPAddressFromRARP(macAddress);
    System.out.println("IP Address for MAC " + macAddress + ": " + ipAddress);
  }

  private String getIPAddressFromRARP(String macAddress) {
    String ipAddress = rarpCache.getIPAddress(macAddress);

    if (ipAddress == null) {
      ipAddress = generateRandomIP();
      rarpCache.addToCache(macAddress, ipAddress);
    }
    return ipAddress;
  }

  private String generateRandomIP() {
    return "192.168.1.1";
  }
}

public class RARPSimulation {
  public static void main(String[] args) {
    RARPCache rarpCache = new RARPCache();
    RARPSimulator rarpSimulator = new RARPSimulator(rarpCache);

    rarpSimulator.simulateRARP("00:1A:2B:3C:4D:5E");
    rarpSimulator.simulateRARP("00:1A:2B:3C:4D:5F");
    rarpSimulator.simulateRARP("00:1A:2B:3C:4D:5E");
  }
}
```

```
set ns [new Simulator]
set nf [open trace1.tr w]
$ns trace-all $nf
set nr [open out1.nam w]
$ns namtrace-all $nr

proc ϐinish {} {
global ns nf nr
$ns ϐlush-trace
close $nf
close $nr
exec nam out1.nam
exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns duplex-link $n0 $n2 2Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 3Mb 10ms DropTail

set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
set tcpsink0 [new Agent/TCPSink]
$ns attach-agent $n3 $tcpsink0
set tcpsink1 [new Agent/TCPSink]
$ns attach-agent $n3 $tcpsink1

$ns connect $tcp0 $tcpsink0
$ns connect $tcp1 $tcpsink1
$ns at 1.0 "$ftp0 start"
$ns at 1.1 "$ftp1 start"
$ns at 10.0 "$ftp0 stop"
$ns at 10.1 "$ftp1 stop"
$ns at 10.2 "finish"
```

```
set ns [new Simulator]
$ns rtproto DV
set nf [open dv1.tr w]
$ns trace-all $nf
set nr [open dv2.nam w]
$ns namtrace-all $nr
proc ϐinish {} {
global ns nf nr
$ns ϐlush-trace
close $nf
close $nr
exec nam dv2.nam
exit 0}
for {set i 0} {$i<12} {incr i} {
set n$i [$ns node]}
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link $n3 $n4 1Mb 10ms DropTail
$ns duplex-link $n4 $n5 1Mb 10ms DropTail
$ns duplex-link $n5 $n6 1Mb 10ms DropTail
$ns duplex-link $n6 $n7 1Mb 10ms DropTail
$ns duplex-link $n7 $n8 1Mb 10ms DropTail
$ns duplex-link $n8 $n0 1Mb 10ms DropTail
$ns duplex-link $n0 $n9 1Mb 10ms DropTail
$ns duplex-link $n1 $n10 1Mb 10ms DropTail
$ns duplex-link $n9 $n11 1Mb 10ms DropTail
$ns duplex-link $n10 $n11 1Mb 10ms DropTail
$ns duplex-link $n11 $n5 1Mb 10ms DropTail
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Trafϐic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n5 $null0
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Trafϐic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
set null1 [new Agent/Null]
$ns attach-agent $n5 $null1
$ns connect $udp0 $null0
$ns connect $udp1 $null1
$ns rtmodel-at 10.0 down $n11 $n5
$ns rtmodel-at 30.0 up $n11 $n5
$ns rtmodel-at 15.0 down $n7 $n6
$ns rtmodel-at 20.0 up $n7 $n6
$ns at .1 "$cbr1 start"
$ns at .2 "$cbr0 start"
$ns at 45.0 "$cbr1 stop"
$ns at 45.1 "$cbr0 stop"
$ns at 50.0 "ϐinish"
$ns run
```

```
set ns [new Simulator]
$ns rtproto LS
set nf [open ls1.tr w]
$ns trace-all $nf
set nr [open ls2.nam w]
$ns namtrace-all $nr
proc ϐinish {} {
global ns nf nr
$ns ϐlush-trace
close $nf
close $nr
exec nam ls2.nam
exit 0}
for {set i 0} {$i<12} {incr i} {
set n$i [$ns node]}
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n3 1Mb 10ms DropTail
$ns duplex-link $n3 $n4 1Mb 10ms DropTail
$ns duplex-link $n4 $n5 1Mb 10ms DropTail
$ns duplex-link $n5 $n6 1Mb 10ms DropTail
$ns duplex-link $n6 $n7 1Mb 10ms DropTail
$ns duplex-link $n7 $n8 1Mb 10ms DropTail
$ns duplex-link $n8 $n0 1Mb 10ms DropTail
$ns duplex-link $n0 $n9 1Mb 10ms DropTail
$ns duplex-link $n1 $n10 1Mb 10ms DropTail
$ns duplex-link $n9 $n11 1Mb 10ms DropTail
$ns duplex-link $n10 $n11 1Mb 10ms DropTail
$ns duplex-link $n11 $n5 1Mb 10ms DropTail
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
set cbr0 [new Application/Trafϐic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 500
$cbr0 set interval_ 0.005
set null0 [new Agent/Null]
$ns attach-agent $n5 $null0
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Trafϐic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ 500
$cbr1 set interval_ 0.005
set null1 [new Agent/Null]
$ns attach-agent $n5 $null1
$ns connect $udp0 $null0
$ns connect $udp1 $null1
$ns rtmodel-at 10.0 down $n11 $n5
$ns rtmodel-at 30.0 up $n11 $n5
$ns rtmodel-at 15.0 down $n7 $n6
$ns rtmodel-at 20.0 up $n7 $n6
$ns at .1 "$cbr1 start"
$ns at .2 "$cbr0 start"
$ns at 45.0 "$cbr1 stop"
$ns at 45.1 "$cbr0 stop"
$ns at 50.0 "ϐinish"
$ns run
```