```java
 1 public boolean routeEvent(EventHolder event, SocketAddress fromAdr)
 2 {
 3   RoutingEntry entry;
 4   ModeHolder modeHolder;
 5   int variableId = event.m_event[0].getInt(Constants.EVENT_VARIABLE_ID_OFFSET);
 6   long timeStamp = event.m_event[0].getLong(Constants.EVENT_CREATED_OFFSET);
 7
 8   // Do we route this variable ?
 9   if ((entry = (RoutingEntry)this.m_tbl.get(variableId)) == null)
10     return false;
11
12   // Do we route this variable in the current operating mode ?
13   if ((modeHolder = entry.m_modeTbl[this.m_currentMode]) == null)
14     return false;
15
16   // Do we flood the event ?
17   if (modeHolder.m_flooding > 0)
18   {
19     ........
20     return true;
21   }
22
23   // Route this event
24   size = modeHolder.m_closeHolders.size();
25
26   // Set the expected ref count for this event
27   event.incrementRef(size);
28
29   for (int i = 0; i < size; ++i)
30   {
31     routingHolder = (RoutingHolderClose) modeHolder.m_closeHolders.get(i);
32
33     // Should we forward this event on this Event Channel ?
34     for (int j = 0; j < routingHolder.m_subIntervals.m_size; ++j)
35     {
36       if ((timeStamp + entry.m_pubIntervalHalf) %
37           routingHolder.m_subIntervals.m_elements[j] < entry.m_pubInterval)
38       {
39         // Is this an alert ?
40         if (routingHolder.m_priority == CommConstants.PRIORITY_ALERT)
41         {
42           if (routingHolder.m_outInterface.pushAlertEvent(event))
43             ++sentTo;
44         } // Send the event with its priority
45         else if (routingHolder.m_outInterface.pushEvent(event,routingHolder.m_priority))
46           ++sentTo;
47
48         // Terminate the loop
49         j = routingHolder.m_subIntervals.m_size;
50       }
51     }
52   }
53
54   // Decrement the reference for all the ECs that the event was not sent on
55   if ((size - sentTo) > 0)
56   {
57     if (event.decrementRef(size - sentTo) == 0)
58       return false; // This event must be recycled
59   }
60 }
```