

```

1 def maintainViews():
2     for viewType in [viewOverlay, viewPSS]: # For each view, do the same
3         peer[viewType] = None
4         if time to maintain viewType: # This viewType needs to be updated
5             peer[viewType] = selectPeer(viewType)          # Select a peer
6             links = selectLinks(viewType, peer[viewType])   # Select links
7             sendTo(peer[viewType], Request[viewType], links) # Send links asynchronously
8
9     while True:
10        block = (peer[viewOverlay] != None) or (peer[viewPSS] != None)
11        sender, msgType, msgData = recvFromAny(block) # Block if expecting something
12
13        if msg == None: # All work has been done, simply return from the call
14            return
15
16        for viewType in [viewOverlay, viewPSS]: # For each view, do the same
17            if msgType == Response[viewType]: # Response to previously sent links
18                updateOwnView(viewType, msgData) # Just update the own view
19
20            elif msgType == Request[viewType]: # Request for exchanging links
21                if peer[viewType] == None:      # No outstanding exchange request
22                    links = selectLinks(viewType, sender)    # Select links
23                    sendTo(sender, Response[viewType], links) # Send them asynchronously
24                    updateOwnView(viewType, msgData)         # Update own view
25                else: # This node already has a pending exchange request, ignore this one
26                    sendTo(sender, IgnoreRequest[viewType])
27
28            elif msgType == IgnoreRequest[viewType]: # Request has been denied, give up
29                peer[viewType] = None

```