Global Tool Scheduling Algorithm

We need to create an Algorithm for IEPM where Monitoring sites can schedule one at a time. If site A runs tool x, then no other site should execute x at that time.We can solve this problem using a combination of two basic Algorithms i.e. Token Ring and Priority Queue. Token Ring algorithm is use to allow one host at a time to execute a tool. There is one token, passing through the sites. The site that have the token will be able to execute the tools, while other will wait for the token to receive. The token will contain the number of bytes the site can use to execute tools. At site, after execution it takes the following actions.

1. Execute the tool with highest priority if bytes received in token is greater then bytes required by the tool else execute the second highest priority tool.
2. Set the priority of executed tool to the lowest priority.
3. Decrement the number of bytes used by the tool from the token.
4. if bytes in token are greater then any tool required byes then go to step 1 else set the token to its initial value and forward to the next site.

Lets say we have an Array of ***N*** Monitoring sites ***M[1...N]*** and each site have ***n*** tools ***T[1..n]*** to execute. Let ***B*** be the bytes received by host in token and ***X [1..x]*** represent no of target to be probed with tools.

```
NextSite(A,i)
if i is equal to length_of(A) then
   return A[1]
else return A[i+1]
```

```
ChangePriority(A){
  x = A[lengthOf(A)]
  for i = lengthOf(A)-1 down to 1
        A[i+1] = A[i]
        i = i -1
  A[1] = x
}
```

ChangePriority takes an array as argument and change its priority from N to 1 and increase the other consecutive elements priority by 1. It actually treats an index of an element as its priority. Hence, It copy A[N] at A[1] and A[i] at A[i+1]

```
GetMaxPriorityElement(A){
        return A[lengthOf(A)]
}
```

```
Schedule(M,T,X){
  while isTokenRecieved() is false
         // wait 1 sec for another check
  totalBytes = B                              // B is the total no. of bytes received in Token
  target = GetHighestPriorityElement(X)
  ChangePriority(X)
  while totalBytes > 0
        tool = GetHighestPriorityElement(T)
        exec (tool,target)
        totalBytes = totalbytes - tool.bytesRequired
        ChangePriority(T)
  sendToken(NextSite(M,i),B)
}
```