def simulate\_download\_from\_time(self, time, size):

        cum\_time = 0

        timeseg = self.get\_current\_timesegment(time)

        while(1):

            #find when next bw change is

            next\_set = None

            try:

                next\_set = self.bwlist[ self.bwlist.index(timeseg) + 1 ]

#是的，Python 的列表（list）确实支持使用 .index(element) 这种用法来查找元素在列表中的位置。

#.index(element) 方法的功能

#作用：.index(element) 方法用于返回指定元素在列表中第一次出现的索引位置。

#语法：

#list.index(element, start, end)

#element：要查找的元素。

#start（可选）：查找的起始位置（包含该位置）。

#end（可选）：查找的结束位置（不包含该位置）。

#返回值：如果列表中找到了这个 element，返回它的索引（即它在列表中的位置，从 0 开始）。

#错误：如果没有找到该元素，Python 会抛出一个 ValueError，表示该元素不在列表中。

            except( IndexError ):

                pass

            #if no next bw change, calculate the remaining time

            if not next\_set:

                cum\_time += size / (timeseg[1] / 8)

                return cum\_time

            #find time remaining on current bw, drain download by corresponding amt

            down\_time = next\_set[0] - time

            cum\_time += down\_time

            size -= down\_time \* (timeseg[1] / 8)

            if size <= 0:

                #refund unused time

                unused\_time = -1 \* size / (timeseg[1] / 8)

                cum\_time -= unused\_time

                return cum\_time

            timeseg = next\_set

            time = timeseg[0]