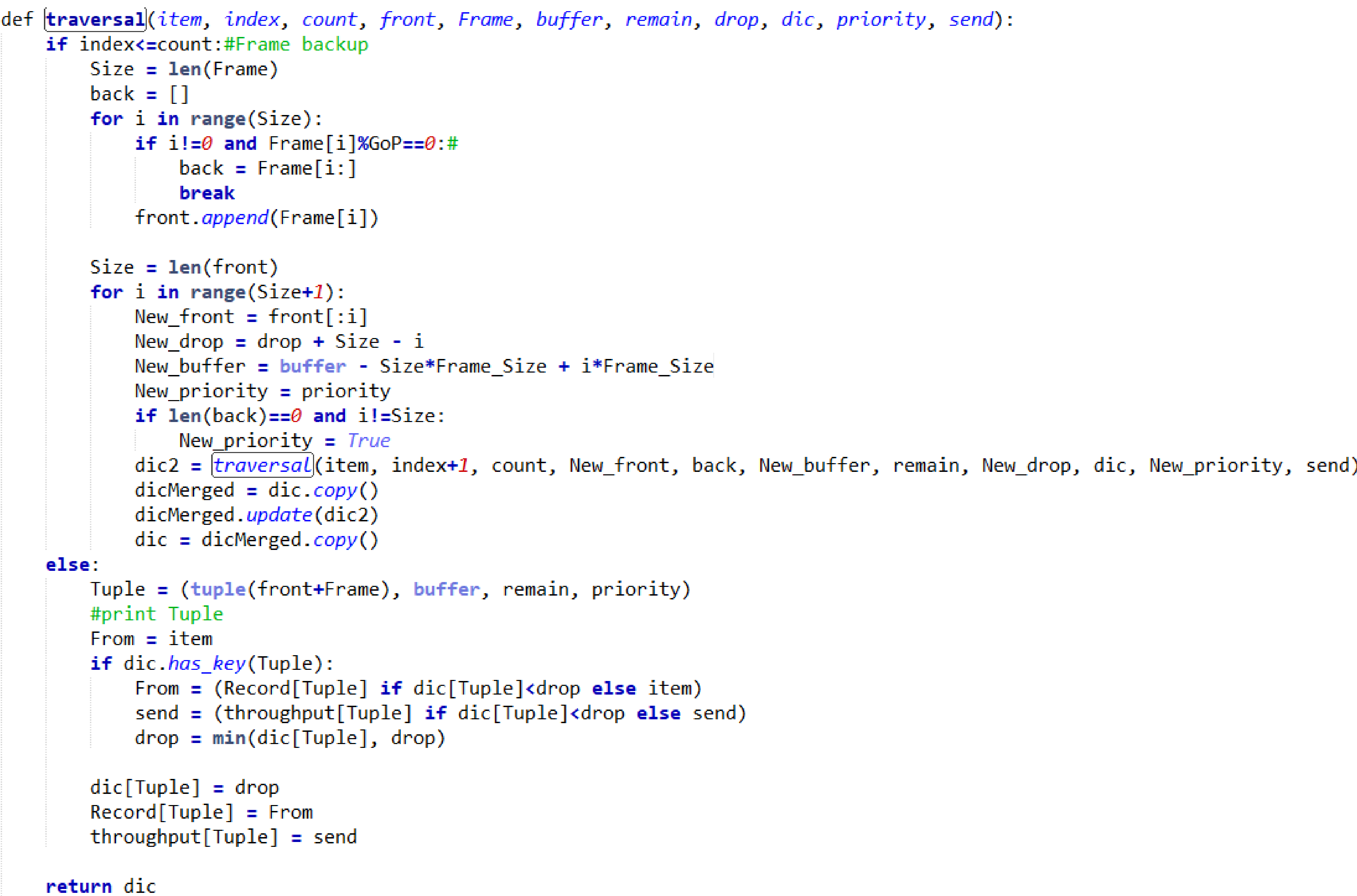

```
def traversal(item, index, count, front, Frame, buffer, remain, drop, dic, priority, send):
    if index<=count:#Frame backup
        Size = len(Frame)
        back = []
        for i in range(Size):
            if i!=0 and Frame[i]%GoP==0:#
                back = Frame[i:]
                break
            front.append(Frame[i])

        Size = len(front)
        for i in range(Size+1):
            New_front = front[:i]
            New_drop = drop + Size - i
            New_buffer = buffer - Size*Frame_Size + i*Frame_Size
            New_priority = priority
            if len(back)==0 and i!=Size:
                New_priority = True
            dic2 = traversal(item, index+1, count, New_front, back, New_buffer, remain, New_drop, dic, New_priority, send)
            dicMerged = dic.copy()
            dicMerged.update(dic2)
            dic = dicMerged.copy()
    else:
        Tuple = (tuple(front+Frame), buffer, remain, priority)
        #print Tuple
        From = item
        if dic.has_key(Tuple):
            From = (Record[Tuple] if dic[Tuple]<drop else item)
            send = (throughput[Tuple] if dic[Tuple]<drop else send)
            drop = min(dic[Tuple], drop)

        dic[Tuple] = drop
        Record[Tuple] = From
        throughput[Tuple] = send

    return dic
```