CS 7260
Scribe Notes: 7 November 2012
Scriber: Kevin Peek

In the previous lecture, we considered one type of event: the arrival of a new packet to an existing flow causing the flow to move down the ladder (movement along the ladder corresponds to rearranging the AVL-tree).

There are two types of events possible:

1) (mentioned above) A packet arrives to an active (under gps) flow
   - Look up the tree node corresponding to the new packet
   - Change its virtual finish time by l/w_j (length over flow weight)
   - Reinsert into tree
2) A packet arrives to an inactive flow
   - We already know the current real time, t
   - Calculate virtual finish time as $v(t) + l/w_j$ (assume we can find v(t))
   - Insert new node

But how do we find v(t) given t?

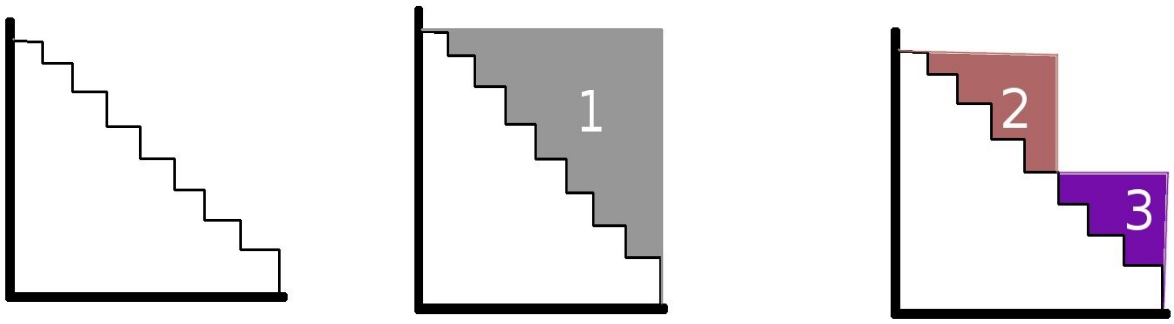
Let,

"slow" = how far we have processed
"fast" = current time
"last" = last t → v(t) mapping we have done

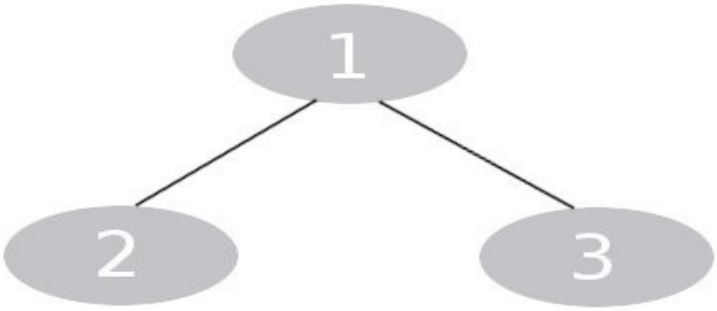
assume t' := "slow" = "last"

So, we know t', v(t'), and t. We want to find v(t). To find v(t), build a binary tree over the ladder and do a binary search on the area over the ladder.

Consider the following ladder:

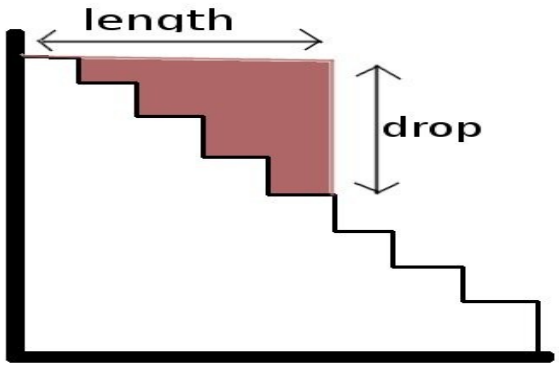


In order to perform the binary search to find v(t), we need to upgrade the data structure from a heap to a balanced tree. In addition, each node of the tree must store the area of some triangle above the ladder.



The root node stores the value of area 1. The left child stores the value of area 2, and the right child stores the value of area 3.

When a packet arrives in case 1, a delete, update, and insert occurs. This means the node will “dance” up or down the AVL tree twice leaving 2 “paths of destruction” where the area field must be recalculated. In order to calculate the area, we must define two additional values which will be stored at each node.



We must store the 'length' and 'drop' corresponding to the values shown above. To calculate the area for a given node we set

parent.area := left.area + right.area + left.drop x right.lenght

This can be calculated in O(logn) time.