```java
import java.util.ArrayList;

class ChannelSet{
    /**
     * This is a list of Channels that have the same Frequency
     */

    private List<Channels> channels;
    private int totalNumberOfChannelsAfterOrdering = -1; // If this value is -1 there has been a problem in algorithm.
    private int frequencyOfSet = -1;
    private int[] ticks;

    public ChannelSet(){
        channels = new ArrayList<>();
    }

    public void add(Channel channel){
        channels.add(channel);
    }

    /**
     * After this function is run the array ticks contains the t values of all ticks that have a data read out for this
     * Frequency
     */
    public void calculateOrderedTotal(){
        if(totalNumberOfChannelsAfterOrdering == -1){
            totalNumberOfChannelsAfterOrdering = channels.size();
        }
        if(totalNumberOfChannelsAfterOrdering > 0){
            frequencyOfSet = channels.get(0).getFrequency();
            ticks = new int[1/frequencyOfSet]; // calculates the number of ticks in a second thats fired by this frequency set
            // The fiddle factor offset here is to account for the fact that all frequencies are multiples of 1000
            // but they tick at t=0 not t=1000
            ticks[0] = 1;
            for(int i=0;i<ticks.length-1;i++){
                ticks[i+1] = (i*1000)/frequencyOfSet;
            }
        }
    }


    public int getFrequencyOfSet(){
        return frequencyOfSet;
    }

    public int[] getTicks(){
        return ticks;
    }

    public boolean doesTickAppearInThisSet(int tick){
        for(int t : ticks){
            if(t == tick){
                return true;
            }
        }
        return false;
    }

    public List<Channel> getChannelList(){
        return channels;
    }
}
```