

One key issue posed from the consensus mechanism described was the propagation and distribution of information across primarily producers but eventually across the entire network. With larger producer pool which are necessary for security consideration, large lists of information are impractical for distribution. When it is considered the every PID is 32Bytes long, for a group of 1000 producers lists would reach 32KBytes if the raw lists of producers are distributed. While compression can be used, what is gained in size of the elements is lost in time for compression and decompression of the lists. This is especially true when it is considered that every producer could potentially have to decompress 999 lists from a pool of 1000 producers. Furthermore using a User Datagram Protocol (UDP) as used in Catalyst, the potential for lost information on elements of that size is high. Therefore a more compact and efficient method is required.

Bloom filters are data structures that allow proof of membership of a set. They work in such a way that they can never provide a false negative i.e. showing that an element is in fact in the bloom filter as false. They can however show false positives meaning that an element that is not in the bloom filter can in fact come back true. Bloom filters are simple byte arrays of a specified length. Upon initialisation all bits within the byte array are set to 0. For each element to be added to the bloom filter it is hashed a defined number of times and the corresponding bit for the hash is flipped to a 1. For example for a small bit array 0bx0000 and element **Hello** is hashed to give the digest 04 then the 4<sup>th</sup> bit would be flipped so the bloom filter would become 0bx1000. This works on a modulo point, so if the digest was 05 then the first bit would be flipped giving the bloom filter 0bx0001. For each element added to the bloom filter multiple hashes are created and added to the bloom filter, for example if 3 hashing functions are used 3 bits in total will be flipped in the bloom filter and these 3 flipped bits will represent the element.

The reason for choosing a bloom filter over directly transferring the lists is for size considerations. Distributing a list containing 100 or a 1000 elements will be extremely cumbersome due to the size. While a bloom filter for 1000 elements would be approximately 1.4kBytes in size. While if