# Stripped Down PeerCube Definition

August 25, 2016

# 1 Algorithm

This algorithm describes a distributed hash table comprised of hypercube of clusters of peers. It is a simplified version of the PeerCube algorithm. It does not account for merging or splitting of clusters or the failure or addition of peers. It also does not differentiate between peer types because no peers can join, so no peers could ever be non-core. It does remove some uniqueness constraints so that the network does not break in the case of hash collisions, however unlikely they may be.

## 1.1 Constants

$m$ : The length of an id. $2^m$ is the maximum number of clusters
$S_{max}$ : maximum size of a cluster. $2^m * S_{max}$ is the max number of peers
$S_{min}$ : minimum size of a cluster

## 1.2 Globals

Values used for analysis
Peers : The set of all unique peer instances (by address) in the network
Clusters: The set of all unique clusters instances in the network
Addrs : The set of all peers' addresses in the network

## 1.3 Starting State

$|Peers| > S_{min} * m$
$|Clusters| = 2^m$
$\forall p \in Peers, |p.RT| = m = p.cluster.dim$
$\forall p \in Peers, p.RT_i = C \in Clusters \mid C.label = b_0...\bar{b}_i...b_{m-1}$

## 1.4 Functions

**function** D$(A, B)$ ▷ Returns the distance between a and b
**return** $\sum_{i=0, a_i \neq b_i}^{m-1} 2^{m-i}$

## 1.5 Key $k$

### 1.5.1 Attributes

$str$ : Eliminates hash collision issues. The plain english representation of the key.
$id \leftarrow hash(k.str)$ : Used to determine the location for the key; A bitstring

## 1.6 Peer $p$

The network contains an arbitrary number of peers. Peers are representations of individual computers, and are the only datastructure that is directly linked to a physical location.

### 1.6.1 Attributes

$id \leftarrow$ ID A randomly assigned, non-unique identifier for peer $p$.
$addr$ : ip address of the peer
$cluster$ : $p$ belongs to exactly one cluster
$RT$ : Routing table for peer $p$, contains cluster instances. $p.RT_i = C \iff \exists C \mid C.label = b_0...\bar{b}_i...b_{m-1}$
$RHC$ : Response Handler Count for $p$. A map of string to integers, with an undefined index of the map returning -1.
$RHD$ : Response Handler Data for $p$. A map of string to byte arrays. The map returns nil for an undefined index
$DataStore$ : A map of string to an array of bytes. The map returns nil for an undefined index.

### 1.6.2 Methods

**function** FINDCLOSESTCLUSTER(id)
  **if** p.cluster.label PREFIXOf k or $p.dim = 0$ **then**
    **return** p.cluster
  $closest = p.RT_0$
  $i \leftarrow 1$
  **for** $i <$ **do**
    **if** **then**$\mathrm{D}(p.RT_i, id) < \mathrm{D}(closest.label, id)$
      $closest \leftarrow p.RT_i$
  **return** $closest$
**function** LOOKUP($key$)
  **assume:** No ongoing lookup operations for $key$.
  $p.RHC[key.str] \leftarrow \frac{S_{min}+1}{3} + 1$
  $C \leftarrow p.\text{FINDCLOSESTCLUSTER}(key.id)$
  **for all** $\varphi \in R_c \subset C.V_c \mid |R_c| = \frac{S_{min}-1}{3} + 1$ **do**
    **send** $('LOOKUP'$, $key$, $p$, $p)$ to $\varphi$
  **wait** until $p.RHC[key.str] = 0$
  **return** $p.RHD[key.str]$

### 1.6.3 Network Messages

$('LOOKUP'$, $key$, $origin$, $prev)$ A lookup request
$('LOOKUPRETURN'$, $key$, $data$, $origin)$ A lookup response

### 1.6.4 Network Reactions

**if** $('LOOKUP'$, $key$, $origin$, $prev)$ recieved from $N$ **then**
  $p.RHC_{key.str} \leftarrow \frac{S_{min}+1}{3} + 1$
  $C \leftarrow p.\text{FINDCLOSESTCLUSTER}(key.id)$
  **if** $C.label = p.cluster.label$ **then**
    **if** $p.RHC_{key.str} = 0$ **then**
      $p.RHC_{key.str} \leftarrow \frac{S_{min}+1}{3} + 1$
      **for all** $\varphi \in C.V_c$ **do**
        **send** $('LOOKUP'$, $key$, $p$, $p)$ to $\varphi$
    **else**
      **send** $('LOOKUPRETURN'$, $key$, $p.DataStore[key.str]$, $p)$ to $\varphi$
  **else**
    **for all** $\varphi \in R_c \subset C.V_c \mid |R_c| = \frac{S_{min}-1}{3} + 1$ **do**
      **send** $('LOOKUP'$, $key$, $p$, $p)$ to $\varphi$
  **wait** until $p.RHC[key.str] = 0$
  $p.RHC[key.str] \leftarrow -1$
  **send** $('LOOKUPRETURN'$, $key$, $p.RHD[key.str]$, $p)$ to $\varphi$
**if** $('LOOKUPRETURN'$, $key$, $data$, $origin)$ recieved from $N$ **then**
  **if** $p.RHC_{key.str} \neq -1$ **then**
    $p.RHD[key.str] \leftarrow data$
    $p.RHC[key.str] \leftarrow \text{MAX}(p.RHC[key.str] - 1, 0)$

## 1.7 Cluster $C$

The network contains an arbitrary number of peers.

### 1.7.1 Attributes

$dim = |C.label|$
$label = b_0 ... b_{C.dim-1}$; A unique identifier for $C$. $\nexists C' \mid C'.label$ PREFIXOF $C.label$
$V_c \leftarrow \{p \in Peers \mid \text{PREFIXOF}(C.label, p.id))\}$;

## 2 Assertions

$Peers$ is constant
$Clusters$ is constant
$p.V_c$ is constant for all $p \in Peers$