|  |  |  |
| --- | --- | --- |
| Project | **IEEE 802.21 MIHS**  **<**[**http://www.ieee802.org/21/**](http://www.ieee802.org/21/)**>** | |
| Title | **Proposed remedy for Comments 93, 97, 100 and 123** | |
| DCN | **21-14-0xxx-00-MuGM** | |
| Date Submitted | **May 13, 2014** | |
| Source(s) | Yoshihiro Ohba (Toshiba) |  |
| Re: | IEEE 802.21 Session #62 in Waikoloa | |
| Abstract | This document describes a proposed remedy for LB7c Comments #93, #97, #100 and #123 about GKB fragmentation. | |
| Purpose | For LB7c Comment Resolution | |
| Notice | This document has been prepared to assist the IEEE 802.21 Working Group. It is offered as a basis for discussion and is not binding on the contributing individual(s) or organization(s). The material in this document is subject to change in form and content after further study. The contributor(s) reserve(s) the right to add, amend or withdraw material contained herein. | |
| Release | The contributor grants a free, irrevocable license to the IEEE to incorporate material contained in this contribution, and any modifications thereof, in the creation of an IEEE Standards publication; to copyright in the IEEE’s name any IEEE Standards publication even though it may include portions of this contribution; and at the IEEE’s sole discretion to permit others to reproduce in whole or in part the resulting IEEE Standards publication. The contributor also acknowledges and accepts that IEEE 802.21 may make this contribution public. | |
| Patent Policy | The contributor is familiar with IEEE patent policy, as stated in [Section 6 of the IEEE-SA Standards Board bylaws](http://standards.ieee.org/guides/opman/sect6.html#6.3) <[http://standards.ieee.org/guides/bylaws/sect6-7.html#6](http://127.0.0.1:4664/cache?event_id=757737&schema_id=1&s=5X0vID10lu_E6yrIkWkNd4Wz2H8&q=hancock)> and in *Understanding Patent Issues During IEEE Standards Development* <http://standards.ieee.org/board/pat/faq.pdf> | |

# Comments



# Discussion

* If GKB is not fragmented, we can use either CreateCompleteSubtree or CreateCompleteSubtreeFragments with setting M (Maximum number of subtrees per fragment) to a very large number. If an implementation dynamically determines whether a GKB is fragmented or not, it should always use CreateCompleteSubtreeFragments.
* CreateCompleteSubtreeFragements can be defined as the default algorithm for GKB fragmentation, without precluding to use other algorithms that satisfy the conditions described in 9.5.2.3.
* A GKB fragment including a single-fragment (or non-fragmented) GKB consists of at most one SUBGROUP\_RANGE, one COMPLETE\_SUBTREE, and at most one GROUP\_KEY\_DATA, and they are carried in MIH\_MN\_Group\_Manipulate.response and MIH\_Net\_Group\_Manipulate.request primitives and MIH\_MN\_Group\_Manipulate response and MIH\_Net\_Group\_Manipulate request messages.
* An example for obtaining T and R in Annex U is provided below.

# Suggested Remedy:

[1] Change 2nd paragraph in 9.5.3.1.1 as follows:

A GKB Generator. This component is comprised of CreateCompleteSubtree procedure (see 9.5.1.2) or CreateCompleteSubtreeFragments procedure (see 9.5.2.3), and MasterGroupKeyWrapping procedure (see 9.5.2.1). If GKB is always not fragmented, CreateCompleteSubtree procedure should be used. Otherwise, CreateCompleteSubtreeFragments procedure should be used.

[2] Change Step d) ii and iii of 9.5.3.1.1 as follows:

ii. If ComplementSubtreeFlag = 0

1. If CreateCompleteSubtree procedure is used, the MIH User sends MIHF IDs of the group members and all Node Indices to the CreateCompleteSubtree procedure and receive CompleteSubtree for the GKB.
2. If CreateCompleteSubtreeFragments procedure is used, the MIH User sends MIHF IDs of the group members, all Node Indices, and a threshold for fragmentation to the CreateCompleteSubtreeFragments procedure and receive CompleteSubtree and SubGroupRange for each GKB fragment. If there is only one GKB fragment created, SubgroupRange is removed.

iii. If ComplementSubtreeFlag = 1,

1. If CreateCompleteSubtree procedure is used, the MIH User sends MIHF IDs of the non-group members and all Node Indices to the CreateCompleteSubtree procedure and receive CompleteSubtree for the GKB.
2. If CreateCompleteSubtreeFragments procedure is used, a threshold for fragmentation to CreateCompleteSubtreeFragments procedure, and receive CompleteSubtree and SubGroupRange for each GKB fragment. If there is only one GKB fragment created, SubgroupRange is removed.

[3] Remove step d) iv of 9.5.3.1.1.

[4] Add the following sentence after step i). “Note: steps e) through i) are performed for each GKB fragment.

[5] Change the following text in 9.5.2.3: “An algorithm by which Complete Subtrees and Subgroup Ranges that satisfy these conditions 1 is defined as follows.”

to:

“The default algorithm by which Complete Subtrees and Subgroup Ranges that satisfy these conditions is defined as follows.”

[6] Add the following text to Annex U.

“

(T, R) for a given depth d can be created as the output of CreateTree(d) function in the following code.

class Index:

def \_\_init\_\_(self, len, val=""):

self.len=len

self.val=val

class Node:

def \_\_init\_\_(self, index, l=None, r=None):

self.index=index

self.left=l

self.right=r

def CreateTree(depth):

T=[]

def CreateNode(index):

if index.len >= depth: # leaf

n=Node(index)

else: # non-leaf

l=CreateNode(Index(index.len+1, index.val+"0"))

r=CreateNode(Index(index.len+1, index.val+"1"))

n=Node(index,l,r)

T.append(n)

return n

n=CreateNode(Index(0))

return T,n

“