Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Efficient Systematic CID Allocation and Relay Path Configuration Mechanism for IEEE 802.16j (Multi-hop Relay)**
Date Submitted	**2007-1-8**

Source(s)	Aik Chindapol Jimmy Chui Hui Zeng Siemens Corporate Research Princeton, NJ, 08540, USA	Voice: +1 609 734 3364 Fax: +1 609 734 6565 Email: aik.chindapol@siemens.com
	Teck Hu Siemens Networks Boca Raton, FL 33431, USA	
	Yuan-Ying Hsu Telcordia Applied Research Center Taiwan Co., Taipei, Taiwan	yyhsu@tarc-tw.research.telcordia.com
	Jen-Shun Yang, Tzu-Ming Lin, Wern-Ho Sheen, Fang-Ching Ren, Chie Ming Chou, I-Kang Fu Industrial Technology Research Institute (ITRI)/ National Chiao Tung University (NCTU), Taiwan 195,Sec. 4, Chung Hsing Rd. Chutung, Hsinchu, Taiwan 310, R.O.C.	jsyang@itri.org.tw
	Byung-Jae Kwak, Sungcheol Chang, Young-il Kim ETRI 161, Gajeong-Dong, Yuseong-Gu, Daejeon, Korea 305-350	Voice: +82-42-860-6618 Fax: +82-42-861-1966 bjkwak@etri.re.kr
	Kyu Ha Lee Samsung Thales San 14, Nongseo-Dong, Giheung-Gu, Yongin, Gyeonggi-Do, Korea 449-712	Voice: +82-31-280-9917 Fax: +82-31-280-1562 kyuha.lee@samsung.com

Re:	IEEE 802.16j-06/034:“Call for Technical Proposals regarding IEEE Project P802.16j”

Abstract	This document describes the route update procedures with efficient CID management.
Purpose	This contribution is provided as input for the IEEE 802.16j baseline document.
Notice	This document has been prepared to assist IEEE 802.16. 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 this contribution may be made public by IEEE 802.16.
Patent Policy and Procedures	The contributor is familiar with the IEEE 802.16 Patent Policy and Procedures <http://ieee802.org/16/ipr/patents/policy.html>, including the statement "IEEE standards may include the known use of patent(s), including patent applications, provided the IEEE receives assurance from the patent holder or applicant with respect to patents essential for compliance with both mandatory and optional portions of the standard." Early disclosure to the Working Group of patent information that might be relevant to the standard is essential to reduce the possibility for delays in the development process and increase the likelihood that the draft publication will be approved for publication. Please notify the Chair <mailto:chair@wirelessman.org> as early as possible, in written or electronic form, if patented technology (or technology under patent application) might be incorporated into a draft standard being developed within the IEEE 802.16 Working Group. The Chair will disclose this notification via the IEEE 802.16 web site <http://ieee802.org/16/ipr/patents/notices>.

Efficient Systematic CID Allocation and Relay Path Configuration Mechanism for IEEE 802.16j (Multi-hop Relay)

1 Introduction

In 802.16e, each connection (both management and data) is identified by a Connection ID (CID) [2]. There is no routing required; data is transmitted solely between the BS and the MS. In a centralized multi-hop relay system, all relay stations form a tree topology with MR-BS as the root. The routing for each MS is decided by the MR-BS. The routing path could be mainly based on the network topology and other number of factors such as measured channel qualities, QoS of each connection, fairness, etc.

This contribution proposes to use CID assignment in a multi-hop relay system as a means to indicate network topology and further help the MR-BS to perform routing path selection. In this scheme, each relay station is assigned a range of CIDs for which the relay is responsible for decoding and forwarding. The parent node will control a superset of this CID range, and any child nodes (both RS and MS) will be assigned disjoint subsets of the CID range. Because of the structure of this CID assignment, the MR-BS could compute the relay path simply based on CIDs of destination station and each relay station can recognize its packets and forward them to corresponding stations. In this way, the routing can be maintained automatically along with CID assignment and thus significantly decrease complexity, signal overhead, and path setup latency. The systematic CID allocation could also help to reduce the delay when handoff of MRS or MS occurs, which could be achieved by locating anchor RS and updating CID of moving station to maintain the topology quickly.

The proposal for this implementation has the following advantages:

- Simplified operation of the relay
- Reduction of overhead and delay in route updates
- Reuse of existing signaling to reflect topology changes due to the movement of MRSs or MSs
- Quickly locating the anchor RS for fast handoff while MRSs or MSs move

2 General Description

A unidirectional connection between BS and MS or between BS and RS is established for service flow traffic, and each connection is identified by a connection identifier (CID) [2][3]. The CID for each connection is inserted within the MAC header of a packet. When it is received, first each station checks whether the CID of the packet is for itself or for its subordinate stations. Each station accepts the packet and does the process if the packet is intended to itself or its subordinate station. Otherwise, it ignores the packet and does nothing.

Each station can distinguish the received packets by examining the CID in the MAC header or in the MAP-IE, and this can be used to maintain the routing structure. By combining CIDs with the routing for each connection, the routing structure can be updated and maintained easily along with CIDs, and the overall overheads for the routing can be reduced.

3 Partitioning of Integers

3.1 General Idea

To systematically assign CIDs to the MR-BS and RSs, the proposed CID allocation mechanism adopts the partitioning of the positive integers into subsets. In this section, we describe two methods of the partitioning of integers. This can be accomplished by factorization into prime numbers [6], or into contiguous blocks [7]. The idea is to map these subsets to nodes in a network (assumed to be a tree topology) which will assist in identifying the placement of the node in the tree.

Each node of the tree represents a subset of **Z**, the set of all positive integers. The leaves of the tree are pairwise disjoint subsets of the integers. Each parent node is a superset of the union of its children. For example, in Figure 1, $B \supset (D \cup E)$ and $D \supset (H \cup J \cup K)$. The tree can grow; at a particular node, its children must satisfy two conditions. 1) the children must be subsets of the parent node; 2) the children are pairwise disjoint.

Due to this structure, any node (root, leaf, or intermediary) can determine whether a particular integer will exist in its subtree (with itself as the root). Intermediary nodes must distinguish between two types of integers; those that terminate at the node (terminal integers), and those that do not terminate at the node (non-terminal integers). We provide two examples of integer partitioning that assume only one terminal integer at each intermediary node, and briefly mention how multiple terminal integers (per intermediary node) can be attained.

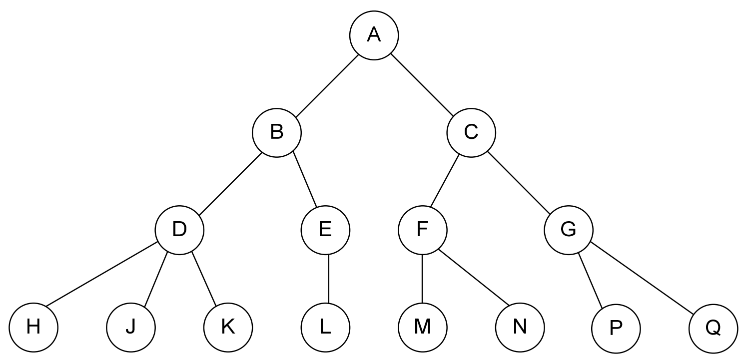

Figure 1: an example of a network tree (an abstract model)

3.2 Examples of integer partitioning: contiguous integer blocks

This is a simple implementation. The root node represents **Z**. Each of its children (1^{st} tier nodes) are assigned a contiguous range of **Z** (and pairwise disjoint). For a particular 1^{st} tier node (with range $[p_1, p_2]$), its children (2^{nd} tier nodes) are each assigned a contiguous subset of $[p_1, p_2]$ (and pairwise disjoint). This process continues for the entire tree. In Figure 2a, we demonstrate how the tree in Figure 1 can partition the integers using contiguous integer block methods.

In Figure 2a, the terminal integers for nodes B, C, D can be set to 1000, 2000, and 400 respectively. Allowing multiple terminal integers per intermediary node is trivial.

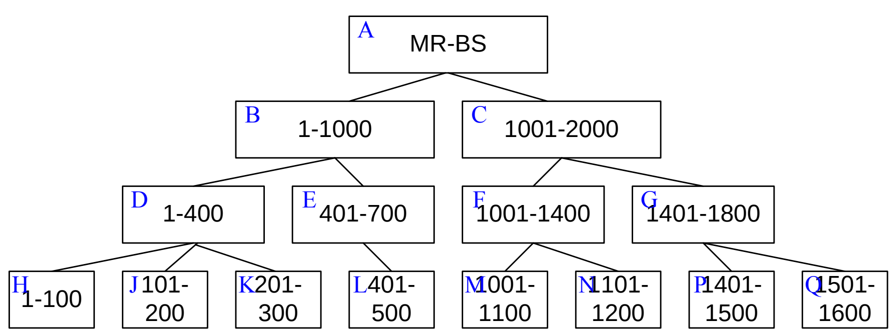

Figure 2a: partitioning of integers using contiguous blocks. The choice of range length being multiples of 100 is arbitrary.

3.3 Examples of integer partitioning: prime factorization

Each node can be represented by a number. The root node is assigned a number of 1, and each of its immediate children (1st tier nodes) is identified by a unique prime number. For a particular 1st tier node (with value p), its children (2nd tier nodes) is identified by the product pq, where q is prime number such that $q \geq p$. This restriction is in place so that the following condition is satisfied: All 2nd tier nodes are identified by a unique integer with two prime factors. A 3rd tier node is identified by the product pqr, where r is a prime number such that $r \geq q$. Then all 3rd tier nodes are identified by a unique integer with three prime factors. This process continues for the entire tree.

To convert these values into subsets of **Z** (as discussed in Section 3.1) is simple; if a node has a terminal value of $n = \prod_{i=1}^{k} p_i$ (and has highest prime factor p_k), then the node represents all integers in the set $n\mathbf{Z}^{p_k}$, where $\mathbf{Z}^q = \bigcup_{k=0}^{\infty} \left\{ \prod_{i=1}^{k} p_i \middle| p_i \geq q \right\}$ (p_i are prime). In words, Z^q is the set of all positive integers, including 1, whose lowest prime factor (if it exists) is greater than or equal to q. The condition as set out in Section 3.1 is satisfied. We also demonstrate how the tree in Figure 1 can partition the integers using prime factorization method in Figure 2b.

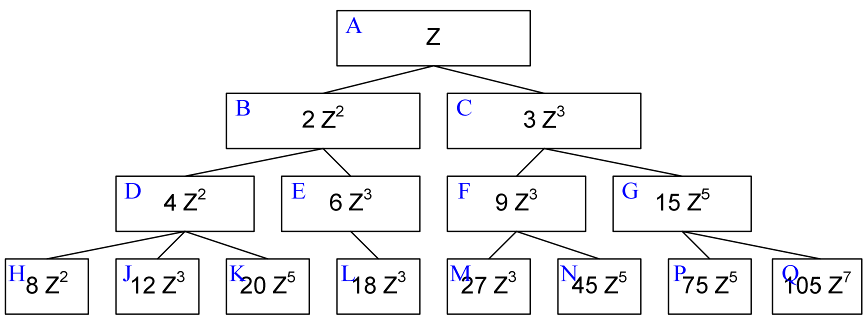

Figure 2b: partitioning of integers using prime factorization.

Note that a simple way of allowing multiple terminal values at intermediary nodes is to merge nodes. For example, the logical nodes H and J can represent the same physical node.

An alternate method is to have a strict inequality during the assignment of immediate children nodes. That is, for a node labeled represented as 2, instead of assigning immediate children nodes as 4 = 2x2 and 6 = 2x3, we assign the values 6 = 2x3 and 10 = 2x5. Then, for this node, all numbers that are multiples of 4 can be the terminal integers.

4 CID Assignment

We describe the CID allocation for both contiguous integer partitioning method and prime factorization method in the following. This structured CID assignment can be used to assign tunnel CIDs (i.e. CIDs terminating at an access RS), or RS and MS management CIDs. It can also be used to assign MS transport CIDs. Our examples describe the assignment of tunnel CIDs.

4.1 CID allocation using contiguous integer partitioning method

We perform this CID assignment scheme *ignoring the MS in the topology*. This method is compatible with the notion of encapsulating CIDs [9] or tunneling CIDs [8] or embedded MAP IE. [8].

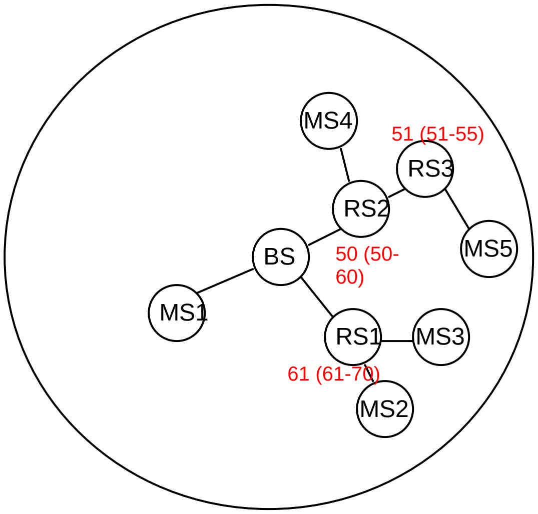

Figure 3: CID Assignment for 4.1

In Figure 3, the MR-BS indicate the range of CIDs that each RS is responsible for, as well as the CID for the connection for each RS node. For each subordinate RS, the assigned CID range has to be a subset of its connecting RS.

4.2 CID allocation using prime factorization method

To systematically assign CIDs to the nodes (MR-BS and RSs), the proposed CID allocation mechanism adopts the partitioning of the positive integers by prime factorization. However, the range of CIDs allowed is 2^{16} (and in fact, for the interesting case of transport CIDs/secondary management CIDs, the number is less due to reserved CIDs ranging from 0 through 2*m*, as well as 0xFEFF through 0XFFFF). For prime factorization partitioning, the value used for factoring must be offset from the true CID by 2*m* to take account for the reservation of the reserved CIDs from 0 to 2*m*.

For the purpose of this description, we assume that each RS has one CID (this assumption can be loosened). Transmitting a packet to a particular RS is simply accomplished by placing the destination RS's CID in the header of the packet. Due to the CID construction, and uniqueness of prime factorization, the intermediate relay nodes know which packets to forward and which packets to ignore.

Figure 4 depicts the scenario for CID distribution for Figure 1, assuming that the network contains only the MS-BS and RSs. We assume node A is the MR-BS and the order of RSs joining the network is from RS B to RS Q. First of all, the MR-BS configures its own CID as 1. RS B and RS C then join the MR network and obtain two prime numbers 2 and 3 as their CIDs, respectively. For RS D and E associating to B, they will obtain CIDs 4 and 6 which are 2 multiplied by two prime numbers 2 and 3. Similarly, for RS F and G, they would have CIDs 9 and 15. We could assign CIDs to other RS in the same manner. Consequently, Figure 5 also depicts the CID allocation tree.

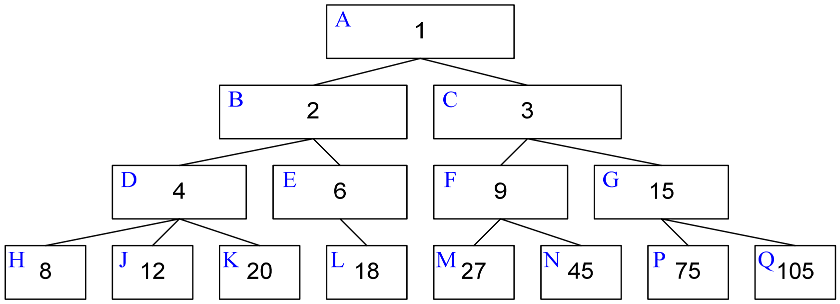

Figure 4: CIDs for MS-BS/RS tree assuming Figure 1 topology. The root node, A, is the BS; all other nodes are RS.

For the CID assignment for MSs, the MR-BS assigns CIDs to MSs from larger numbers to enhance the utilization of CID space. This could prevent the CID allocation tree from exploring too fast. Note that this assignment does not introduce conflicts between MS and RS CID allocation. During the creation of CIDs, prime numbers (for generating RS CIDs) can be skipped.

5 Relay Path Configuration

To utilize the proposed systematic CID allocation in relay path configuration, it is possible to use tunneling [8] [10] or non-tunneling solution [9]. In the following, we take prime factoring method to describe tunneling relay path configuration and take integer contiguous method as an example for non-tunneling relay path configuration. *Both methods could be applied two solutions of relay path configuration.*

5.1 Examples of relay path configuration using tunneling solution

We take figure 5 for example. There are two MSs, which associate to RS K (CID:18) and RS G (CID: 15), in the network. The MR-BS has records for these two MSs and knows their serving RSs. The whole relay path could be divided into two segments: from the source RS to the MR-BS and from the MR-BS to the destination RS. In the first segment, the source RS or other interim RSs could simply relay traffic to its parent RS. The MR-BS could then compute prime factoring of destination RS, which is 15 in figure 6. If the prime factoring sequence of the destination RS is (p0, p1, p2, ..., pm), the second segment of relay path will be $(\prod_{i=0}^{i=1} p_i, \prod_{i=0}^{i=2} p_i, \ldots, \prod_{i=0}^{i=m} p_i)$. As in this example, the prime factoring sequence of 15 is (3, 5), so the path to the destination RS is (3, 3*5). If MS2 moves to the coverage area of RS F (CID: 9), the MR-BS could simply compute the anchor RS of the serving RS and target RS is RS C with CID 3. The anchor RS could help to buffer and redirect traffic to the target RS to reduce handoff delay.

When the tunneling [8] [10] is applied for relaying, the Tunnel CID could be set as the CID of destination RS. Each intermediate RS could compute if the destination RS belongs to its subordinate RSs by the algorithm in Figure 6 where *”self”* is the CID of a certain intermediate nodes executing this algorithm and *”dst”* is the CID of destination RS.

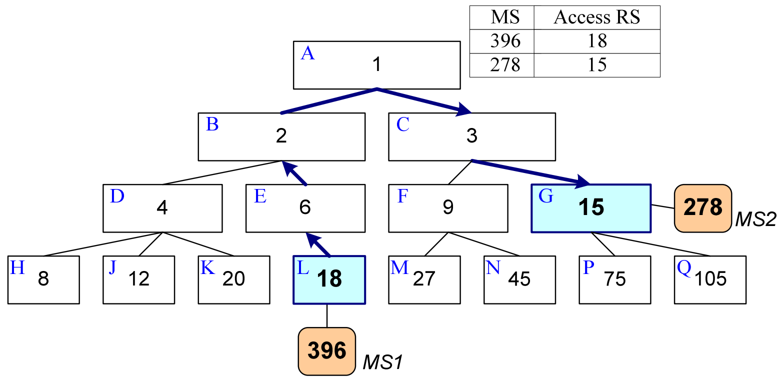

Figure 5: An example of relay path configuration using prime factoring method.

```
int smallerPrime[]
smallerPrime= all prime numbers smaller than the smallest prime factor of self
subordinate=true;
If(dst%self==0){
        for (int i=0; i< number of elements in smallerPrime; i++)}{
                if(dst%smallerPrime[i]==0){
                        subordinate=false;
                        break;
                }
        }
}else{
        subordinate=false;
}
```

Figure 6: Subordinate RS differentiation algorithm

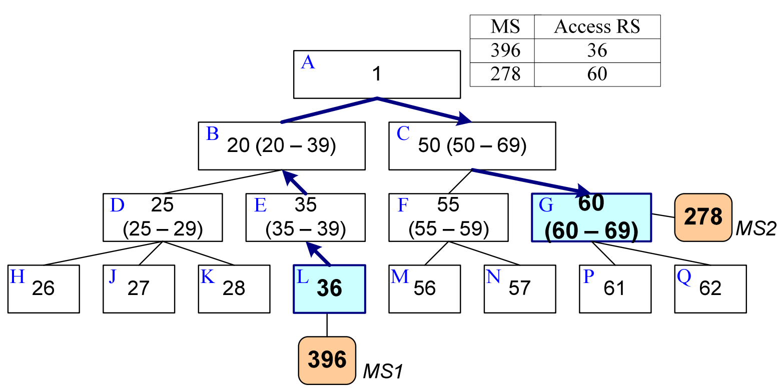

Figure 7: An example of relay path configuration using contiguous integer partitioning method. The number in parenthesis is the range of CIDs that the MR-BS could allocate to the subordinate RS.

Similarly with contiguous integer partitioning, the MR-BS keeps records of the access RS for each MS. For data directed towards MS2, the MR-BS sends the data to the access RS with CID 60. Since this CID belongs to the range of CID of the RS C, it forwards the data to the RS G. Meanwhile, the RS B ignores this data as the CID is not within its range. The similar procedure can be done on the uplink.

5.2 Examples of relay path configuration using non-tunneling solution

The following example, as shown in Figure 7, describes how the relay operation is performed if the same CID is maintained for every hop involved in the transmission. Relay involving the changing of CIDs, or relay involving encapsulation of CIDs [1] can be done in a similar manner.

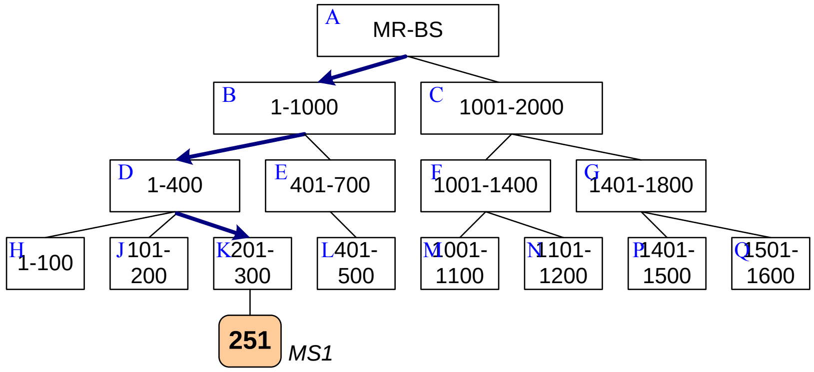

Figure 7: An example of relay path configuration using contiguous integer method.

Consider the transmission of a packet with CID 251. RS B decodes and forwards the message to RS D, and RS C ignores this transmission because the CID does not belong to its CID range 1001-2000. Then, RS D forwards the message to RS K in the same manner. RS K is able to decode and forward this packet because CID 251 is within its forwarding range. Finally, the packet with CID 251 is received by MS 1. The uplink transmission from each MS can be delivered to BS through RS in a similar way.

Once the routing decision is made by the BS, CIDs of MS are assigned according to the procedures described above. Each RS only listens to the packets with CIDs which are within its CID range and ignores otherwise. This removes the necessity to maintain and broadcast routing information.

In addition, the handover of moving MS from one RS to another RS within a cell can be handled easily. For example, if MS 1 moves close to RS L and BS decides to serve MS 1 through RS L, then BS changes MS 1's CID from 251 to 405 directly. Then RS L knows automatically that it should detect and decode the packets for MS 1. This handover within a cell is done transparently to MS. In other words, MS only knows that BS changes its CID and does not know that the routing path has been changed. The BS can request the change of CID using BS-initiated DSC procedures as specified in subclause 6.3.14.9.4.2 of [2], which is summarized in Figure 8. A DSC-REQ is sent by BS to dynamically change the service flow (SF) parameters, including CID, of an existing SF according to the specified SFID. Once the SS receives the DSC-REQ and validates the request, the SF parameters will be changed and DSC-RSP will be sent by SS. After receiving the DSC-RSP, the BS will update the service flow profile accordingly, and send DSC-ACK to confirm. The whole procedure is completed once the SS receives DSC-ACK and finalizes the adjustment.

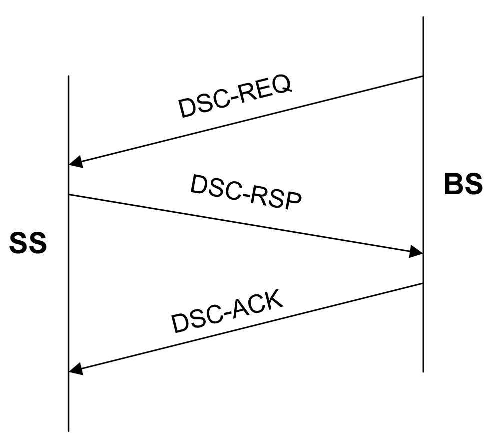

Figure 8: BS-initiated DSC.

CID ranges that have to be used by RS are assigned by BS and transmitted to RSs via CIDRNG-REQ and CIDRNG-RSP management messages as specified in Section 6.

6 Proposed Text

----------------------------Beginning of Text Changes---

[Add the following text into section 6.3.1.3]

6.3.1.3.1 Addressing Scheme for Relaying

In the procedure of network entry and initialization for a new RS, the MR-BS shall assign a range of CIDs for the new RS. The range could be contiguous integer blocks as in Figure 6.3.1.3.X (a) or prime factorization range as in Figure 6.3.1.3.X (b). In the prime factorization range assignment, the MR-BS allocates prime numbers in ascending order to RSs until to the maximum number of CIDs for RSs associated to the MR-BS directly. For a RS associated to another RS with CID range *nZ*, the MR-BS factorizes $n=\Sigma\ p_i$, where all p_i are prime numbers and $p_m \leqq p_n$ if $m<n$. The MR-BS assigns RSs, which associate with RS with CID *n*, CIDs each of which equals to *n* multiplied by a prime number, starting from the largest prime factor p_m of *n*. The whole CIDs space could be shared by RSs and SSs (MSs), and the MR-BS assigns CIDs to MSs from large numbers in the CID space.

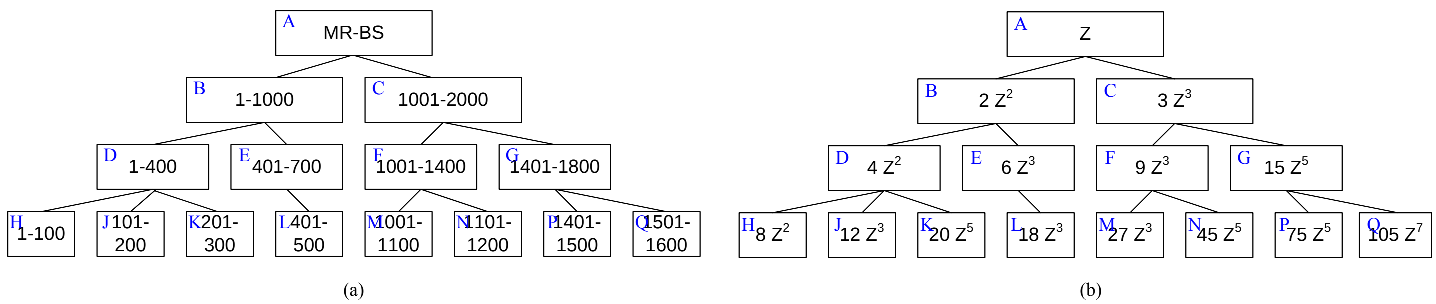

Figure 6.3.1.3.Y: CID range allocation example, (a) contiguous integer block, (b) prime factorization method.

If the topology changes due to some factors like load balancing or QoS management, RSs might change its preceding RS and the CID range could be updated during network re-entry time as shown in Figure 6.3.1.3.Y in two cases.

Case 1: the selected preceding RS is the same as the temporary associated RS for initialization. The MR-BS determines the selected preceding RS still remains RS1 and sends a *RLY_TPY-IND* with original CID range to inform the unchanged of preceding RS (RS1) for the new RS.

Case 2: the selected preceding RS is different from the temporary associated RS for initialization. The MR-BS determines the selected preceding RS is RS2 which is different from the temporary associated RS for initialization. It must issue the new CID range carried *RLY_TPY-IND* messages to the new RS. Upon receiving *RLY_TPY-IND* message, the temporary associated RS (RS1) shall forward the message to the new RS. Thereafter, the new RS shall start to associate with RS2.

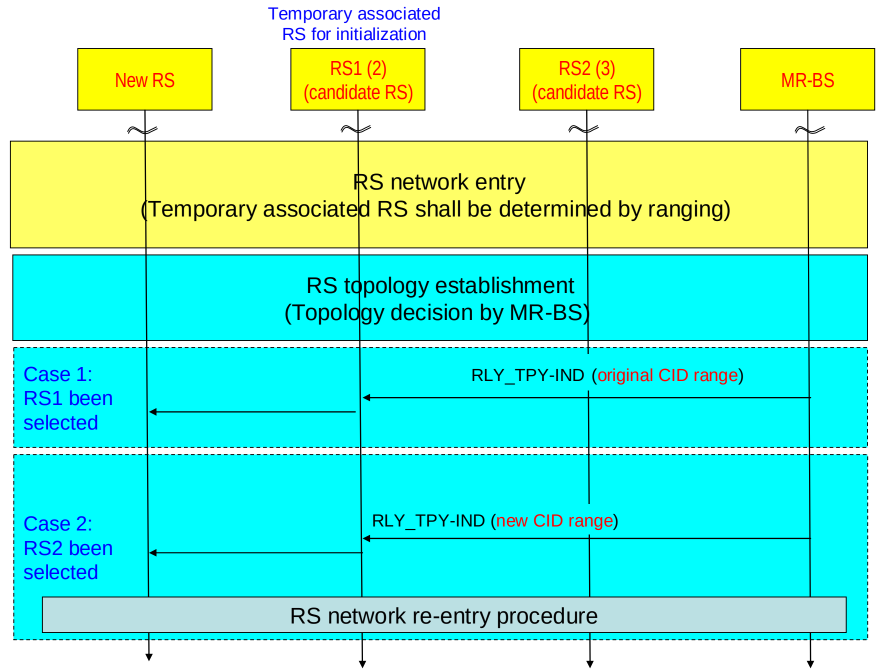

Figure 6.3.1.3.Y: CID range allocation Message flow of a new RS during network re-entry time

6.3.2.3 MAC management messages

[*Insert the following text into Table 14.*]

Type	Message name	Message description	Connection
67	CIDRNG-REQ	CID range assignment request	Basic
68	CIDRNG-RSP	CID range assignment response	Basic
69	RLY_TPY-IND	Relaying mode RS topology indication	Basic

[Add a new subclause 6.3.2.3.62]

6.3.2.3.62 CID Range Assignment Request (CIDRNG-REQ) message

Table *** -- CIDRNG-REQ message format

Syntax	**Size**	**Notes**
CIDRNG-REQ message format {	_	_
Management Message Type = 67	8 bits	Type = 67
If(contiguous integer blocks){		
CID min	16 bits	The minimum value of CID range
CID max	16 bits	The maximum value of CID range
}		
If(prime factorization){		
CID prime base	16 bits	The CID for this RS

}		
}		

[Add a new subclause 6.3.2.3.63]

6.3.2.3.63 CID Range Assignment Response (CIDRNG-RSP) message

Table *** -- CIDRNG-RSP message format

Syntax	**Size**	**Notes**
CIDRNG-RSP message format {	_	_
Management Message Type = 68	8 bits	Type = 68
Confirmation	1 bit	1: the CID range has been changed successfully 0: error
Reserved	7 bits	Shall be set to zero
}		

[Add a new subclause 6.3.2.3.64]

6.3.2.3.64 Relaying mode RS topology indication (*RLY_TPY-IND*) message

An MR-BS shall transmit a RLY_TPY-IND message for indicating what the suitable preceding RS is and indicating the CID for new RS, and then trigger the network re-entry. A RLY_TPY-IND message may also be transmitted to the selected preceding RS for the notification that a new RS with CID descending from this preceding RS.

Table 6.3.2.3.X– *RLY_TPY-IND* message format

Syntax	Size	Notes
RLY_TPY-IND_Message_format(){	–	–
Management Message Type=69	8 bits	Type=69
Target_Station_ID	48 bits	–
Preamble_Index/Subchannel Index	8 bits	This parameter defines the OFDMA PHY specific preamble
HO process optimization	8 bits	HO Process Optimization is provided as part of this message is indicative only. HO process requirements may change at time of actual HO. For each Bit location, a value of '0' indicates the associated reentry management messages shall be required, a value of '1' indicates the reentry management message may be omitted. Regardless of the HO Process Optimization TLV settings, the target Station may send unsolicited SBC-RSP and/ or REG-RSP management messages: Bit #0: Omit SBC-REQ/RSP management messages during re-entry processing Bit #1: Omit PKM Authentication phase except TEK phase during current re-entry processing Bit #2: Omit PKM TEK creation phase during re-entry processing Bit #3: Omit REG-REQ/RSP management during current re-entry processing Bit #4: Omit Network Address Acquisition management messages during current re-entry processing Bit #5: Omit Time of Day Acquisition management messages

		during current reentry processing Bit #6: Omit TFTP management messages during current re-entry processing Bit #7: Full service and operational state transfer or sharing between serving station and target station (ARQ, timers, counters, MAC state machines, etc…)
If(contiguous integer blocks){		
CID min	16 bits	The new minimum value of CID range
CID max	16 bits	The new maximum value of CID range
}		
If(prime factorization){		
CID prime base	16 bits	The new CID for this RS
}		
Padding	variable	If needed for alignment to byte boundary
TLV encoded information	variable	–
}	–	–

6.3.25 Relay path management and routing

Each relay station is assigned a range of CIDs for which the relay is responsible for decoding and forwarding. The CID range is assigned by the MR-BS, and are transmitted to RSs via CIDRNG-REQ and CIDRNG-RSP management messages. During operation, the RS is only responsible for listening to CIDs transmitted within this range.

The BS is responsible for managing the entire CID range. Each RS connected to a parent node (BS or RS) is assigned a subset of the CIDs assigned to the parent node. These subsets are non-overlapping.

By assigning a CID to each service flow, the MR-BS already specifies the relay routing path of the connection.

7 References

[1] IEEE C802.16j-06/004r1, "Recommendations on IEEE 802.16j".

[2] IEEE 802.16-2004, "Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems".

[3] IEEE 802.16e-2005, "Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands *and* Corrigendum 1".

[4] IEEE C802.16j-06/014r1, "Harmonized definitions and terminology for 802.16j Mobile Multihop Relay"

[5] IEEE C802.16j-06/015, "Harmonized Contribution on 802.16j (Mobile Multihop Relay) Usage Models"

[6] IEEE C802.16j-06/171, "Systematic relay station identification allocation and relay path configuration mechanism for IEEE 802.16j (Multi-hop Relay)".

[7] IEEE C802.16j-06/253, "Route Update with Efficient CID Management".

[8] IEEE C802.16j-06/274, "Proposal on addresses, identifiers and types of connections for 802.16j".

[9] IEEE C802.16j-06/254, "Fast Connection Establishment and Maintenance with Relays".

[10] IEEE C802.16j-06/170, "Connection Identification and Transmission for Relay Support"