**IEEE P802.15**

**Wireless Personal Area Networks**

|  |  |  |
| --- | --- | --- |
| Project | IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) | |
| Title | Technical Proposal for IEEE 802.15.8 | |
| Date Submitted | May 6, 2013 | |
| Source | Jeongseok Yu  Woongsoo Na Hyoungchul Bae  Taejin Kim  Yunseong Lee  Juho Lee  Zeynep Vatandas  Sungrae Cho  Junbeom Hur  [Chung-Ang University, Korea] | E-mail:[jsyu@uclab.re.kr]  E-mail:[wsna@uclab.re.kr]  E-mail:[hcbae@uclab.re.kr]  E-mail:[tjkim@uclab.re.kr]  E-mail:[yslee@uclab.re.kr]  E-mail:[jhlee@uclab.re.kr]  E-mail:[zvatandas@uclab.re.kr]  E-mail:[srcho@cau.ac.kr]  E-mail:[jbhur@cau.ac.kr] |
| Re: | This is the original document | |
| Abstract | Technical Proposal for IEEE 802.15.8 | |
| Purpose | To provide the technical guidance including functional and technical requirements to the P802.15 Working Group. | |
| Notice | This document has been prepared to assist the IEEE P802.15. 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 acknowledges and accepts that this contribution becomes the property of IEEE and may be made publicly available by P802.15. | |

Table of Contents

[1. Overview 3](#_Toc361024331)

[2. Definitions 3](#_Toc361024332)

[3. Abbreviations and acronyms 3](#_Toc361024333)

[4. General descriptions 3](#_Toc361024334)

[4.1. Concepts and architecture 4](#_Toc361024335)

[4.1.1 Packet Format 4](#_Toc361024336)

[4.2. Topology 5](#_Toc361024341)

[4.3. Reference model 5](#_Toc361024342)

[5. MAC layer 5](#_Toc361024343)

[5.1. MPDU structure 5](#_Toc361024344)

[5.2. Multiple access 5](#_Toc361024345)

[5.3. Synchronization procedure 5](#_Toc361024346)

[5.4. Discovery procedure 5](#_Toc361024347)

[5.5. Peering procedure 5](#_Toc361024348)

[5.6. Scheduling 5](#_Toc361024349)

[5.7. QoS 5](#_Toc361024350)

[5.8. Interference management 5](#_Toc361024351)

[5.9. Transmit power control 5](#_Toc361024352)

[5.10. Multicast 6](#_Toc361024353)

[5.10.1 Multicast Group Management 6](#_Toc361024354)

[5.10.2 Multicast Data Transmission 14](#_Toc361024355)

[5.10.3 Prevention Loopback Problem 15](#_Toc361024356)

[5.10.4 Reliable Multicast 15](#_Toc361024357)

[5.10.5 Multicast Protocol Using Directional Antenna 16](#_Toc361024358)

[5.11. Broadcast 17](#_Toc361024359)

[5.12. Multi-hop operation 17](#_Toc361024360)

[5.12.1 Multi-hop Unicast Data Transmission 17](#_Toc361024361)

[5.13. Relative positioning 18](#_Toc361024362)

[5.14. Power management 18](#_Toc361024363)

[5.15. Security 18](#_Toc361024364)

[5.15.1 Security modes 19](#_Toc361024365)

[5.16. Coexistence 27](#_Toc361024366)

[5.17. Higher layer interaction 27](#_Toc361024367)

# Overview

The 802.15.8 specification shall be developed according to the P802.15.8 Peer Aware Communication (PAC) project authorization request (PAR), document number 15-12-0063r2 and Five Criteria (5c), document number 15-12-0064r1, which were approved by the IEEE-SA in March of 2012.

# Definitions

# Abbreviations and acronyms

PD PAC Device

FFPD A PD which supports whole specification including routing table and it is always relay-enabled.

RFPD It has two types. These are relay-enabled PD and relay-disabled PD. Relay-enabled is a reduced-function PD which supports routing table for specific multicast group and relay-disabled is a reduced-function PD which does not support routing table.

# General descriptions

This clause provides the basic framework of PDs. The framework serves as a guideline in developing the functionalities of PDs and their interactions specified in detail in the subsequent clauses.

## Concepts and architecture

## Packet Format

# Table 1. ACF (Advertisement Command Frame)

| Name | Type | Description |
| --- | --- | --- |
| Originator Address | Integer | Originator PD’s Address |
| Destination Address | Integer | Destination PD’s Address |
| Sender Address | Integer | Sender PD’s Address |
| Application Type ID | Integer | Application Type ID |
| Application Specific ID | Integer | Application Specific ID |
| Application Specific Group ID | Integer | Application Specific Group ID |
| Device Group ID | Integer | Device Group ID |
| Time To Live (TTL) | Integer | Time To Live (TTL) |
| Sequence Number | Integer | Frame’s Sequence Number |
| Frame Type | Integer | Frame Type |
| Sub Type | Integer | Frame’s Sub Type |

# Table 2. ARCF (Advertisement Reply Command Frame)

| Name | Type | Description |
| --- | --- | --- |
| Originator Address | Integer | Originator PD’s Address |
| Destination Address | Integer | Destination PD’s Address |
| Sender Address | Integer | Sender PD’s Address |
| Receiver Address | Integer | Receiver PD’s Address |
| Application Type ID | Integer | Application Type ID |
| Application Specific ID | Integer | Application Specific ID |
| Application Specific Group ID | Integer | Application Specific Group ID |
| Device Group ID | Integer | Device Group ID |
| Sequence Number | Integer | Frame’s Sequence Number |
| Frame Type | Integer | Frame Type |
| Sub Type | Integer | Frame’s Sub Type |

# Table 3. MGNF (Multicast Group Notification Frame)

| Name | Type | Description |
| --- | --- | --- |
| Originator Address | Integer | Originator PD’s Address |
| Destination Address | Integer | Destination PD’s Address |
| Sender Address | Integer | Sender PD’s Address |
| Receiver Address | Integer | Receiver PD’s Address |
| Application Type ID | Integer | Application Type ID |
| Application Specific ID | Integer | Application Specific ID |
| Application Specific Group ID | Integer | Application Specific Group ID |
| Device Group ID | Integer | Device Group ID |
| Time To Live (TTL) | Integer | Time To Live (TTL) |
| Sequence Number | Integer | Frame’s Sequence Number |
| Notification Type | Integer | Notification Type (Type: 0, 1, 2, 3, 4, 5, 6, 7, 8) |
| Frame Type | Integer | Frame Type |
| Sub Type | Integer | Frame’s Sub Type |

# Table 4. Multicast Data Frame

| Name | Type | Description |
| --- | --- | --- |
| Originator Address | Integer | Originator PD’s Address |
| Destination Address | Integer | Destination PD’s Address |
| Sender Address | Integer | Sender PD’s Address |
| Receiver Address | Integer | Receiver PD’s Address |
| Application Type ID | Integer | Application Type ID |
| Application Specific ID | Integer | Application Specific ID |
| Application Specific Group ID | Integer | Application Specific Group ID |
| Device Group ID | Integer | Device Group ID |
| Time To Live (TTL) | Integer | Time To Live (TTL) |
| Sequence Number | Integer | Frame’s Sequence Number |
| Originator Sequence Number | Integer | Originator Sequence Number |
| Notification Type | Integer | Notification Type (Type: 0, 1, 2, 3, 4, 5, 6, 7, 8) |
| Frame Type | Integer | Frame Type |
| Sub Type | Integer | Frame’s Sub Type |
| Payload | Integer | Multicast Data |

## Topology

## Reference model

# MAC layer

## MPDU structure

## Multiple access

e.g. Contention-based access, Contention-free access

## Synchronization procedure

## Discovery procedure

## Peering procedure

## Scheduling

## QoS

## Interference management

## Transmit power control

## Multicast

### 5.10.1 Multicast Group Management

*5.10.1.1 Finding/Joining Multicast Group*

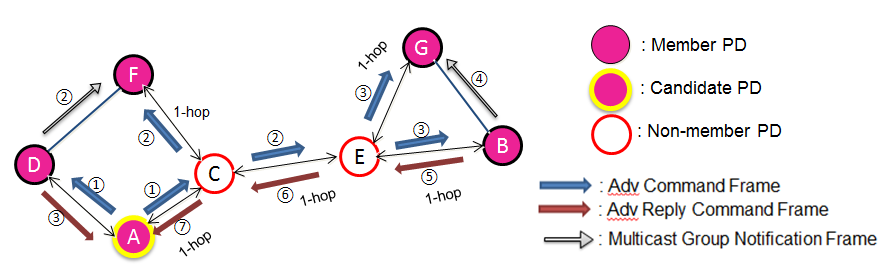
A multicast group consists of two or more PDs with the same application type ID, application specific ID, application specific group ID, and device group ID. It can be formed only if two or more PDs can recognize themselves. Before a PD joins a multicast group, it has to find the multicast group within K-hop coverage. If the PD cannot find the group, then it finds the group periodically. In order to find a multicast group, a PD broadcasts an Advertisement Command Frame (ACF) after random timer *Tj* where the maximum TTL is set to K. Range of *Tj* is [0, *Tjmax* ]. If a PD receive the ACF, it stores the ACF in order to forward it to other PDs and saves backward path in the routing table during expiration timer where is calculated by one-hop RTT and K if it is relay-enabled (Backward path is originator of the ACF, one-hop PD sending the ACF, Device Group ID & Application type ID & Application-specific ID & Application-specific group ID.) It compares the receiving frame’s Device Group ID & Application type ID & Application-specific ID & Application-specific group ID with its own. If they all are same, it replies an ARCF (Advertisement Reply Command Frame) to the PD sending the ACF (Reply of the ARCF depends upon the MGNF explained in the following pages). If any of them is not same, it decrements the TTL of the ACF and forwards the ACF.

In order to limit the duplicate ARCF, PDs replying the ARCF multicast a Multicast Group Notification Frame (MGNF) after random time. (MGNF is explained detail in later section) Then, the PD multicasting the MGNF replies source PD with an ARCF by using backward path. A PD receiving both ACF and MGNF does not reply an ARCF. A PD receiving the ARCF whose destination is not itself saves the route information of ID of the source PD sending the ARCF, ID of the one-hop PD sending the ARCF, Device Group ID, Application type ID, Application-specific ID, Application-specific group ID. Then, this PD is referred to as “FORWARDING PD.” A PD receiving the ARCF whose destination is itself saves the route information same as FORWARDING PD. Then, this PD is referred to as “JOINED MULTICAST GROUP MEMBER.”

When a PD wants to join a multicast group, it broadcasts an ACF to its neighbor PDs. When a PD which is not in the multicast group receives the ACF, it saves the route information and forwards to the others. When a multicast group member PD receives an ACF, it broadcasts an ARCF to it. If there is a duplicate ARCF, the member PD broadcasts a MGNF to the other member PD by notifying that it will send an ARCF instead of it. The node sending MGNF may send KEKs (key encryption keys) to the node which wants to join the multicast group after authentication for secure multicast if needed. It also may send rekeying messages to update an existing group key for backward secrecy. Hereby, a PD receiving the ARCF joins the multicast group.

For example, in Figure 1,

1. If PD A wants to join the multicast group, it broadcasts an ACF,
2. PD C is not in the multicast group. Therefore, it saves the route information and forwards to the others (e.g., F and E). PD D already received the ACF and PD F now receives the ACF. Then, there will be a duplicate ARCF from both of D and F to A. In order to limit the duplicate ARCF, PD D multicasts a MGNF notifying that D will send an ARCF instead of F.
3. PD E forwards the ACF to B and G. D replies with ARCF to A. D may send KEKs to A if needed. D may send rekeying messages also. F does not transmit an ARCF to A.
4. When PD A receives the ARCF, A joins D’s multicast group. Also, PD B receives the ACF from PD E. Then, B multicasts a MGNF to limit a duplicate ARCF from G.
5. PD B replies with an ARCF to A through E. B may send KEKs to A if needed. B may send rekeying messages also. G does not transmit an ARCF to A since the PD G already received the MGNF from B.
6. E forwards the ARCF to PD C.
7. C forwards the ARCF to PD A. Then, PD A is aware of route between PD A and PD B. Now, PD A joins PD B’s multicast group. Although group of B and G are not aware of group of D and F currently, multicasting service still works by simply forwarding multicast data frames.



**Figure 1. Finding/Joining Multicast Group**

*5.10.1.2 Device Group ID Creation*

A multicast group is determined by application type ID, application specific ID, and application specific group ID. Therefore, it is inefficient if transmitting all IDs in a frame and managing routing table. Therefore, we propose a device group ID creation scheme. Device group ID should be unique and distributed by a PD sending the first ARCF in the group. The PD manages/updates group keys for secure and dynamic multicast group communications after authentication procedures described in security section. The PD generates device group ID based on its unicast ID. Since a PD’s unicast ID is unique, prefix concatenated by PD’s unicast ID is also unique.

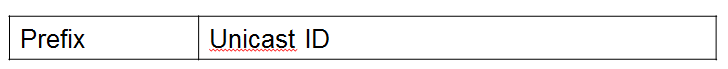
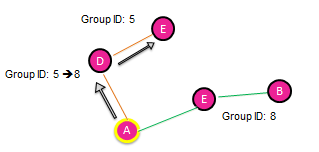


Figure 2. Device Group ID

When two or more multicast groups are merged, the device group ID should be same. The PD recognizing the existence of two or more multicast groups determines the device group ID for those groups randomly. Then, the PD sends MGNF (notification type: 3 and TTL: ∞) to the group that does not have the selected group ID to update multicast group ID.

In Figure. 3, A recognizes that two device group IDs exist in networks. A determines that device group ID is 8. A sends MGNF to D and E to change device group ID. A may send rekeying messages to update an existing group key for backward secrecy.



**Figure 3. Device Group ID Creation**

*5.10.1.3 Multicast Group Notification Frame (MGNF)*

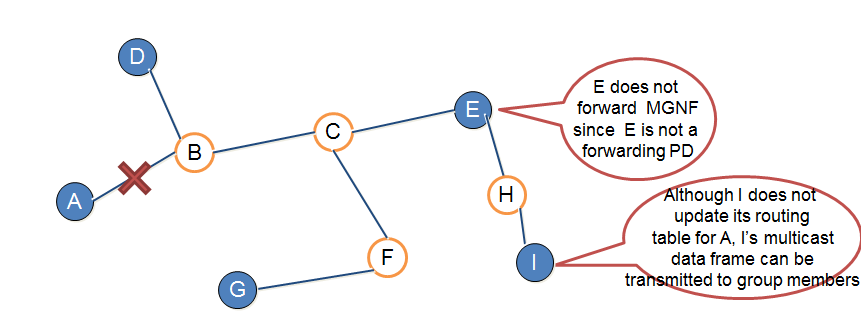
The purposes of MGNF are limiting duplicate ARCF, management of routing table, notifying leaving multicast group, device group ID creation, request for unicast routing, reply for unicast routing, mobility support, local repair for relay-enabled PDs and notification of removed routing entry.

MGNF has eight notification types for above purposes. These are;

Table 5. Type of MGNF

|  |  |
| --- | --- |
| Type | Description |
| 0 | Limiting duplicate ARCF |
| 1 | Management of routing table |
| 2 | Notifying leaving multicast |
| 3 | Device group ID creation |
| 4 | Request for unicast routing |
| 5 | Reply for unicast routing |
| 6 | Mobility support |
| 7 | Local repair |
| 8 | Notification of removed routing entry |

1. Limiting duplicate ARCF(Notification Type: 0): A PD receiving an ACF multicasts MGNF with notification type set to 0 with random timer *Tj*. Range of *Tj* is [0, *Tjmax* ]. Although missing MGNF can increase duplicate ARCF, MGNFs are not retransmitted to avoid flooding. In this case, payload of MGNF contains source of the ACF.
2. Management of routing table (Notification Type: 1): In order to reduce routing entries in the table, each PD maintains entries only for PDs that has exchanged ACFs and ARCFs in its routing table. Each PD in multicast group multicasts MGNFs periodically (with notification type set to 1. Range of random timer *Tj* is [0, *Tjmax*]. Upon receiving a MGNF, a forwarding PD updates the entries of the originator of the MGNF and one-hop PD sending the MGNF in its routing table, and forwards the MGNF. Upon receiving a MGNF, a non-forwarding PD updates the entries of the originator of the MGNF and one-hop PD sending the MGNF in its routing table, but does not forward the MGNF. Also it can be used for detection and routing table update for link breakage. When a link between two nodes is broken, the other member node in the group which is aware of the link breakage removes the entry of the node which has a broken link to its routing table. For example, in Figure 4, we suppose that the link between A and B is broken. E does not receive MGNF from A. E is aware that the link is broken, because A’s MGNF does not arrive before E’s routing table expiration timer is expired. Then, E removes the entry of A to E’s routing table.



**Figure 4. Detection and Routing Table Update For Link Breakage**

1. Notifying leaving multicast (Notification Type: 2): will be explained in *5.10.1.5* section.
2. Device group ID creation (Notification Type: 3): Explained in the *5.10.1.2* section.
3. Request for unicast routing (Notification Type: 4): will be explained in *5.12.1* section.
4. Reply for unicast routing (Notification Type: 5): will be explained in *5.12.1* section.
5. Mobility support (Notification Type: 6): will be explained in *5.10.1.6* section.
6. Local repair (Notification Type: 7): will be explained in *5.10.1.6* section.
7. Notification of removed routing entry (Notification Type: 8): will be explained in *5.10.1.4.2* section.

*5.10.1.3.1* *Reliable MGNF Transmission*

We classify the level of reliable MGNF transmission in two types. These are ACK-based MGNF transmission and no ACK-based transmission. The node which transmits MGNF has to receive ACK-based MGNF from its neighbor nodes for notification types 2,3,4,5 and 8 for reliable MGNF transmission. But it does not have to receive ACK-based MGNF for notification types 0, 1, 6 and 7. Because, the notification type 0 is for limiting duplicate ARCF, thus the PD does not need to receive it. The notification type 1 is for management of routing table and MGNF is transmitted periodically. The notification type 6 is for mobility support. Due to changed route information, the node does not need to receive the ACK-based MGNF from the moving node.

*5.10.1.3.2* *Reliable MGNF Transmission*

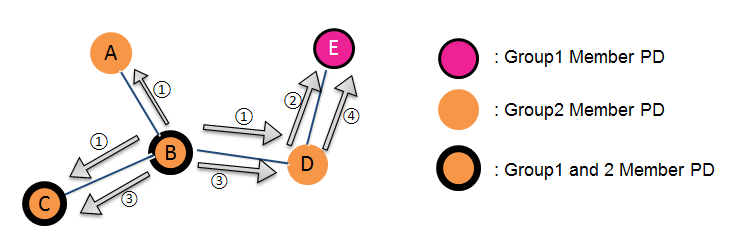
When a PD belongs to multiple multicast groups, it multicasts MGNF multiple times. In order to resolve the above problem, a PD may set Destination Address field of MGNF as *United Multicast Address (UMA)***.** The UMA unifies multiple different groups into a single one. The UMA can reduce the MGNF traffic as the following pages.

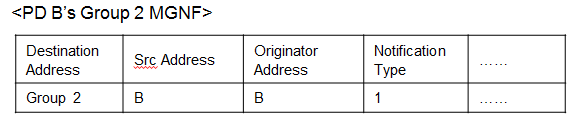
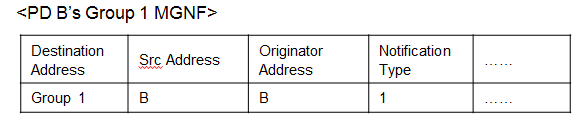
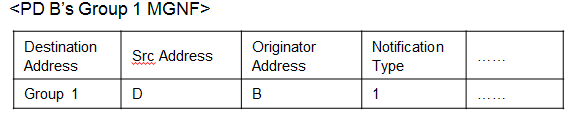
*5.10.1.3.3* *Redundant MGNF Transmission*

When a PD multicast its MGNF to the other nodes, if there are nodes which are member of multiple groups simultaneously, there will be a redundant MGNF transmission.

For example, in Figure 5, we suppose that A, B, C, and D are in group 2 while B, C and E are in group 1. B and C are in groups 1 and 2 simultaneously.

1. B starts to multicast its MGNF for group 1 to C and D (forwarding to E). After C and D verify device group ID field inPDB’s MGNF, PDs C and D recognize that the MGNF belongs to group 1.
2. Then, D forwards the MGNF from B to E. After PD E verifiesdevice group IDin B’s MGNF, E recognizes that the MGNF belongs to group 1.
3. B starts to transmit its MGNF for group 2 to PDs A, C, and D.
4. Then A, C, and D verify their device group ID in PD B’s MGNF. They recognize that the MGNF belongs to group 2. There are 3 transmissions for MGNF (B transmitted MGNF twice).





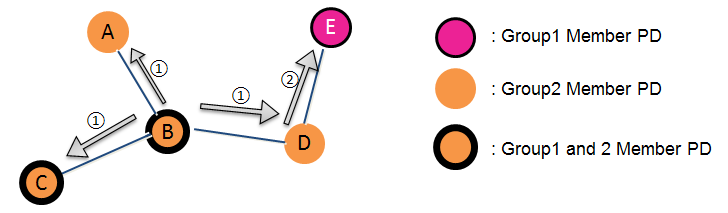
**Figure 5. Redundant MGNF Transmission**

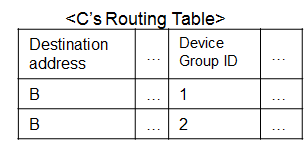
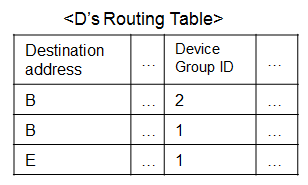
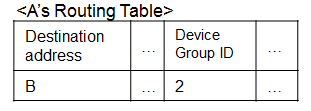
*5.10.1.3.4* *Redundant MGNF Transmission*

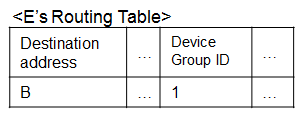
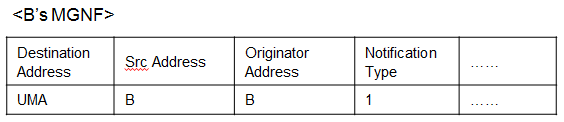
In order to prevent redundant MGNF transmission, we proposed a UMA-based MGNF transmission. In this proposal, we set the destination address fields to UMA for nodes which are member of two or more groups simultaneously. When a node receives MGNF, it checks its destination address and if it finds that the destination address is equal to UMA and two or more device group IDs that have the same value (forwarding PD for that group ID) and those device group ID have the same destination address with the originator of the received MGNF, then, the PD forwards the received MGNF.

For example, in Figure 6,

1. PD B starts to transmit its MGNF whose destination address field set to UMA for Groups 1 and 2 to A, C, and D.
2. Then A, C, and D verify the device group ID field in PD B’s MGNF, they recognize that the MGNF belongs to UMA and they forward the received MGNF. PD E receives the MGNF from D. E does not forward the MGNF. There are2 transmissions for MGNF. (B transmitted MGNF just once).







**Figure 6. UMA-Based MGNF Transmission**

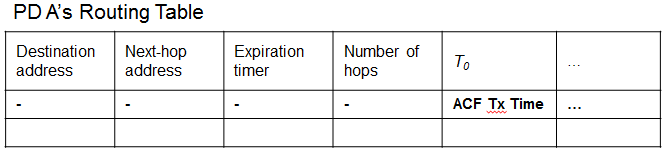
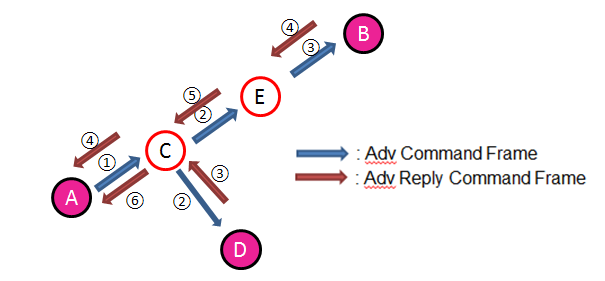
*5.10.1.4 Creation and Management of Routing Table*

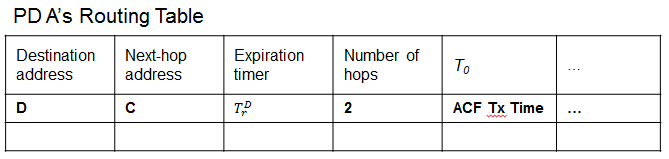
*5.10.1.4.1 Creation of Routing Table*

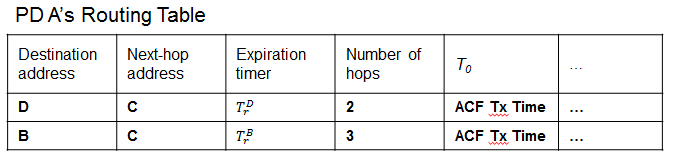
Whenever a relay-enabled PD receives ACF or ARCF, a routing entry is created in the routing table. Also, the routing table is updated by receiving MGNF. A routing table contains; *destination address, next-hop address, expiration timer, number of hops, current\_SN, device group ID*, and *last ACF reception time ()*.

For example, in Figure 8,

1. If A wants to join a multicast group, it broadcasts an ACF (assume K=4 hop). The initial routing table for A is as the following table (Note that originator of ACF has to save the ACF Tx time). When C receives the ACF, it creates the routing entry of A in its routing table.
2. Since C is not in the multicast group, C forwards the ACF to the others (e.g., D and E). When D and E receive the ACF, they create the routing entry of A in their routing table.
3. Since D is in the multicast group, D replies A with ARCF by using routing table. C creates the routing entry of D in its routing table. Suppose C forwarded an ACF at *T0*. Suppose C receives an ARCF from D at *T1.* Expiration timer of route entry to D in C’s routing table is updated to At the same time, since E is not in the multicast group, it creates the routing entry of A in its routing table and forwards to the others (e.g., B). When B receives the ACF, it creates the routing entry of A in its routing table.
4. Since B is in the multicast group, B replies A with ARCF. E creates the routing entry of B in its routing table. C forwards ARCF from D to A. A updates a routing entry with expiration timer .
5. E forwards the ARCF received from B to C. C creates the routing entry of B in its routing table.
6. PD C forwards the ARCF received from E to A. A creates a routing entry with expiration timer .
7. Finally, A is aware of route to B and D.







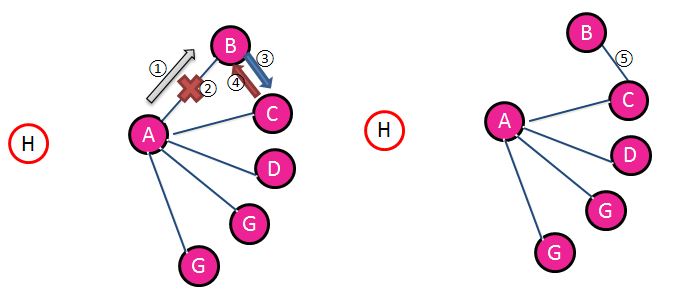
**Figure 7. Creation of Routing Table**

*5.10.1.4.2 Management of Routing Table*

When a node’s routing table is full of routing entries, it chooses a node from its routing table with shortest distance (hop-based) firstly. It sends a MGNF (type 8) to the chosen node and sets timer. When the node receives the MGNF, it breaks the link between the node and itself, and sends an ACF. The node which has timer ignores the ACF and another node which receives ACF sends ARCF to it and creates a new link between the node which sends ACF and itself.

For example, in figure 8,

1. A has limited routing entry in its routing table and H wants to join the multicast group. A chooses B in its routing tree randomly. A sends MGNF to B and sets a timer.
2. When B receives the MGNF, it breaks the link between A and B.
3. B sends ACF. A ignores B’s ACF.
4. C receives the ACF and sends ARCF to B.
5. Then C creates a link between C and B. Hereby, H can join the multicast group.



**Figure 8. Management of Routing Table**

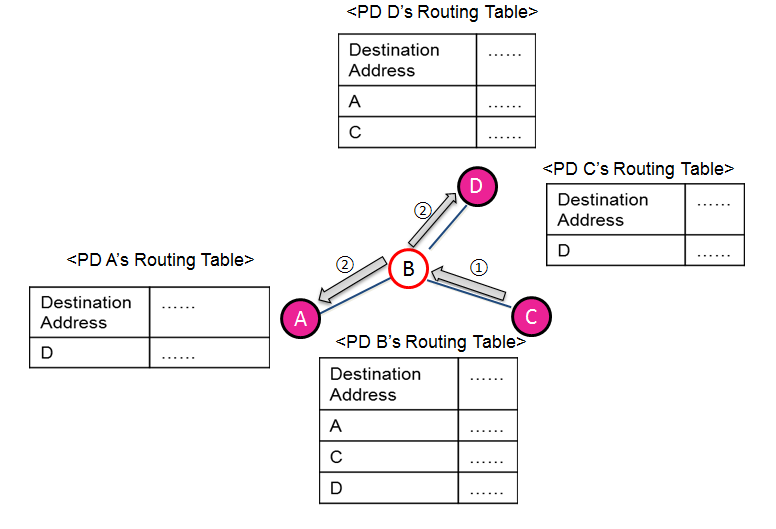
Since MGNF (notification type: 1) is sent periodically, a MGNF implosion problem can occur in the network. To prevent this problem, we proposed an adaptive MGNF transmission technique based on the distance between nodes. The distance can be measured from received signal strength indicator. (RSSI) This technique is processed when the distance between nodes becomes longer or shorter. If the node moves away from the network, the MGNF transmission gets more often since there can be a link breakage. Vice versa, if the node moves within one-hop coverage (e.g mobility support), the MGNF transmission gets rarer to prevent redundant MGNF traffic.

*5.10.1.5 Leaving From Multicast Group*

There are several reasons for a PD to leave from the network: (i) by its intention, (ii) by mobility, (iii) by limited resources. If a PD wants to leave from a multicast group, it multicasts (within K-hop) a MGNF with notification type set to 2. A recipient of MGNF may send rekeying messages to valid group members for forward secrecy. Upon receiving the MGNF, a forwarding PD deletes the entry of the originator of the MGNF, and forward the MGNF. Upon receiving the MGNF, a non-forwarding PD deletes the entry of the originator of the MGNF, but does not forward the MGNF.

In the following figure 9,

1. If PD C wants to leave from a multicast group, it multicasts a MGNF with notification type set to 2.
2. B receives the MGNF from C and deletes the entry of the originator of the MGNF, and forwards the MGNF. PD D receives the MGNF and it deletes the entry of the originator of the MGNF. Finally, PD C leaves from the multicast group.

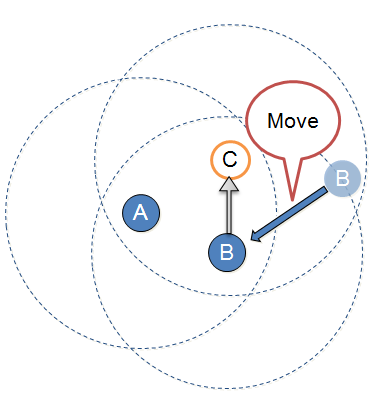


**Figure 9. Leaving From Multicast Group**

*5.10.1.6 Mobility Support for Multicast*

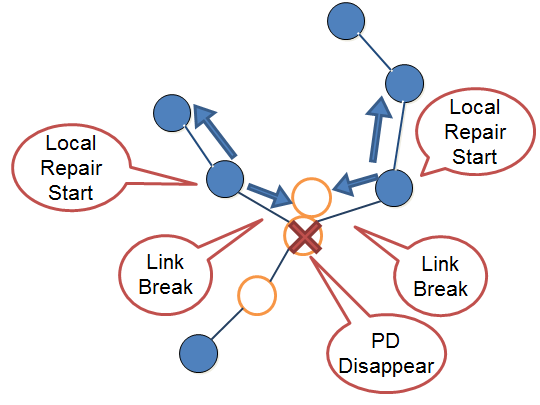
When a node moves within the one-hop coverage of another node, both of the nodes are aware of each other within the one-hop by MGNF’s TTL and they update their routing table. Then one of them sends a MGNF (notification type: 6) to another neighbor node. Therefore, it can be aware of that the nodes became closer. But, if the node is a multicast group member, it does not do anything. If it is a forwarding PD, it deletes routing entry whose destination field is the both of that nodes.

In figure 10, previously, B was 2-hop away from A. If B moves within the one-hop coverage of A, both of A and B are aware of each other and update their routing table. Then, A or B sends a MGNF (notification type: 6) to C. When C receives that MGNF, C is aware of that A and B became closer. Since C is a forwarding PD, it deletes routing entries whose destination field is A and B.



**Figure 10. Mobility Support for Multicast**

Since expiration timer is proportional to the number of hops from the impaired PD, the closest multicast group member detects link breakage. A PD starts local repair if it detects link breakage between multicast group members (due to expiration timer). Then, the PD multicasts MGNF (notification type: 7). If PDs receiving the MGNF create routing entry of originator of the MGNF during *Tw*, the PD performing local repair broadcasts an ACF within K-hop coverage.



**Figure 11. Local Repair Procedure for Multicast**

If a PD receives the ACF, it compares the receiving frame’s Device Group ID & Application type ID & Application-specific ID & Application-specific group ID with its own. If they all are same, it finds entry the originator of the ACF in its routing table. When it finds the entry, it checks the expiration timer. If the expiration timer is less than Tm, then it replies the PD sending the ACF with an ARCF. (Tm is threshold time to decide reply) In other case, the PD receiving the ACF does not reply. If any of them is not same, it decrements the TTL of the ACF and forwards the ACF.

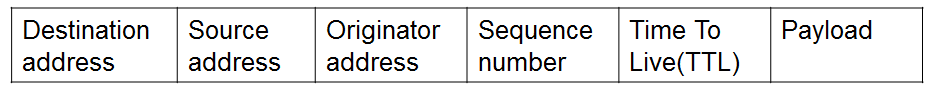
*5.10.1.7 Merging Multicast Groups*

Find/join procedure by using ACFs and ARCFs can help merging disjoint groups. If a PD in a group receives an ACF from different disjoint groups with the same Device Group ID & Application type ID & Application-specific ID & Application-specific group ID, it can initiate merging process. Then the PD replies the ACF originator with ARCF. Then, these two disjoint groups are merged by Device ID Creation Scheme. Local repair by using ACFs and ARCFs also can help merging disjoint groups. Each group member performs local repair periodically during *TL*(long duty cycle)in order to merge disjoint groups.

### 5.10.2 Multicast Data Transmission

If a PD receives a multicast data frame, it has to decide forwarding the frame or not. The PD receiving the multicast data frame compares the source address of the data frame and next-hop address entries of its routing table. PD checks the next hop addresses in its routing table. If it finds one or more next-hop entries which are not overlapped with the source address of the received frame and those next-hop entries have same device group ID with the received frame, then, the PD forwards the incoming data frame to other PDs. Otherwise, the PD does not forward the incoming frame.

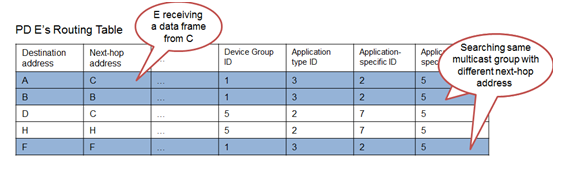
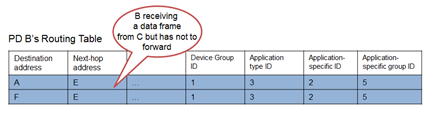
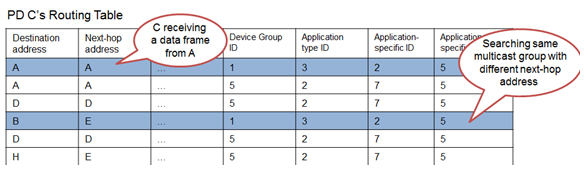
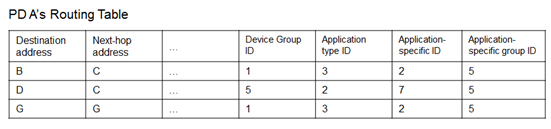
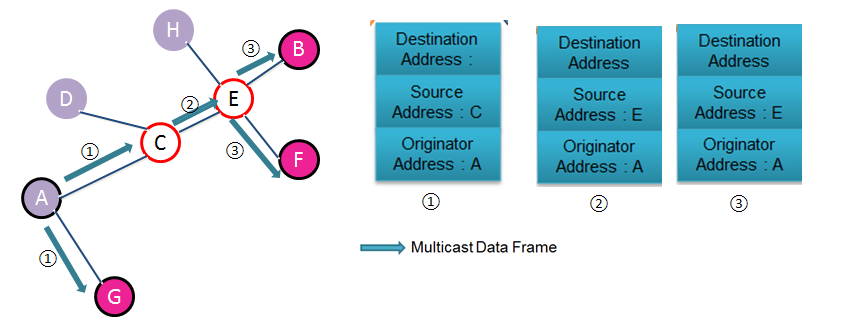
For the multicast data transmission, we have to know destination address, source address, originator address, sequence number and time to live (TTL) from multicast data frame. This information should be included in all multicast data frames. Therefore, the header in multicast data frames should contain the followings.



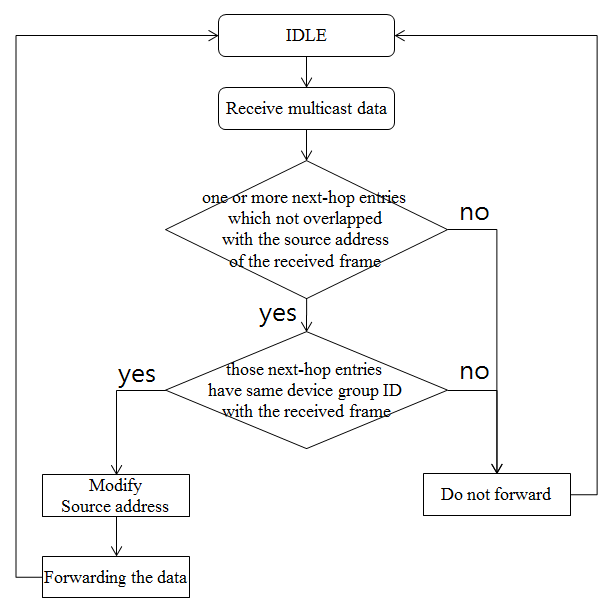
**Figure 12. Multicast Data frame**

For example, in Figure 13, we assume that 1) A, B, F, and G are in a same group which ID is 1, 2) A, D and H are in a same group which ID is 2, 3) A has routing information of B, D, and G, 4) B has routing information of A and F, and 5) D has routing information of A and H. A multicasts data frame to group 1.(Device Group ID: 1, Application type ID: 3, Application-specific ID: 2, Application-specific group ID: 5).

1. C receiving the data frame compares the source address of the data frame and next-hop address entries in its routing table which has same group IDs from the received data frame. C finds any routing entry matching with the source address of the received frame and it searches another routing entry that has same multicast group with different next-hop address fields. Since C found, it will forward the data.
2. C forwards the data. E receiving the data frame checks the condition whether forward or not. E will forward the data frame.
3. E forwards the data. B and F do not forward the multicast data frame because B and F do not have next-hop address in their routing table, except E.



**Figure 13. Multicast Data Transmission**



**Figure 14. Multicast flowchart**

### 5.10.3 Prevention Loopback Problem

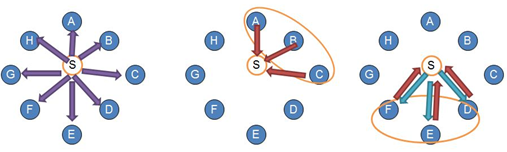
PDs can prevent multicast/broadcast loopback problem by using SN (Sequence Number) of data frames. When a PD forwards data with SN = n, the PD sets current\_SN field in its routing table to n. If the receiving frame’s SN is not greater than the current\_SN , the PD discards the frame.

### 5.10.4 Reliable Multicast

We proposed a Block ACK technique for reliable multicast to know whether the nodes receive Multicast Data Frame fully or not. Thus, when sender PD sends Multicast Data Frame, it chooses the groups in the multicast group by depending on the order of the information of the nodes in its routing table. If the sender wants to have ‘n’ groups, it reached it by dividing the numbers of all nodes in the multicast group to n. ‘n’ is decided dynamically. Then, the sender transmits Multicast Data Frame including the information of the group which sends the block ACK.

When the nodes in the group receive all multicast data frames successfully, they transmit Block ACK to the sender by notifying that they received all frames from it. If they did not receive any data frame or they did not receive it fully, they do not transmit Block ACK to the sender. If sender does not receive Block ACK or receives a negative notification, then it retransmits the data frame to the group. These steps are processed for all groups in the multicast group respectively. If the sender does not have any data frame to transmit, it sets a timer. If the timer is expired, it transmits Request ACK to the groups respectively. The advantage of sending Block ACK is that; nodes send fewer frames than when they send implicit ACK. Thus, there is less collision.

In figure 14, sender PD S transmits Multicast Data Frame to other nodes and chooses the groups. Suppose that S decided that PDs A, B and C are in group1, D, E and F are in group2, G and H are in group3. Then A, B and C transmit Block ACK to S. If S does not have any data frame for PDs D, E and F, then it transmits Request ACK, and they reply S with a Block ACK.

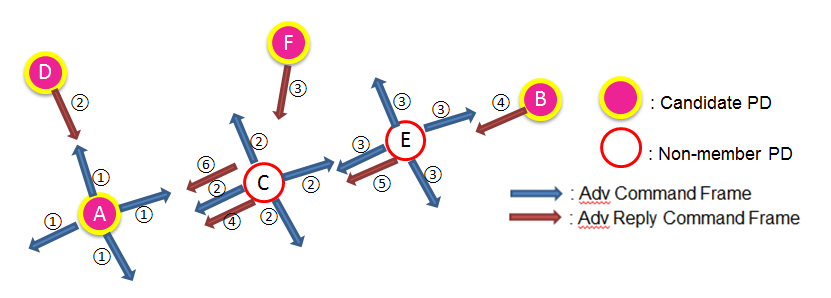
**Figure 15. Reliable Multicast**

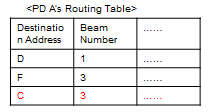
### 5.10.5 Multicast Protocol Using Directional Antenna

If a PD can support directional antenna, it has to have additional routing table field which is describes its beam number forwarding to destination in order to transmit frame. That routing table contains; *destination address, next-hop address, expiration timer , number of hops, current\_SN, device Group ID, last ACF reception time ()* and *beam number*. If each PD wants to join a multicast group, it performs find/join procedure by using all of beams. During finding/joining procedure, PDs receiving a ACF check the SNR per beams and saves the specific beam with the highest SNR measured in order to find communication beam. If non-multicast group member receives the ACF forwards the ACF by using all of beams. If multicast group member receives the ACF, it replies ARCF by using saved specific beam with the highest SNR measured to the originator of the ACF. When PD receives ARCF, it saves the specific beam with the highest SNR measured into its routing table and forwards the ARCF by using backward path. After updating routing table, all of frames, except broadcast frame, can be transmitted directionally.

For example in Figure 15,

1. If PD A wants to join a multicast, it broadcasts ACF by using all of beams.
2. Then PD D and C receive the ACF. Since D is a multicast group member, it replies ARCF by using beam 3. Then, A finds the beam with the highest SNR measured and updates its routing table. C is a non-multicast group member then forwards ACF by using all of beams.
3. Since F is a multicast group member, it replies ARCF by using beam 4. Then, C receives ARCF and updates its routing table. Also, E forwards its ACF by using all of beams.
4. When A receives ARCF from C, it updates its routing table. Also, B replies ARCF by using beam 1. Then, E receives ARCF and updates its routing table.
5. Also, E forwards ARCF by using beam 1. Then, C receives ARCF and updates its routing table.
6. C forwards ARCF by using beam 1. Then, A receives ARCF and updates its routing table.





**Figure 16. Multicast Protocol Using Directional Antenna**

## Broadcast

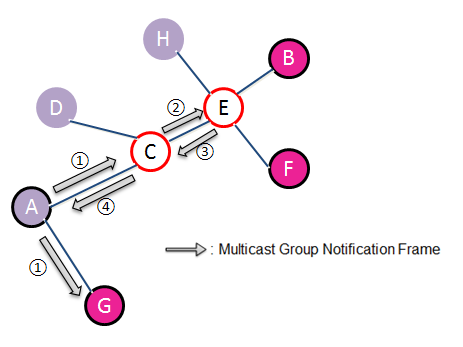
## Multi-hop operation

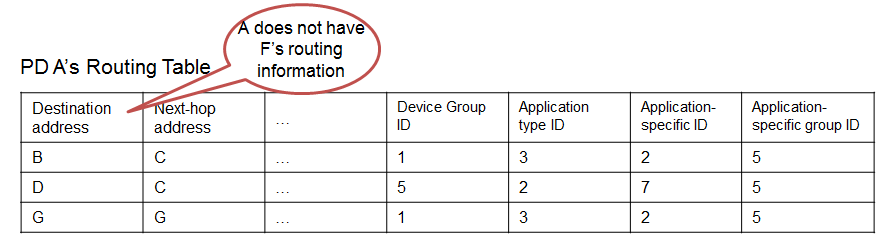
### 5.12.1 Multi-hop Unicast Data Transmission

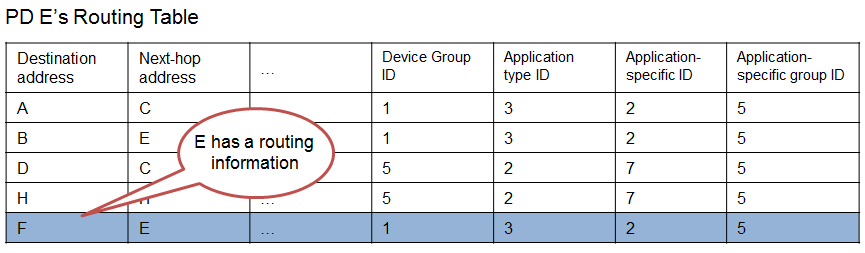
When a PD wants to unicast data frame, the PD searches routing entry of destination address in its routing table. If the PD finds the routing entry of destination address, it starts to unicast immediately. If it does not find routing entry of destination address, it multicasts a MGNF (Notification type: 4) to group. When the other PD receives the MGNF, it saves backward route information in its routing table during Tw. Then, it starts to find a routing entry of destination address in its routing table. If a PD receiving MGNF finds routing entry of destination address, it unicasts a MGNF (notification type: 5) to the PD which wants to unicast.

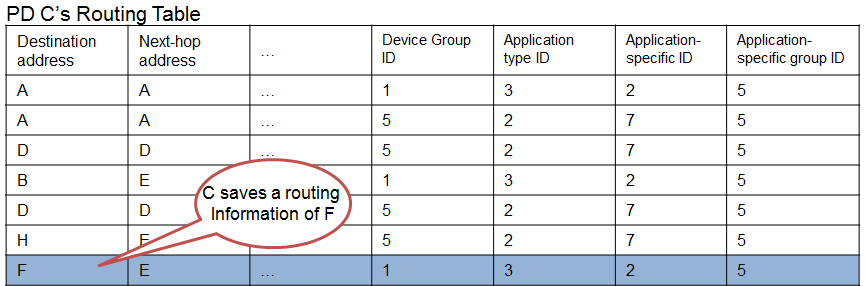
In the following figure,

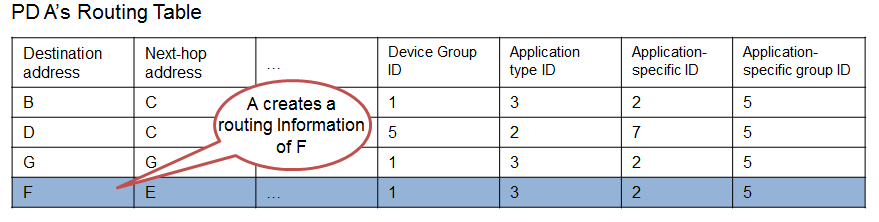
1. A wants to unicast to F within same multicast group which ID is 1, but A does not have routing information of F in A’s routing table. Then, A multicasts a MGNF (Notification type: 4) to group 1. When C receives the MGNF, it saves backward route information in the routing table during Tw. It starts to find a routing entry of F in its routing table.
2. Since C did not find the routing entry, C forwards the MGNF. When E receives the MGNF, it saves backward route information in a routing table during Tw, it starts to find a routing entry of F in its routing table.
3. Since E finds routing information of F, E unicasts a MGNF (notification type: 5) to A. C creates routing entry of F in its routing table.
4. C forwards the MGNF to A. When A receives the MGNF, it creates a routing entry of F in its routing table. Now, A can unicast data to F.











**Figure 17. Unicast Data Transmission**

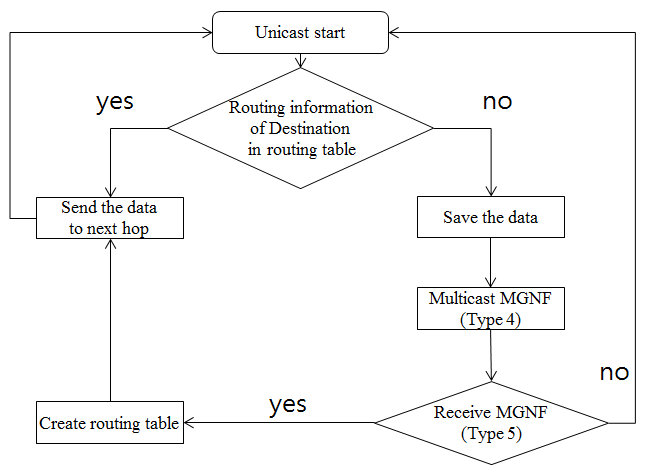


Figure 18. Unicast Flowchart

## Relative positioning

## Power management

## Security

Security layer provides users with privacy, authentication, and authorization across the network.

### 5.15.1 Security modes

PAC security layer provides three different security modes on the basis of security requirements of network connections.

*5.15.1.1 Security mode 1 (non-secure)*

When a PAC device is in security mode 1, it shall never initiate any security procedure.

*5.15.1.2 Security mode 2 (service level enforced security)*

When a PAC device is in security mode 2, it shall initiate security procedures after a channel establishment request has been received or a channel establishment procedure has been initiated by itself. Whether a security procedure is initiated or not depends on the security requirements of the requested channel of service.

A PAC device in security mode 2 should classify the security requirements of its services using the following attributes.

|  |  |
| --- | --- |
| Security requirement | Description |
| Authentication required | - Before connecting to the application, the remote device must be authenticated |
| Authorization required | - Access is only granted automatically to trusted PAC devices, or untrusted devices after an authorization procedure  - Always requires authentication to verify that the device is the right one |
| Encryption required | - The link must be changed to encrypted mode, before access to the service is possible |

Security mode 1 can be considered as a special case of security mode 2 where o service has registered any security requirements.

*5.15.1.3 Security mode 3 (link level enforced security)*

When a PAC device is in security mode 3, it shall initiate security procedures before the channel is established.

5.15.2 Security parameters

PAC security layer uses the following security parameters.

|  |  |
| --- | --- |
| Parameters | Description |
| PAC\_ADDR | PAC device address (unique for each device) |
| AUTH\_KEY | Authentication key used for authentication purposes |
| ENC\_KEY | Encryption key for secure unicast |
| GENC\_KEY | Group encryption key for secure group communication |
| RAND | Frequently changing random or pseudo-random number |

5.15.3 Key Derivation

For secure communications between PAC devices in networks, several key materials are derived using the shared secret information between the devices.

Following figure shows the key derivation procedure between devices using PIN secretly shared during the peering phase.

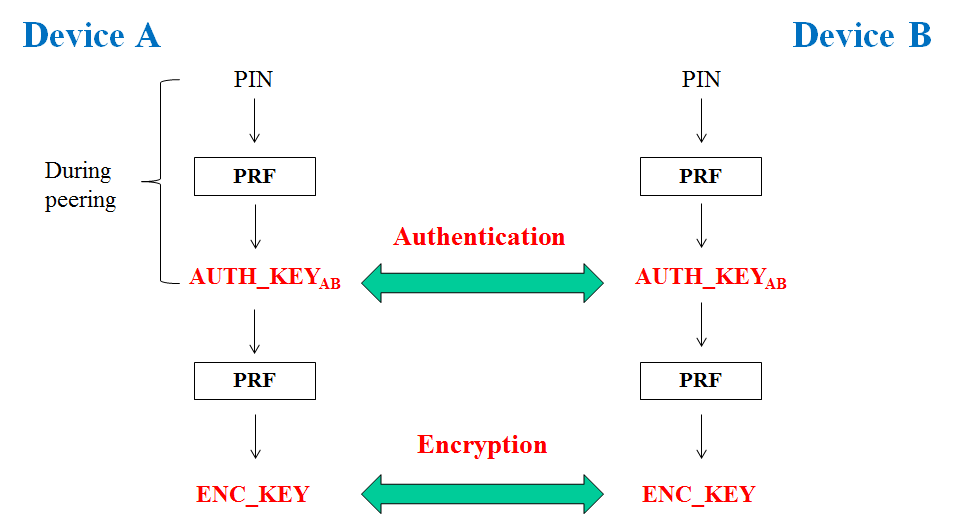


Figure 19. Key derivation

Each key material shall be derived selectively on the basis of the security mode. For example, when a PAC device is in security mode 1, it shall never initiate any security procedure. When a PAC device is in security mode 3, it shall derive all of the above key materials, such as AUTH\_KEY and ENC\_KEY between the devices.

5.15.4 Authentication

Authentication is the process of verifying ‘who’ is at the other end of the link. In PAC security layer, authentication is performed for devices, or services.

Technically, authentication is achieved based on the stored authentication key (AUTH\_KEY) or by peering (entering a PIN).

Flowchart for authentication is shown as a follow.

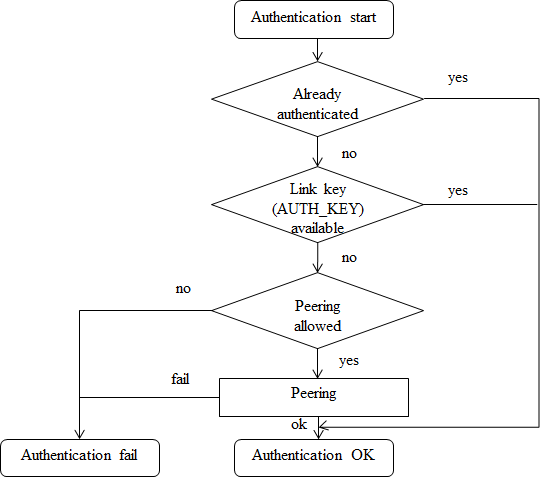


Figure 20. Flowchart for authentication

*5.15.4.1 Infrastructureless authentication*

In PAC networks where there is no coordinator or AAA(Authentication, authorization, accountability) server, authentication between PAC devices are done using PIN, or certificate issued by the trusted authority.

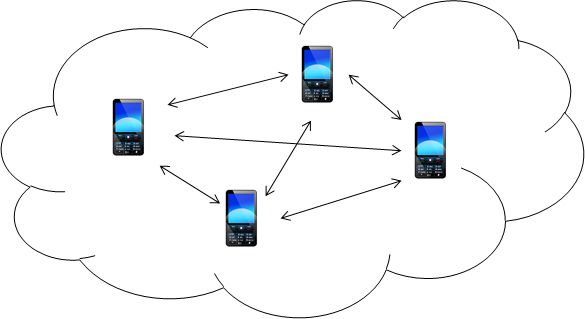


Figure 21. Infrastructureless architecture

*5.15.4.1.1 One-way authentication procedure*

Some applications might require only one-way authentication. Following figure shows the one-way authentication procedure between PAC devices A (verifier) and B (claimant).

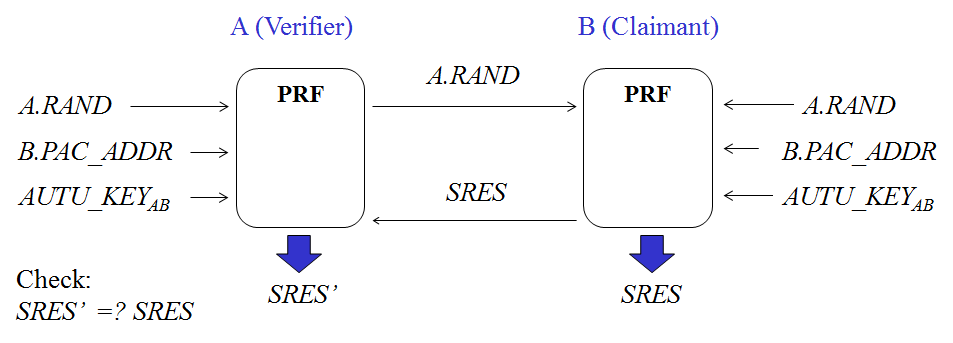


Figure 22. One-way authentication

When a verifier A requires to authenticate B, (1) A sends a random number A.RAND to B, (2) A and B computes a secret information SRES’ and SRES, respectively, using pseudo-random function(PRF) on inputs of A’s random number (A.RAND), PAC address of B (B.PAC\_ADDR), and authentication key shared between them (AUTH\_KEY), (3) B sends SRES, (4) A checks if SRES is equal to SRES’. If they match, authentication succeeds; if not, authentication fails.

*5.15.4.1.2 Mutual authentication procedure*

Some applications might require mutual authentication. Following figure shows the mutual authentication procedure between PAC devices A and B.

Figure 23. Mutual authentication procedure

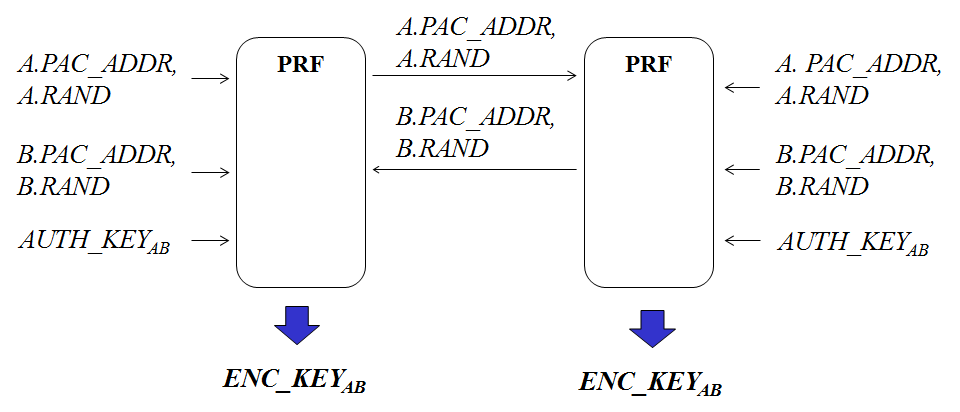


Figure 3 Mutual authentication

When PAC devices A and B authenticate each other, (1) A sends its address (A.PAC\_ADDR) with a random number (A.RAND) to B, (2) B also sends its address (B.PAC\_ADDR) with a random number (B.RAND) to A, (3) A and B optionally computes encryption key (ENC\_KEY) on the input of all of the previously exchanged information and AUTH\_KEY. Then, they shall verify each other when they communicate using a secure channel protected by the shared ENC\_KEY.

*5.15.4.2 Infrastructure authentication*

In PAC networks where there is an AAA(Authentication, authorization, accountability) server and a dynamic coordinator, which is a PAC device with intermittent connection to the AAA server, authentication between PAC devices are done using symmetric master key, or certificate issued by the AAA server

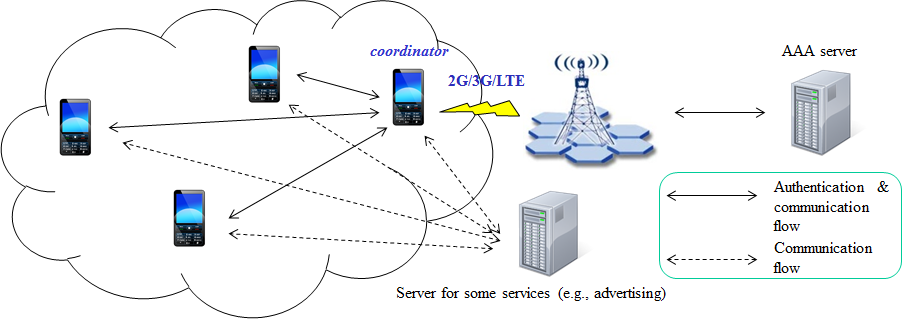


Figure 24. Infrastructure architecture

When a PAC device A and B (coordinator) authenticate each other, the mutual authentication procedure shall progress as a following figure.

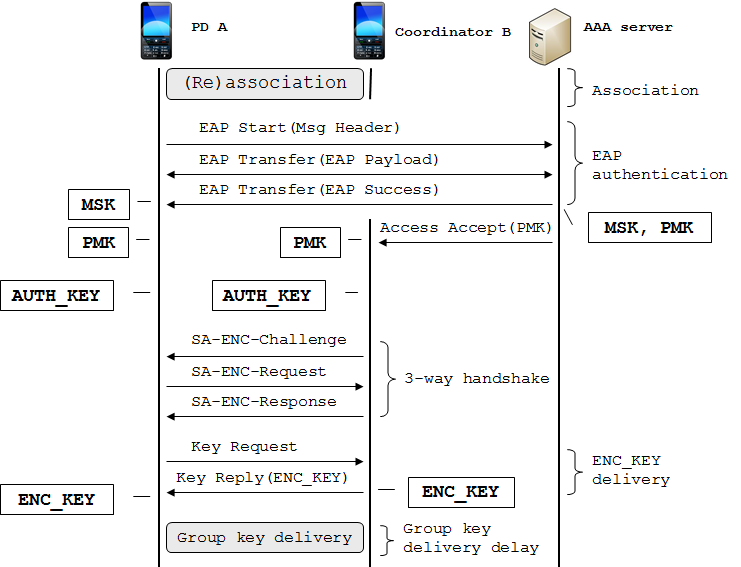


Figure 25. Infrastructure authentication

The authentication procedure consists of EAP authentication between a PAC device A and an AAA server, authentication key generation and 3-way handshake between PAC device A and B, and encryption key/group key delivery process.

*5.15.4.2 .1EAP authentication*

EAP (Extensible authentication protocol) authentication uses Extensible Authentication Protocol [IETF RFC 3748] in conjunction with an operator-selected EAP Method (e.g. EAP-TLS [IETF RFC 2716]). The EAP method will use a particular kind of credential – such as an X.509 certificate in the case of EAP-TLS, or a Subscriber Identity Module in the case of EAP-SIM.

The particular credentials and EAP methods that are to be used are outside of the scope of this specification. However, the EAP method selected should fulfill the following mandatory criteria listed in section 2.2 of RFC 4017: (1) mutual authentication, (2) protection against the man-in-the-middle attack. Use of an EAP method not meeting these criteria may lead to security vulnerabilities.

In the PAC authentication procedure, EAP yields the512-bit master secret key (MSK), which is delivered to a PAC device A by an AAA server.

Then the other key encryption keys (KEK) and HMAC/CMAC keys are derived from the MSK

*5.15.4.2.2 Authentication key generation*

After EAP authentication, the PAC device A and AAA server generate PMK using a truncation function as a follow.

*PMK = Truncate(MSK, 160)*

Then, the AAA server sends PMK to the coordinator B securely. On receipt of it, the coordinator B and the PAC device A generate authentication key using PMK.

*AUTH\_KEY = PRF(PMK, A.PAC\_ADDR, B.PAC\_ADDR)*

Then, PD A and coordinator B derive shared KEK, HMAC/CMAC key from the AUTH\_KEY.

*5.15.4.2.3 SA-ENC 3-way handshake*

SA-ENC 3-way handshake consists of the following messages.

(1) SA-ENC-Challenge message (A ← B): B.random, sequence number, PMK lifetime, HMAC/CMAC digest

(2) SA-ENC-Request message (A → B): A.random, B.random, sequence number, security capabilities, security negotiation parameters, HMAC/CMAC digest

(3) SA-ENC-Challenge message (A ← B): A.random, B.random, sequence number, [SA-ENC\_KEY-update,] SA-descriptor, security negotiation parameters, HMAC/CMAC digest

Integrity of the above handshake messages are protected by MAC digest against forgery attack. Optional SA-ENC\_KEY-update contains all the keying materials for the ENC\_KEY update and distribution, which is encrypted with KEK

SA-ENC 3-way handshake provides the following security guarantees:

* Full mutual authentication,
* Message (2) indicates to the coordinator B that a PD A is alive and that A possesses the AUTH\_KEY
* Message (3) indicates to the PD A that the coordinator B is alive
* The coordinator B is guaranteed that SA-ENC-Update is sent by the PD A and is fresh

*5.15.4.2.4 Encryption key delivery*

After a successful authorization, the PAC device A shall dynamically requests parameters for SA(security association) including ENC\_KEY through KEY\_Request and KEY\_Reply messages. When a secure communication is required between the devices, the connections are encrypted using the ENC\_KEY.

*5.15.4.2.5 Group key delivery*

In a network environment where the secure multicast and broadcast service (MBS) is supported, additional group key (GENC\_KEY) generation and delivery process shall be performed optionally after ENC\_KEY distribution procedure. Then, the group communication are encrypted using the GENC\_KEY.

*5.15.4.3 Authorization*

Authorization is the process of deciding if device X is allowed to have access to service Y.

Authorization always includes authentication, and grants access rights to devices on the basis of their trust levels.

|  |  |
| --- | --- |
| Device trust level | Description |
| Trusted device | - The device has been previously authenticated  - An authentication key is stored  - The device is marked as “trusted” in the device DB |
| Untrusted device | - The device has been previously authenticated  - An authentication key is stored  - But, the device is not marked as “trusted” in the device DB |
| Unknown device | - No security information is available for this device  - This is also an untrusted device |

Trusted devices (authenticated) are allowed to services, but untrusted or unknown devices may require authorization based on interaction before access to services is granted.

Technically, key would be derived or given (established) to the authorized user during the authorization procedure.

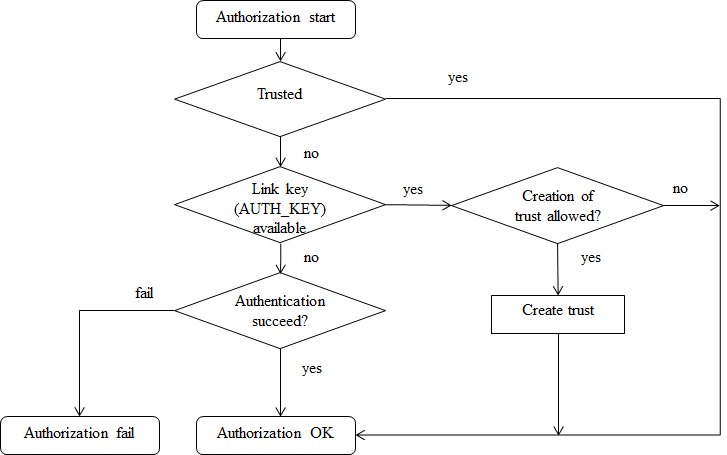
Flowchart for authorization is shown as a follow.

Figure 26. Flowchart for authorization

## Coexistence

## Higher layer interaction