Chapter 1

SHIM DIF for WiFi

1.1 Introduction

In this chapter we will stipulate the specifics for a functional Shim DIF over WiFi. Firstly we must note that this is not a fully functional DIF and this DIF will only provide support for RINA DIFs. This DIF uses the WiFi legacy protocol on which it constructs an adaptor. The purpose of this Shim DIF is to ~~represent WiFi as a DIF towards the DIF on top of this.~~ Due to this we will not try to improve or change specifics of the WiFi protocol but we will try to map them as seamlessly as possible towards ~~the~~ RIN ~~Architecture~~. The Shim DIF will be used as an adaptor, this implies that the very bottom layer will still be WiFi. The layer directly above this Shim DIF will see this Shim DIF as a DIF part of RINA. This Shim DIF will also be named the 0-DIF. As it will function as the bottom DIF for ~~the~~ RIN ~~Architecture~~.

802.2	LLC
802.11	MAC
802.11	PHY

Table 1.1: Overview of 802.11 protocol parts

WiFi consists of 3 main parts and the adaptor will try to span over all these. On top we have the 802.2 LLC layer, below that is the 802.11 MAC layer and finally we have the 802.11 physical layer. We must note that the LLC layer is an old protocol that has been reused for this 802.11, it is not likely to be updated soon. The MAC layer has been changed as recently as 2007 with 802.11e. It has several fields reserved for future

use and could be changed later on. This means that when this MAC layer is changed that these changes will have to be addressed in the Shim DIF. Finally is the physical layer presented at the bottom of the WiFi scope. We instantly note that this changed quite often but provides no use towards the Shim DIF. This physical layer will thus not be used for the Shim DIF and changes to this should not reflect in the wrap above the WiFi protocol.

The Shim DIF over WiFi is not a fully functional DIF. This means that some limitations apply to this protocol:

- Limited amount of flows ~~who are statically determined by~~ LLC header (~~802.2 standard(Society, 1998)~~).
- Type 1 Operation is the only practically useable operation mode of LLC due to restrictions of other Types of Operation.
- Limited QoS cubes provided by 802.11 MAC layer.

These limitations clearly show that the Shim DIF is not a fully functional DIF. It provides support for other DIFs to build further on. We have a limited amount of IPC process within one DIF because each one is linked to a MAC address. These MAC addresses are limited within their scope, ~~and a DIF is unique as~~ its SSID is ~~used for~~ DIF name. ~~Similar issues occur when distinguishing between flows.~~ Due to the ~~static content~~ of LLC headers we note that SAPs are also limited. Finally we note that the WiFi protocol does not guarantee any reliability for the transfer of SDUs.

1.2 Mapping of 802.11 MAC header

Since the physical layer is not to be utilized, we will start the mapping from the lowest layer that will be used. This layer is the 802.11 MAC layer. Here we will use only the header section of the layer. Furthermore ~~do~~ we focus on 802.11e, this IEEE standard from 2005 implements the most recent version of MAC. This includes the introduction of QoS, used later on in this specification. Because DIFs at the bottom of RINA are mostly exceptional compared to other DIFs we will specify the needed changes to use this DIF.

The fields in the 802.11 MAC header 1.1 that will be used are the *Address Fields* and Payload field.

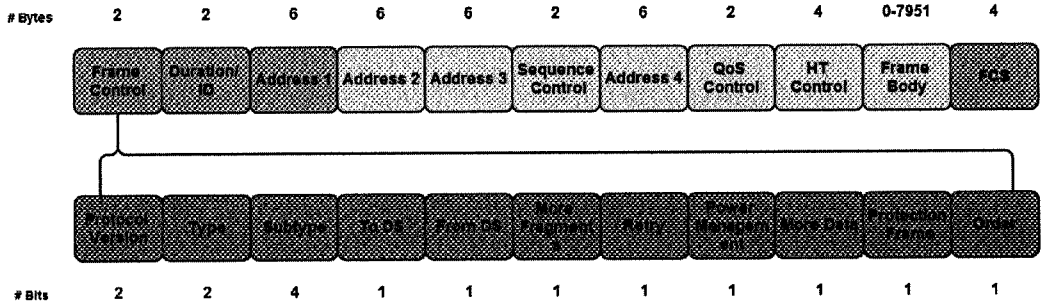

Figure 1.1: 802.11 MAC frame

These 4 address fields will be used according to the 802.11 MAC standard (Society, 2012). This means the following address fields will be mapped:

Address fields The MAC addresses used ~~to bind~~ shim IPC Processes ~~to~~. to identify

Network Type	To DS bit	From DS bit	Address 1	Address 2	Address 3	Address 4
IBSS (ad hoc)	0	0	DA	SA	BSSID	*N/A*
BSS (infrastructure)	0	1	DA	BSSID	SA	*N/A*
BSS (infrastructure)	1	0	BSSID	SA	DA	*N/A*
WDS	1	1	RA	TA	DA	SA

Table 1.2: Overview of Address Fields[2]

DON'T KNOW IF THIS IS THE BEST CHOICE OF WORDS

Destination Address (DA) The MAC address corresponding to the WiFi interface that the destination Shim IPC Process is affiliated with.

Source Address (SA) The MAC address corresponding to the WiFi interface that the source Shim IPC Process is affiliated with.

[2]Only DA and SA have use for RINA in the current development stage.

ARE USED IN THE SHIM IPC PROCESS

MAC ADDRESS = SHIM IPC PROCESS ADDRESS

Note that these address fields are not static and change according to the values in the Frame Control field, further details shall not be provided for this as this is provided in the IEEE standard.

Frame Body This will carry the SDUs it receives from the upper DIF. This can be fragmented as 802.11 supports fragmented payload[3]

For the DIF name we will use the SSID (Service Set Identifier). This name is provided by periodical advertisement in a beacon frame. Here a remark has to be made that due to this choice the DIF can become quite large. An example could be networks in universities that have the same SSID (Network Name). If this becomes problematic a solution could be to map the DIF to a combination of SSID and MAC address of the Access Point (AP).

The use of QoS cubes will be determined by the application requesting the action. This means that QoS requests come down from 1-DIF to the Shim DIF. These requests will be handled and the mapping of this QoS will be handled by the Shim DIF. Ultimately this mapping will lead to changes in the 802.11 MAC header, these changes should be conform to the 802.11e standard (Society, 2005).

1.3 Mapping of 802.2 LLC header

DSAP address	SSAP address	Control	Information
8 bits	8 bits	8 or 16 bits	M*8 bits

Table 1.3: 802.2 LLC PDU format

The 802.2 LLC header will primarily be used for differentiating between different flows from the 1-DIF. For further information and documentation on this standard we refer to the IEEE document (Society, 1998).

1.3.1 SAP addressing

SAPs will be used to distinguish between different flows, the current address assignments for SAP can be found in the table below. Every SAP is a CEP-id and this will be mapped

[3] More Fragments subfield in the Frame Control field

AFTER FLOW ALLOCATION THEY ARE MAPPED

to a port-id. This port-id forms the boundary between the 0-DIF (Shim DIF) and the 1-DIF.

Hexadecimal code	Address Assignment
0x00	Null SAP
0x02	LLC Sublayer Management
0x04	SNA Path Control
0x0E	PROWAY Network Management & Initialization
0x06	TCP/IP
0x42	Bridge Spanning Tree Protocol
0x4E	Manufacturing Message Service
0x7E	ISO 8208
0x80	3COM.
0x8E	PROWAY Active Station List Maintenance
0xAA	SubNetwork Access Protocol (SNAP)
0xBC	Banyan
0xDC	LAN Address Resolution
0xD4	Resource Management
0xE0	IPX/SPX
0xF0	IBM NetBIOS
0xF4	LAN Management
0xF8	IMPL
0xFC	Discovery
0xFE	OSI Network Layer Protocol
0xFF	Global SAP

Table 1.4: SAP Address Assignment (ISO)

As this table shows several addresses are still free and are available for use. Packets over WiFi using the Shim-DIF will be able to use the following SAP addresses:

AGAIN,

0 x	1	2	3	5	6	9	C	E	+ free to choose 4 bits

Table 1.5: Free SAPs for Flow differentiation

SHOW THE RESERVED ADRESS SPACE (ISO) GROUP ⇔ UNICAST?

This provides 128 possibilities to differentiate between flows and map CEP-ids to Port-ids. Considering this is happening at the 0-DIF level, this should be adequate for current

operations.

1.3.2 DSAP address field

The Destination Service Access Point ~~address field~~ will be used ~~for the CEP-id (Connection EndPoint-identifier) for the destination Shim IPC process on~~. For correct mapping on addressing see 1.5

1.3.3 SSAP address field

The Source Service Access Point address field will be used for the CEP-id (Connection EndPoint-identifier) for the source Shim IPC process on. For correct mapping on addressing see 1.5

1.4 Flow differentiation between port-id

Flow allocation is set up from the 1-DIF while the data transfer is handled by the Shim DIF (0-DIF). For this we utilize the mapping on the LLC header. Every flow will be distinguished by the different CEP-ids who are equal to the SAPs.

The instantiation of the flow is handled by the WiFi protocol. Once the flow has been allocated ~~it will map~~ the port-id to the CEP-ids (SAPs) ~~which provide EndPoints for this particular flow. The amount of flows is limited by the number of different SAPs one interface can be mapped to. In the current implementation we will be utilising a static map of the port-id to CEP-ids.~~

1.5 Use of Address Resolution Protocol

Address Resolution Protocol (ARP) resolves a network layer address in a link layer address. Under normal circumstances this function is almost exclusively used for mapping an IPv4 address to a MAC address. This specification will assume the utilization of the RFC826 compliant ARP implementation. A frame using this implementation has the following format:

Bit 0-7	Bit 8-15
Hardware type	
Protocol ethertype	
Hardware address byte length	Protocol address byte length
Opcode (ARP Request or ARP Response)	
Hardware address sender (n bytes)	
Protocol address sender (m bytes)	
Hardware address target (if known) (n bytes)	
Protocol address target (m bytes)	

Table 1.6: ARP frame format (RFC826 compliant)

ADRESS

This will be used to map the application IPC process name (1-DIF) to the ~~point of attachment~~ of the Shim IPC Process (0-DIF). The use of this protocol will be explained with a simple example. For this we will use the figure provided below 1.2. When IPC Process A0 (1-DIF) wants to communicate with IPC Process B0 (1-DIF) it sends information down to Shim IPC process A1. A1 receives information about B0 and sends out an ARP message with the B0 information in it. Technical details on how to construct this ARP message are presented below. If Shim IPC Process B1 receives this message it notices that it is mapped to B0. This triggers a reply with tells A1 that when A0 wants to communicate with B0 it should send data to B1. Of course is B1 represented by its point of attachment, the MAC address of the interface. This completes the flow allocation phase. Instead of using the RINA flow allocator we will thus use ARP to set up the flow.

REFER TO RFC.

(ALSO PART OF SHIM DEFINITION)

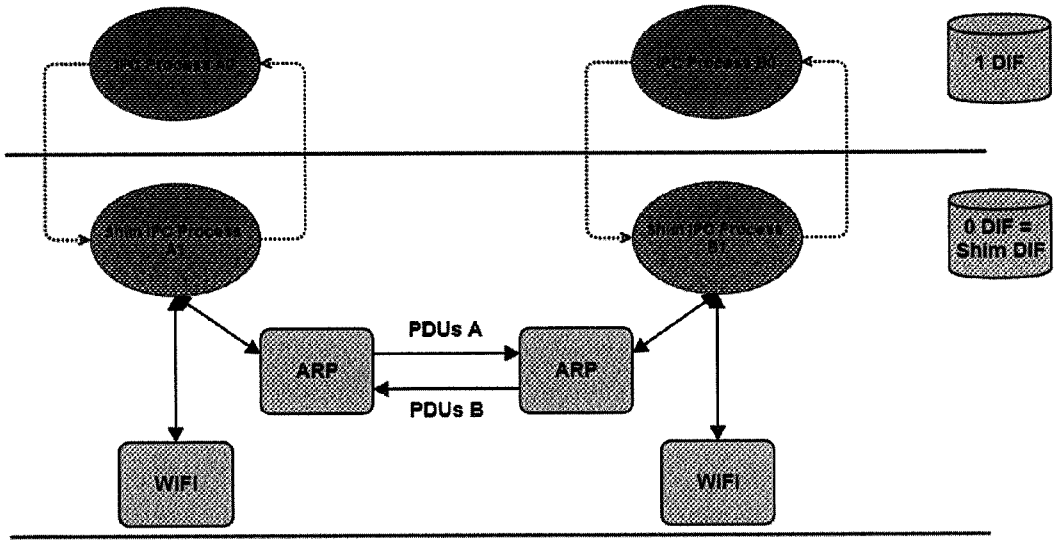

Figure 1.2: ARP example

IT IS. HOW TO KNOW THE NETWORK PROTOCOL?

~~For the technical use of this ARP mapping we note~~ that not ~~every field will be used in the ARP frame~~. The ethertype field is forexample not used in the WiFi stack and has ~~no relevance for this. The hardware addresses that are currently being used are still all 6 bytes~~. However since now the protocol address is mapped to a RINA application name, which can consist of 4 names, some modifications are required. For this reason, application names will be added together in one string with the '/' (slash) symbol used to separate the names. Furthermore is the maximum length of this total string limited to 255 characters, including the separation symbols (assuming ASCII 8bit). If more than one application name is stored, the longest one determines the length. Other application names are padded with zero bytes (0x00) until equal length to the longest application name. In case of 4 application names this means that the longest application name is limited to 63 characters. Take in consideration that we are operating at the bottom layer of the architecture and names should be kept simple at this level. While this does create some overhead we remark that these ARP frames are only being transmitted during the flow allocation phase.

USE A SYNONYM OR USE (LENGTH1)(NAME1)(LEN2)(NAME2) (LEN3)(NAME3) (LEN4)(NAME4)

1.6 Service Definition

In this section the different QoS-cubes that are supported will be addressed.

VB. 5 TEST1 1 1 10 MANAGEMENT 1 2

TEST1 ⟺ MANAGEMENT 2

1.6.1 QoS-cubes supported

The WiFi protocol supports several QoS-cubes, they are a combination of following possibilities.

ID	Depends on QoS-cube (numbered)
Name	Depends on QoS-cube (maps to ID)
Average bandwidth	QoS dependent, variable
Average SDU bandwidth	Depends on 802.11 physical standard
Peak bandwidth-duration	Depends on 802.11 physical standard
Peak SDU bandwidth-duration	Depends on 802.11 physical standard
Burst period	Depends on 802.11 physical standard
Burst duration	Depends on 802.11 physical standard
Undetected bit error rate	Depends on 802.11 physical standard
Partial delivery	Allowed
Order	Depends on 802.11 physical standard
Max allowable gap in SDUs	Depends on 802.11 physical standard
Delay	QoS dependent, variable
Jitter	QoS dependent, variable

Table 1.7: Overview of 802.11 protocol parts

BE ✓

With the addition of ACK policy this means that 5 different QoS parameters can determined:

SPECIFY CUBES, OR WRITE AS SOME BASIC BW = {54, 144, ...}

- Bandwidth
- Packet Loss Rate
- Delay
- Jitter
- ACK policy

Summing all the different options of QoS cubes would be a lengthy task. Because of this these will not be elaborated further on, but it should be stated that QoS is present in WiFi, while unavailable in ~~common~~ Ethernet. ~~Most of~~ these QoS-cubes ~~are currently determined by the type of data that is being transmitted. This currently falls~~ in 4 categories: voice, video, background and best effort. ARE CLASSIFIED

1.7 Configuration

Every Shim IPC Process is assigned to ~~one~~ A WiFi interface. This leads to a one-on-one mapping from Shim IPC Process to a specific MAC address. Before the Shim DIF can become operational it needs a basic amount of information. This information is comprised of:

1.7.1 Shim IPC Process info

Here the Shim IPC Process is given. The WiFi device name is determined by the OS and is bound to the Shim IPC Process. ~~The DIF name is announced as the SSID in a beacon frame. Note that this is only periodically announced and is not included in every frame.~~ Also the mapping of flows is stored here under *CEP-id* to *port-id* linking.

1.7.2 AP to MAC Directory

A directory is needed for the mapping of *AP name* to *MAC address*. This is a local directory that is used when sending out or replying to an ARP message. These messages occur during the flow allocation phase. This provides information about which IPC processes are accesable from the Shim IPC processes. While the directory can take any shape or form, the messages must be able to fit in an ARP packet when being transmitted during flow allocation. For technical details on ARP messages, see the segment about Address Resolution Protocol 1.5.

1.7.3 Port-id to CEP-id Directory

Other specifications such as the ~~Ethernet~~ Shim DIF specification are only limited to one flow. This is not the case with the WiFi Shim DIF, multiple flows are possible here. In the subsection about SAP mapping 1.3.1 we saw that 128 different flows can be differentiated. Each one has a 1-byte long address. This SAP address, 1 per flow per IPC process, is equal to a Connection EndPoint-Identifier (CEP-id). The CEP-id is mapped on the port-id, another local mapping. ~~During this time of the implementation we will opt for fully static mapping that is stored locally.~~

1.8 Bootstrapping

Upon creation the Shim IPC Process communicates with the OS to address the traffic from WiFi towards the Shim IPC Process. This traffic must be part of the same DIF

(SSID), the MAC address must be the same as the interface the Shim IPC process is bound to. Finally the SAP must be ~~the one registered to the port-id.~~ When these conditions are fulfilled we assume that all traffic is RINA traffic and should be handled by the Shim IPC Process.

1.9 Application (un)registration

If an Application Process (AP) ~~from 1-DIF~~ registers with the Shim IPC Process ~~in 0-DIF, it has to follow a set of rules~~. ~~Depending on this ruleset the operation is accepted or rejected~~. Note that the link that maps the AP with the Shim IPC Process is a static one as it is represented by a ARP cache entry, which is a static database entry. This entry maps the *AP name* to the *MAC address*. Since this MAC address ~~is the point of attachment~~ for the Shim IPC process ~~this indirectly links the AP to the Shim IPC Process~~. Finally when an AP un-registers the Shim IPC Process removes the entry in the ARP cache thus resulting in the removal of the link between the AP and the Shim IPC Process. During this process the technical restrictions that apply to the ARP entry should be kept under consideration 1.5.

1.10 Enrollment

All members with an interface active in the same SSID are assumed to be in the same Shim DIF. When a new member enlists in this Service Set it is enrolled in the Shim DIF. ~~Since SAPs are assigned in a static manner this leads to no extra functions for IPC Processes as it can directly map the CEP-id to the port-id on a one-to-one scale.~~

1.11 WiFi Shim IPC Process Definition

The Shim IPC process over WiFi assumes the following API from ARP:

- arpAdd(netaddress).submit: Adds a mapping of network address, in RINA the application name, to the MAC interface in the ARP table of the interface. ~~This MAC interface represents the point of attachment for the Shim IPC process~~. In essence this maps the application process to the Shim IPC process. The function returns: 'success' when the mapping was added, 'failure' when it was not.
- arpRemove(netaddress).submit: This ~~inverses~~ the action of the previous function. It removes a mapping of the application name to the MAC address in the ARP

table. If this entry is removed it returns: 'success', otherwise 'failure' is returned.

- arpMapping(netaddress).submit: Requests ARP for a mapping of a network address to a MAC address. When the mapping is found, arpMapping(netaddress, hwaddress).deliver is called.

These 3 functions are provided by ~~the~~ ARP ~~table and involve the AP to MAC directory 1.7.2.~~

Port-ids are linked to CEP-ids and are tristate variables. The 3 states of a port-id are the following:

NULL The port-id is unuseable in this state

PENDING This state of the port-id can originate from two possibilities. The port-id has initiated the flow allocation and is idle in the PENDING state until it receives an ARP reply. The other option is that it has received a request to create a flow and is currently waiting for the allocateResponse.submit function to finish.

ALLOCATED This state indicates that the flow has been allocated and the port-id can be used to read/write data to/from.

Below are all the possible functions for the Shim IPC process and finally a state diagram with said functions is presented.

1.11.1 applicationRegister(naming-info).submit

When invoked

This primitive is invoked to register an application on top of the shim IPC process. The application becomes available in the shim DIF. This primitive has to be invoked before all other functions. ARP does not differentiate between client/server, which means every application has to be available in the ARP table, even clients.

Action upon receipt

The naming-info is transformed into a single string (application-name), with the method described in segment 1.5. arpAdd(application-name).submit is called. If successful, a mapping of the application name to the hardware address of the device is added in the ARP table of the interface. The other primitives become usable. If it failed, they are not and an error is generated.

1.11.2 applicationUnregister(naming-info).submit

When invoked

This primitive is invoked to unregister an application on top of the shim IPC process. This unregisters the application in the shim DIF. No other primitives can be invoked until applicationRegister(naming-info).submit is called again.

Action upon receipt

The naming-info is transformed into a single string (application-name), with the method described in segment 1.5. arpRemove(application-name).submit is called. If successful, the mapping of the application name to the hardware address of the device is removed from the ARP table of the device. If it fails, an error is generated.

1.11.3 allocateRequest(naming-info).submit

When invoked

This primitive is invoked by a source application to request a new flow. Naming-info consists of the destination upper layer IPC Process name <AP name, AE name> (AP-name if there is a single 0-DIF; AP-ApplicationEntity if there is more than one).

Action upon receipt

If there is already a flow established to the destination application (the port-id is in the ALLOCATED state), or there is a flow pending between the source and destination applications (the port-id is in the PENDING state), a negative allocateResponse(reason).deliver is returned. If the port-id is in the NULL state, arpMapping(netaddr).submit is called. The port-id transitions to the PENDING state.

1.11.4 allocateResponse(reason).submit

When invoked

This primitive is invoked by the destination application in response to an allocateRequest(naming-info).deliver.

Action upon receipt

If the port-id is not in the PENDING state, an error is generated. If it is and if the allocate response is positive a flow has been established for this port-id; any queued

frames are delivered to the destination application. The port-id transitions to the ALLOCATED state. If it is negative the flow creation has failed and reason indicates the failure reason, all queued and future frames from this source MAC address are dropped. The port-id transitions to the NULL state.

1.11.5 arpMapping(netaddr,hwaddr).deliver

When invoked

This is invoked by the ARP protocol machine when a requested mapping becomes available in the ARP table. The Shim IPC process is supplied with the mapping of a network protocol address, application name in RINA, to a hardware address (hwaddr).

Action upon receipt

If the port-id is in the PENDING state, (there is an outstanding allocateRequest(naming-info).submit), an allocateResponse(reason).deliver is invoked, and the hardware address (MAC address) is stored for this flow. In this case the port-id transitions to the ALLOCATED state. If the port-id is in the ALLOCATED state, nothing happens. If it is in any other state, an error is generated.

1.11.6 Frame

When invoked

When the port-id is in the ALLOCATED state, a frame may be sent. Otherwise the frame is dropped.

Action upon receipt

When there is no flow for the MAC address that sent the frame, it is created, the hardware address for this flow is stored and the port-id transitions to PENDING. allocateRequest(naming-info).deliver is called. If there is a flow for the sender's MAC address, and the port-id is in the ALLOCATED state, write.deliver is called. If it is in the PENDING state, the packet is queued. If it is in the NULL state, the packet is dropped.

1.11.7 read.submit

When invoked

This is invoked by the application when it wants to read a SDU.

Action upon receipt

When the shim IPC process receives this primitive in the ALLOCATED state, it will wait for the next frame to arrive, or deliver any outstanding SDUs. It is assumed neither fragmentation nor concatenation is performed by the shim IPC Process; therefore each frame transports one and only one SDU. In any other state this operation is invalid.

1.11.8 write.submit

When invoked

This is invoked by the application when it wants to send one or more SDUs.

Action upon receipt

When the shim IPC process receives this primitive and the port-id is in the ALLOCATED state, it will create a frame, and pass it to the OS for delivery. It is assumed neither fragmentation nor concatenation is performed by the shim IPC Process, therefore the shim IPC Process uses a different frame for each SDU passed to it. It is the responsibility of the upper layer DIF to provide SDUs of adequate size in order to allow for an efficient use of the medium. In any other state, the frame is discarded and an error is generated.

NO FRAGMENTATION?

1.11.9 deallocate.submit

When invoked

This service primitive is invoked by the application to discard all state regarding this flow. It is the responsibility of both the source and destination application to invoke this primitive. It is a local event.

Action upon receipt

When the shim IPC process receives this primitive, the port-id transitions to the NULL state.

1.11.10 Corresponding state diagram

When a / is presented it means that the function, or void if empty, is called after the initial function is called.

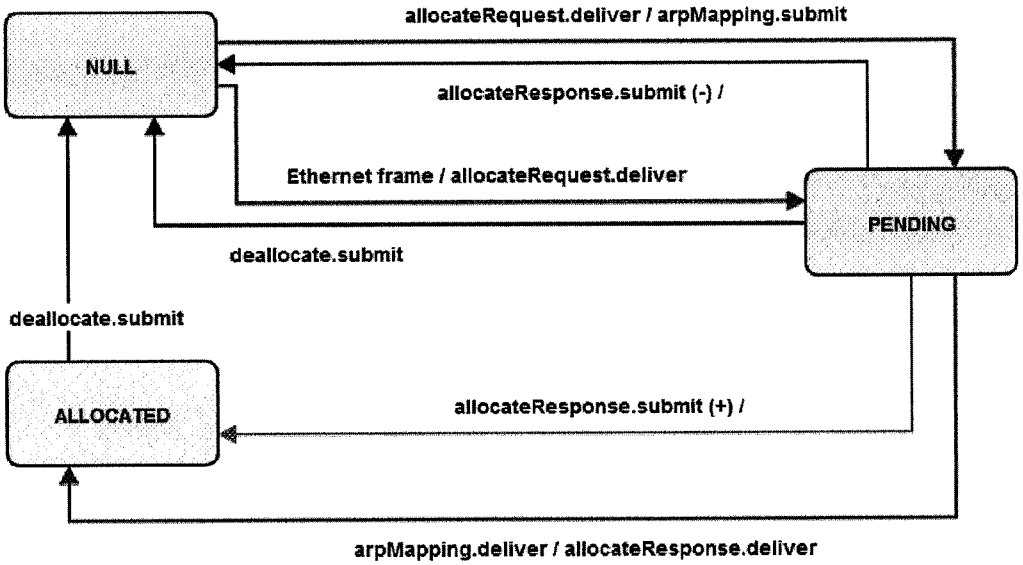

Figure 1.3: Corresponding state diagram

Bibliography

Society, I. C. (1998). Part 2: Logical Link Control. *IEEE Network.*

Society, I. C. (2005). Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements. *IEEE Network.*

Society, I. C. (2012). Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Network.*