Middle East Technical University
Department of Computer Engineering
Wireless Systems, Networks and Cybersecurity (WINS) Laboratory

ALOHANet Implementation on USRP-AHC

CENG 797 Ad-Hoc Networks
2021-2022 Spring
Term Project Report

Prepared by
Berkay Demirören
Student ID: e2309888
berkay.demioren@metu.edu.tr
Computer Engineering
18 June 2022

Abstract

In this scope of study, I will try to explain and analyze how ALOHANet (Additive Links On-line Hawaii Area) mac-layer protocol will be implemented on AHC infrasturcture that developed by Wins-Lab research group that supervised by Ertan Onur.

This study uses ahc python module as a software layer. On the bottom of that OFDM-USRP Component employed as a hardware layer. While, physical, linking and application layers are directly employed by AHC module MAC layer which is ALOHA in this case designed and implemented by myself.

Table of Contents

1 Introduction

Ad-Hoc networks are decentralized type of wireless network.It does not rely on a pre-existing infrastructure, such as routers or access points in wireless networks. Instead, each node participates in routing by forwarding data for other nodes, so the decision of which nodes forward data is made dynamically on the basis of network connectivity and the routing algorithm in use.

ALOHANet, also known as the ALOHA System, was a pioneering computer networking system developed at the University of Hawaii. ALOHAnet became operational in June 1971, providing the first public demonstration of a wireless packet data network.ALOHA originally stood for Additive Links On-line Hawaii Area.

To sum up pure Aloha protocol

• If you have data to send, send the data

• If, while you are transmitting data, you receive any data from another station, there has been a message collision. All transmitting stations will need to try resending later.

For this study ALOHANet implemented on an AHC-USRP infrastructure.

2 Background

2.1 Ad-Hoc Networks

A wireless ad hoc network (WANET) is a type of local area network (LAN) that is built spontaneously to enable two or more wireless devices to be connected to each other without requiring typical network infrastructure equipment, such as a wireless router or access point. When Wi-Fi networks are in ad hoc mode, each device in the network forwards data that is not intended for itself to the other devices.

Because the devices in the ad hoc network can access each other's resources directly through a basic peer-to-peer (P2P) wireless connection, central servers are unnecessary for functions such as file sharing or printing. In a WANET, a collection of devices, or nodes, is responsible for network operations, such as routing, security, addressing and key management.

Devices in the ad hoc network require a wireless network adapter or chip, and they need to be able to act as a wireless router when connected. When setting up a wireless ad hoc network, each wireless adapter must be configured for ad hoc mode instead of infrastructure mode. All wireless adapters need to use the same service set identifier (SSID) and wireless frequency channel number.

Instead of relying on a wireless base station to coordinate the flow of messages to each node in the network, the individual nodes in ad hoc networks forward packets to and from each other. Makeshift by nature, ad hoc wireless networks are useful where there is not a wireless structure built -- for example, if there aren't any access points or routers within range and cabling cannot be extended to reach the location where additional wireless communication is needed.

2.2 AlohaNet

The University of Hawaii used ALOHAnet to connect its campuses to one another. Each campus had a small interface computer—a hub machine—that used two distinct radio frequencies: an outbound channel and an inbound channel. In order to connect, one hub machine broadcasted packets to another computer on the outbound channel, and that computer sent data packets to the first hub machine on the inbound channel.

If data was successfully received at the hub, a short acknowledgment packet was sent back. If an acknowledgment was not received by the computer, it would automatically retransmit the data packet after waiting for a randomly selected amount of time. The mechanism detected and corrected collisions that were created when the machine and the computer attempted to send a packet at the same time.

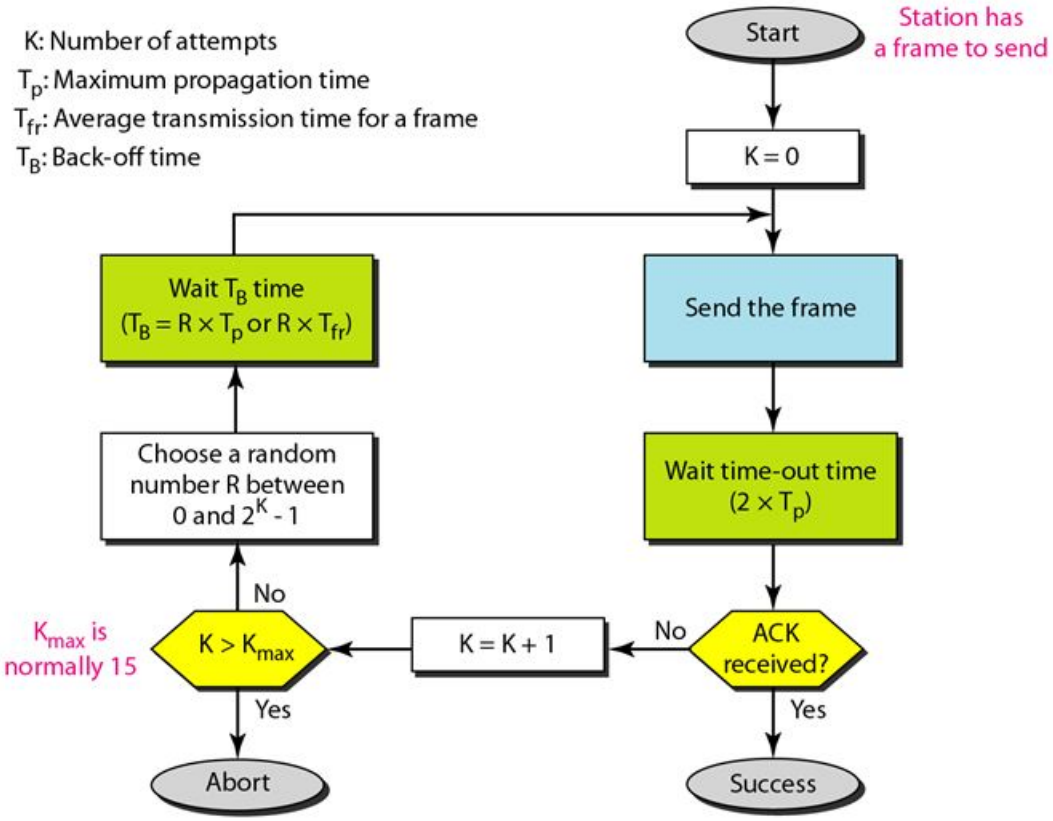
Computer networks were not well understood at the time, and it took several years for the researchers to perfect their design.

"In a sense, [the acknowledgement mechanism is] an obvious thing to do," Abramson said in the article. "But when you start off on this kind of research project, some of the obvious things don't appear as obvious as they do a little later."

ALOHAnet was connected to ARPANET via satellite in December 1972 under the guidance of the U.S. Defense Advanced Research Projects Agency. The connection allowed for reliable computer communications throughout the United States.

In June 1971 the ALOHA packet radio data network began providing inter-island access to computing facilities at the University of Hawaii. ALOHAnet was the first to demonstrate that communication channels could be effectively and efficiently shared on a large scale using simple random access protocols. It led directly to the development of Ethernet and personal wireless communication technologies.

3 Implementation



```
class AlohaEventTypes(Enum):
    ACK = "ACK"
    DATA = "DATA"


class PacketUnit:
    K_MAX = 15

    def __init__(self, event: Event):
        self.event = event
        self.K = 0
        self.send_time = time.time() * 1000

    def get_back_off_time(self):
        R = randint(0, 2 ** self.K - 1)
        return T_P ** R

    def is_expired(self):
        return self.K > self.K_MAX

    def should_resend(self):
        return time.time() * 1000 - self.send_time > T_P
```

Above classes are used to support main class

```
1
2  class AlohaNode(GenericMac):
3      def __init__(self, componentname, componentinstancenumber, context=None,
4                   configurationparameters=None,
5                   num_worker_threads=1, topology=None, sdr=None):
6          super().__init__(componentname, componentinstancenumber, context,
                                            configurationparameters,
                                            num_worker_threads,
7                           topology, sdr)
8
9          self.not_acked_packets: [PacketUnit] = []
10
11     def update_not_acked_packets(self):
12         for packet in self.not_acked_packets:
13             if packet.should_resend:
14                 packet.K = packet.K + 1
15                 packet.send_time = time.time() * 1000
16
17     def on_ack(self, ack: Event):
18         filter(lambda packet: not (packet.event.eventid == ack.eventcontent.payload
                                            or packet.is_expired()),
19                self.not_acked_packets)
20
21     def send_ack(self, eventobj: Event):
22         evt = Event(self, EventTypes.MFRT, eventobj.eventcontent)
23         evt.eventcontent.header.messagetype = AlohaEventTypes.ACK
24         evt.eventcontent.header.messageto = eventobj.eventcontent.header.messagefrom
25         evt.eventcontent.header.messagefrom = self.componentinstancenumber
26         evt.eventcontent.payload = eventobj.eventid
27         self.send_down(evt)  # Send the ACK
```

```
def on_message_from_bottom(self, eventobj: Event):
        """ Message from link physical layer message goes into inbox """
        if eventobj.eventcontent.header.messagetype == AlohaEventTypes.ACK:
            self.on_ack(ack=eventobj)
        else:
            if not self.not_acked_packets:
                self.send_ack(eventobj)
                self.send_up(eventobj)

    def on_message_from_top(self, eventobj: Event):
        packet = PacketUnit(event=eventobj)
        self.not_acked_packets.append(packet)
        self.update_not_acked_packets()
        head = self.not_acked_packets[0]
        self.send_down(head.event)

    def on_init(self, eventobj: Event):
        super().on_init(eventobj)
```

4 Conclusion

In this scope of study, I have tried to explain and analyze how ALOHANet (Additive Links On-line Hawaii Area) mac-layer protocol will be implemented on AHC infrasturcture. This study used ahc python module as a software layer. On the bottom of that OFDM-USRP Component employed as a hardware layer. While, physical, linking and application layers are directly employed by AHC module MAC layer which is ALOHA in this case designed and implemented by myself.