iOS Network Framework for UDP

Michael Ellis · Follow

2 min read · Jun 5, 2020

Listen Share

iOS 12 includes an awesome Network framework, say goodbye to some of your favorite third party libraries! I copied the docs for you from Apple:

Network: https://developer.apple.com/documentation/network

> *Create network connections to send and receive data using transport and security protocols.*

Available on these SDKs: iOS 12.0+, macOS 10.14+, Mac Catalyst 13.0+, tvOS 12.0+, watchOS 6.0+

Overview

> *Use this framework when you need direct access to protocols like TLS, TCP, and UDP for your custom application protocols. Continue to use `URLSession`, which is built upon this framework, for loading HTTP- and URL-based resources.*

Using NWConnection with UDP

Most people use TCP, but I wanted to connect to a device that uses UDP to send messages back and forth. UDP doesn't rely on connection handshakes or anything, you just send messages and you must know that sometimes packets get lost and account for that.

All you need to do is make a connection, then send and receive.

> *let connection = NWConnection(host: address, port: port, using: .udp)*

> *connection.start(queue: .global())*

It doesn't matter if the state changes because UDP is stateless. You send messages when you want, and the receiver can respond if they get the message. Apple's Network framework works a bit differently than other frameworks so you can send and receive in the same function to achieve what you may need a loop for in other frameworks.

To make a UDP Listener you can read my other article.

Here's the full code for a basic send and receive using UDP!

```swift
//
//  UDPClient.swift
//

import Network
import Foundation

protocol UDPListener {
    func handleResponse(_ client: UDPClient, data: Data)
}

class UDPClient {

    var connection: NWConnection
    var address: NWEndpoint.Host
    var port: NWEndpoint.Port
    var delegate: UDPListener?

    var resultHandler = NWConnection.SendCompletion.contentProcessed { NWError in
        guard NWError == nil else {
            print("ERROR! Error when data (Type: Data) sending. NWError: \n \(NWError!)")
            return
        }
    }

    init?(address newAddress: String, port newPort: Int32) {
        guard let codedAddress = IPv4Address(newAddress),
            let codedPort = NWEndpoint.Port(rawValue: NWEndpoint.Port.RawValue(newPort)) e
                print("Failed to create connection address")
                return nil
        }
        address = .ipv4(codedAddress)
        port = codedPort

        connection = NWConnection(host: address, port: port, using: .udp)
        connection.stateUpdateHandler = { newState in
            switch (newState) {
            case .ready:
                print("State: Ready")
                return
            case .setup:
                print("State: Setup")
            case .cancelled:
                print("State: Cancelled")
            case .preparing:
```

```
            case .preparing:
                print("State: Preparing")
            default:
                print("ERROR! State not defined!\n")
            }
        }
        connection.start(queue: .global())
    }

    deinit {
        connection.cancel()
    }

    func send(_ data: Data) {
        self.connection.send(content: data, completion: self.resultHandler)
        self.connection.receiveMessage { data, context, isComplete, error in
            guard let data = data else {
                print("Error: Received nil Data")
                return
            }
            guard self.delegate != nil else {
                print("Error: UDPClient response handler is nil")
                return
            }
```


UDPClient.swift hosted with ❤️ by GitHub view raw

```
class UDPMessageReader: UDPListener {

  var commandClient = UDPClient(address: IPAddress, port: CommandPort)

  init() {
    commandClient.delegate = self
  }

  func handleResponse(_ client: UDPClient, data: Data) {
    // Do Something
  }
}
```

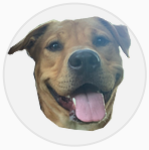
UDPClientInit.swift hosted with ❤️ by GitHub view raw

What if you want to listen to a UDP data stream?

Well here is another article I wrote about how to make a Combine enabled UDP Listener using Apple's Network Framework

https://medium.com/@michaelrobertellis/how-to-make-a-swift-ios-udp-listener-using-apples-network-framework-f7cef6f4e45f

Udp Nwconnection Swift IOS Network



Follow

Written by Michael Ellis

23 Followers

MichaelRobertEllis.com

More from Michael Ellis