DCU School of Electronic Engineering Assignment Submission

Student Name(s):	**Michael Lenehan**
Student Number(s):	**15410402**
Programme:	**B.Eng in Electronic and Computer Engineering**
Module Code:	**EE452**
Lecturer:	**J. McManis**
Project Due Date:	**12/12/2018**

Declaration

I declare that this material, which I now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited andacknowledged within the text of my work. I understand thatplagiarism, collusion, and copying is a grave and serious offence in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion, or copying. I have read and understood the Assignment Regulations set out in the module documentation. I have identified and included the source of all facts, ideas, opinions, viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the source cited are identified in the assignment references.

I have not copied or paraphrased an extract of any length from any source without identifying the source and using quotation marks as appropriate. Any images, audio recordings, video or other materials have likewise been originated and produced by me or are fully acknowledged and identified.

This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study. I have read and understood the referencing guidelines found at http://www.library.dcu.ie/citing&refguide08.pdf and/or recommended in the assignment guidelines.

I understand that I may be required to discuss with the module lecturer/s the contents of this submission.

I/me/my incorporates we/us/our in the case of group work, which is signed by all of us.

Signed: ***Michael Lenehan***

Assignment 2

Michael Lenehan

Introduction

The aim of this assignment is to analyse the 802.11ac Distributed Coordinator Function MAC level implementation, and the throughput of the 802.11ac standard. The throughput will be measured using both the TCP and UDP transfer protocols, each using one source node and one sink node. In order to demonstrate the effects on throughput of transmission to multiple clients, extra sink nodes are added to each example.

Procedure

2.1 Analysis of TCP and UDP

In order to analyse the difference between the throughput of 802.11ac in the TCP and UDP protocols, two systems must be set up, one using the TCP protocol, and one using the UDP protocol. Each system must have an equal number of nodes, an equal number of sources, and an equal number of sinks. Sink nodes must be in the same positions in both the TCP and UDP system.

In order to make the provided code compatible with ns-3.29, the constructors used for the "WifiHelper" and NqosWifiMacHelper" classes must be changed. The RAA used must also be changed, as the provided option "AarfWifiManager" is no longer supported for "HT" rates in ns-3.29. As such, the MinstrlHTWifiManager is used.

2.1.1 Setup for UDP Throughput Analysis

With the provided code, in order to correctly analyse the throughput, a FlowMonitor object is created. This object is used to output an XML file, containing all relevant information from the packet transfers between the source and sink nodes.

2.1.2 Setup for TCP Throughput Analysis

As with the UDP system, a FlowMonitor object must be created in order to output an XML used for analysis. Also required in the TCP system is the addition of the "Config::Connect()" method must be called, with the source node as an argument to the method. This is due to the fact that the source also receives packets in TCP, in the form of Acknowledgements.

2.2 Impact of Load on Performance

The impact of load on the performance of the 802.11ac standard may be analysed by adding multiple sink nodes to the TCP and UDP systems. For the purpose of this assignment, it was chosen to use a 5 Sink, 10 Sink and 15 Sink system for the comparison of each protocol.

The provided "ReceivePacket()" and "CalculateThroughput()" methods are used as arguments within "Config::Connect()" and Simulator::Schedule()" methods respectively. However, adding multiple sinks, while calling the same methods for each, will calculate total system throughput. In order to calculate throughput separately for each sink, separate "ReceivePacket()" and "CalculateThroughput()" methods must be implemented.

2.2.1 Modified UDP with multiple Nodes Code Snippets

There are a test

2.2.2 Modified TCP with multiple Nodes Code Snippets

There are a

Results

3.1 Analysis of TCP and UDP

3.1.1 Listed 802.11ac data rates

The advertised physical layer data rate of 802.11ac is approximately 7Gbps (at 5GHz). Single users may achieve MAC throughput of 500Mbps, with multiple users achievingrates of 1Gbps in total [1]. There are a number of factors within the protocol which influence the increases in throughput from the previous standards. Enhancements have been made to a number of the 802.11n standards specifications, with channel bonding (combining channels to increase throughput) being implemented both statically and dynamically, and the implementation of MU-MIMO, allowing for the increased data rates. Beamforming for MU-MIMO is also a specification of the 802.11ac standard, and the number of spatial streams, i.e. antennas, increased to eight from the previous standard of four [2].

3.1.2 Obtained Throughput

The following figure displays the throughput graph for a system using the TCP protocol, consisting of a single source node, and a single sink node.

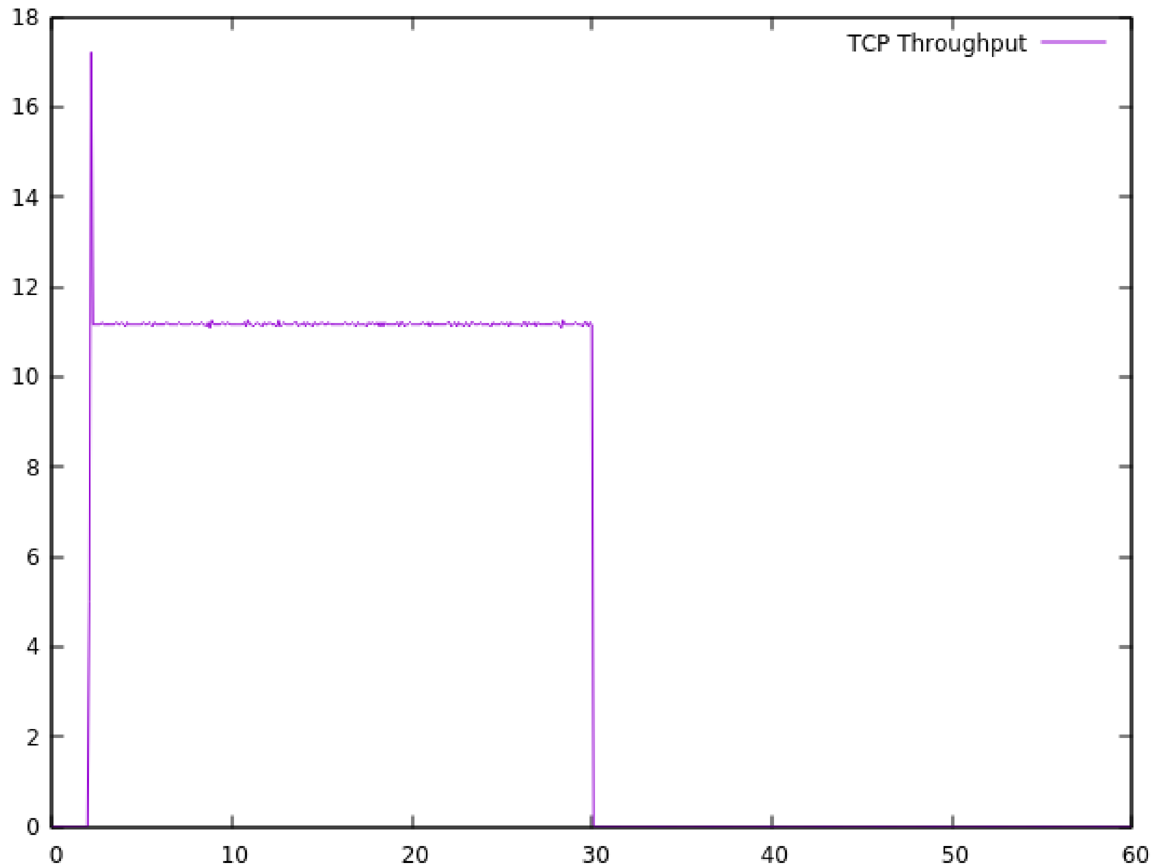

Figure 1:

The following figure displays the throughput graph for a system using the UDP protocol, consisting of a single source node, and a single sink node.

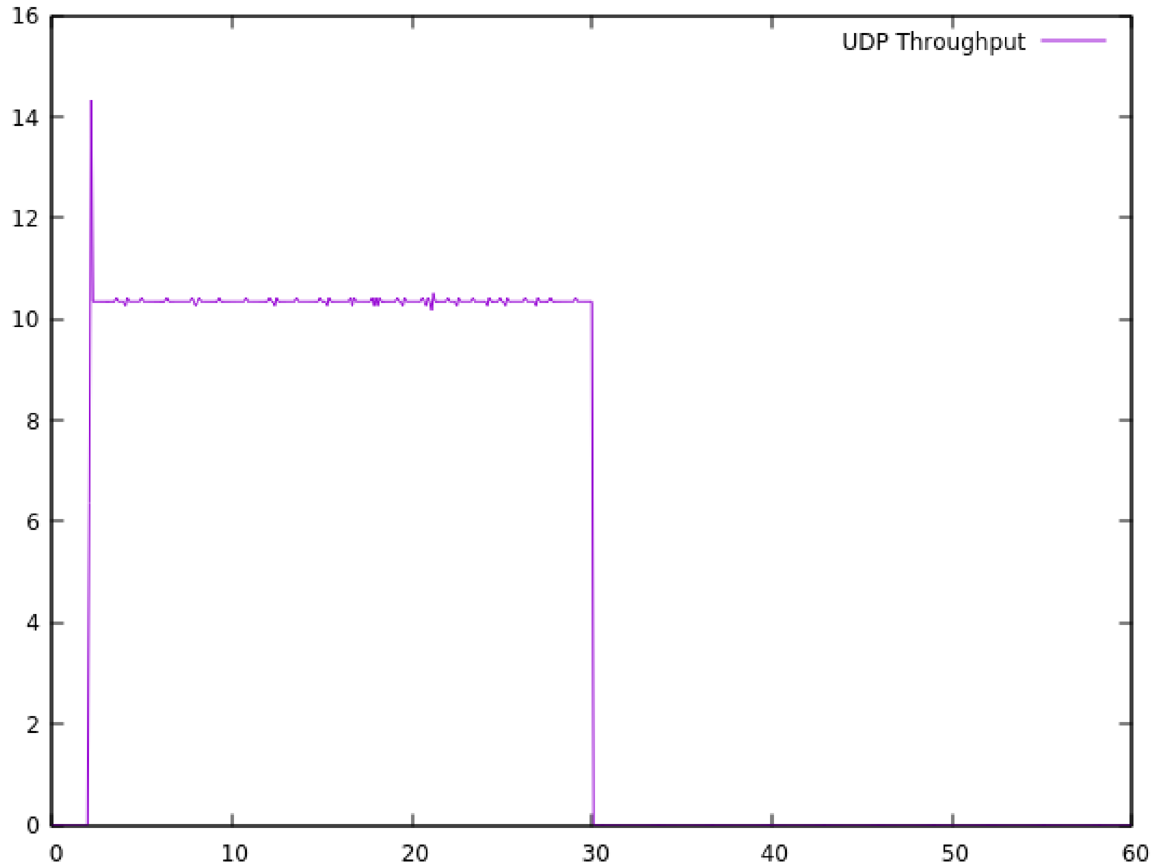

Figure 2:

3.1.3 UDP Deviation

3.1.4 TCP Deviation

3.1.5 Trace

3.2 Impact on Load on Performance

3.2.1 Throughput for All nodes in All Examples

TCP 5 Sinks The following figure displays the total throughput graph for a system using the TCP protocol, consisting of a single source node, and five sink nodes.

The individual node throughputs may be seen in the following graph, giving a more detailed view on the impact of multiple sink nodes on throughput.

TCP 10 Sinks The following figure displays the total throughput graph for a system using the TCP protocol, consisting of a single source node, and ten sink node.

TCP 15 Sinks The following figure displays the total throughput graph for a system using the TCP protocol, consisting of a single source node, and fifteen sink node.

UDP 5 Sinks The following figure displays the total throughput graph for a system using the UDP protocol, consisting of a single source node, and five sink node.

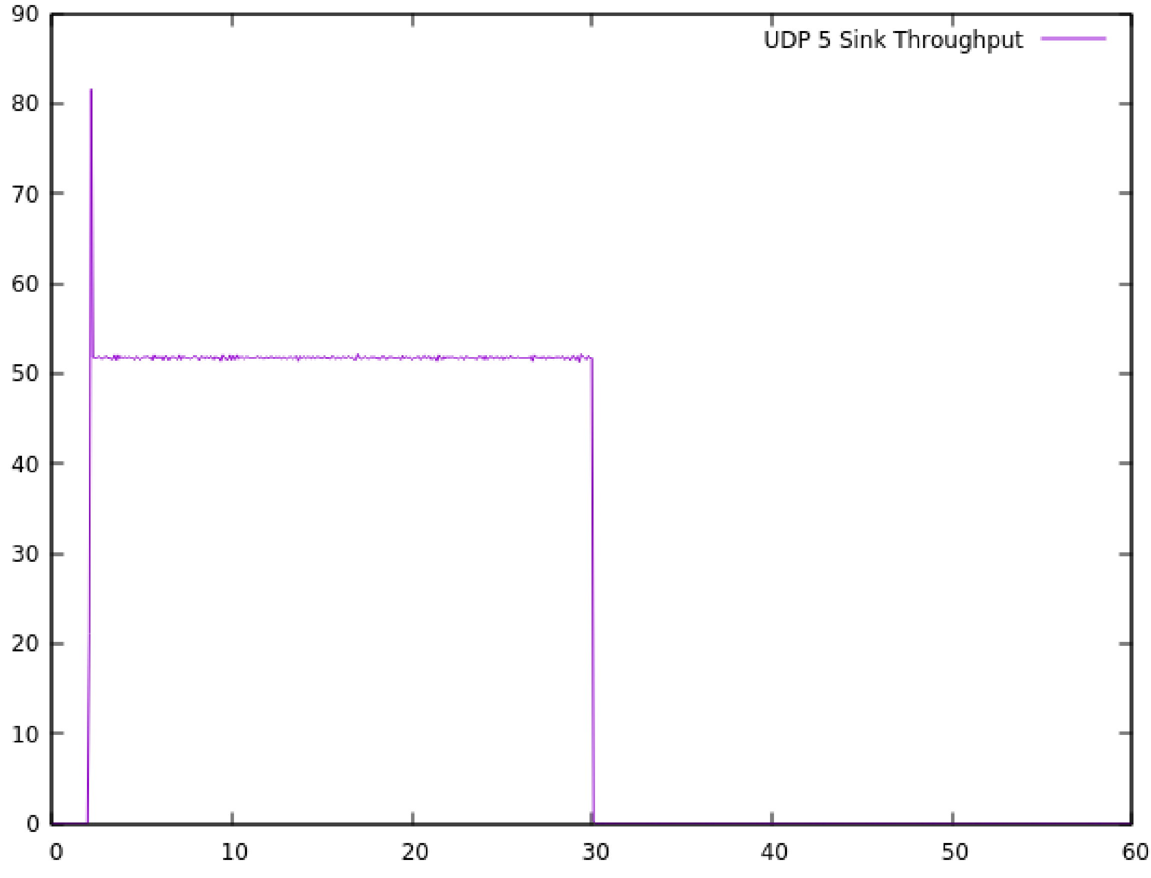

Figure 3:

UDP 10 Sinks The following figure displays the total throughput graph for a system using the UDP protocol, consisting of a single source node, and ten sink node.

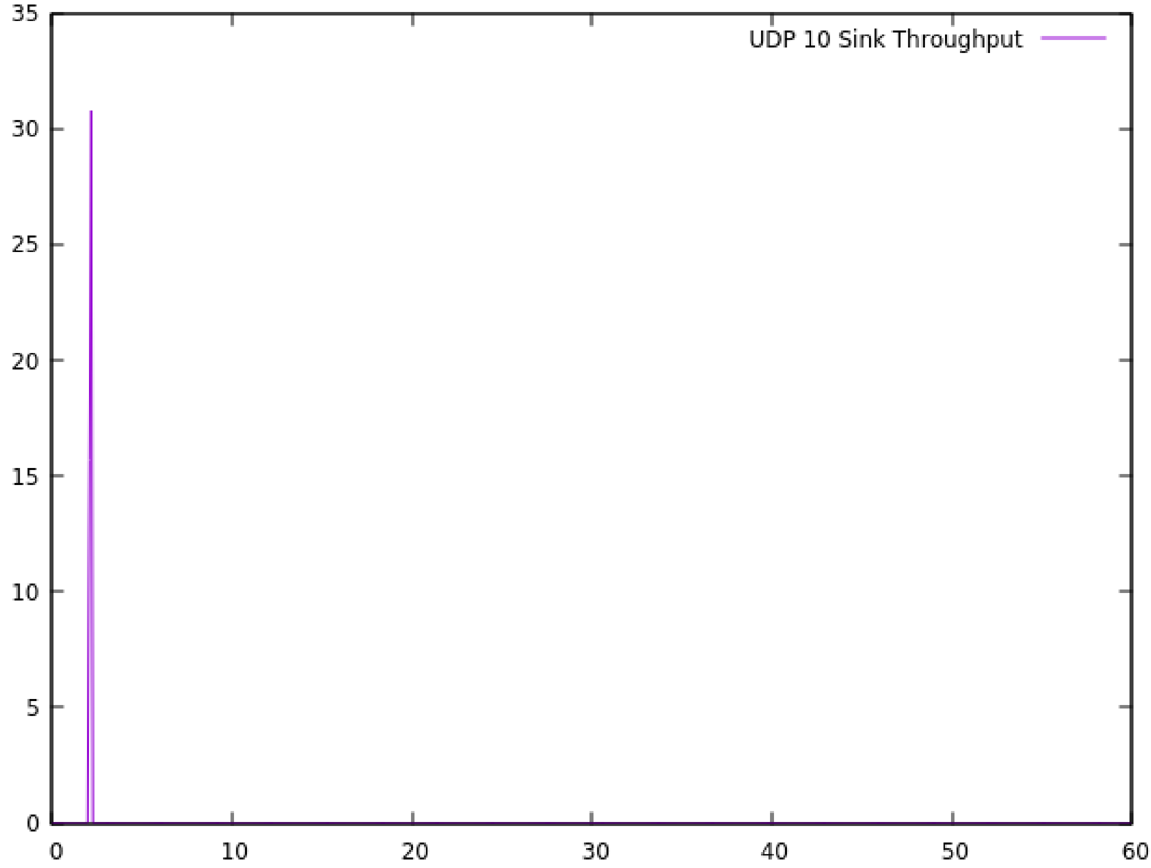

Figure 4:

UDP 15 Sinks The following figure displays the total throughput graph for a system using the UDP protocol, consisting of a single source node, and fifteen sink node.

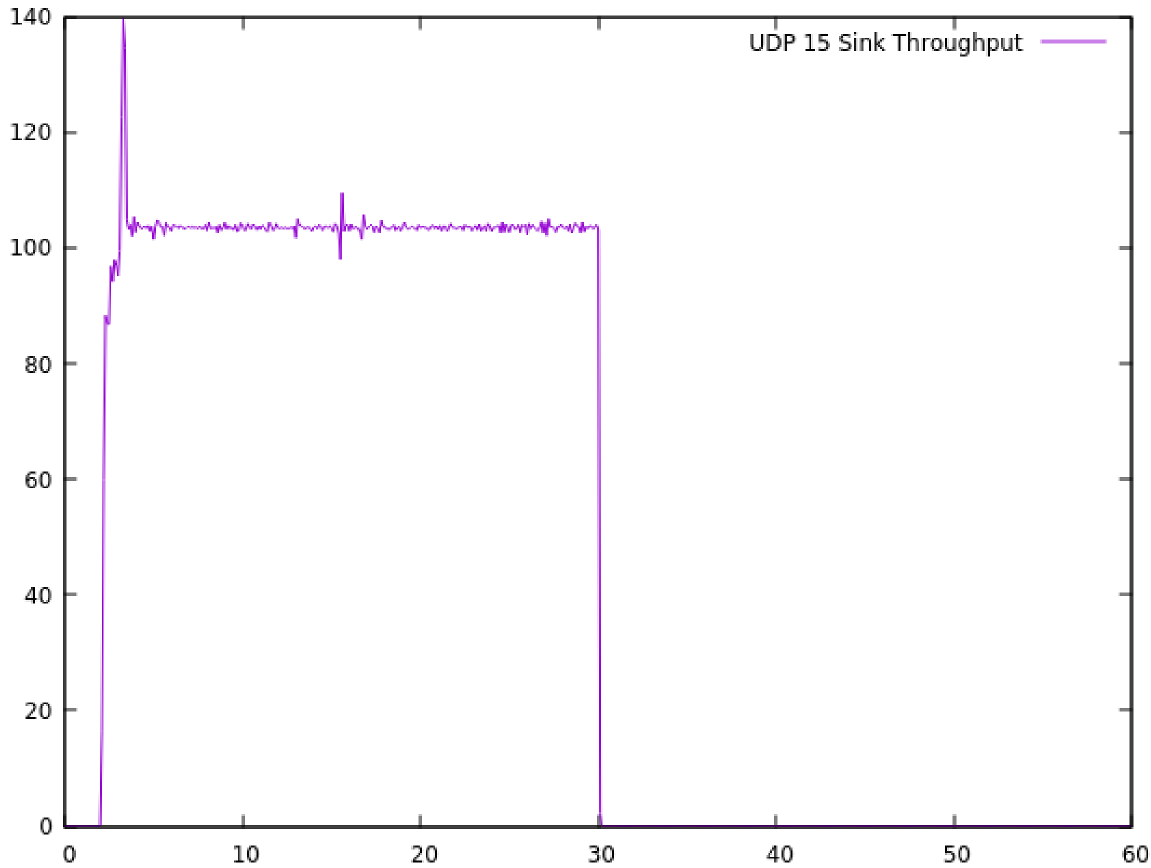

Figure 5:

3.2.2 Comparison of Results

Conclusion

Appendix

5.1 Code

The following code was used in the completion of this assignment:

5.1.1 TCP

tcpWireless1Sink

tcpWireless5Sink

tcpWireless10Sink

tcpWireless15Sink

5.1.2 UDP

udpWireless1Sink

udpWireless5Sink

udpWireless10Sink

5.1.3 udpWireless15Sink

Bibliography

[1] E. H. Ong, J. Kneckt, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtilä, "IEEE 802.11ac: Enhancements for very high throughput wlans," in *2011 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2011, pp. 849–853.

[2] O. Bejarano, E. W. Knightly, and M. Park, "IEEE 802.11ac: From channelization to multi-user mimo," *IEEE Communications Magazine*, vol. 51, pp. 84–90, 2013.