NS3 Project Report

ET4394 Wireless Networking

Andri Rahmadhani - 4505611

CONTENTS

1

INTRODUCTION

In recent decades, wireless communications have gained a lot of attentions in the area of communications, especially for developing mobile telephony [1]. WLANs, nowadays, can support a variety of industrial and home automation applications such as Internet of Things (IoT). WLANs can be divided into two main categories based on the topology architecture, namely: infrastructure WLANs Access Point (AP) and ad-hoc WLANs. In AP-infrastructure WLANs, at least one AP acts as a server in which all kind of communication pass through this AP. In ad-hoc WLANs, each node can communicate autonomously point-to-point without having to communicate through AP. Most of the network use infrastructure mode due to its simplicity and scalability [2].

As the WLANs are widely used in many areas, cross-vendor industries have emerged. Thus, a standard is needed to ensure compatibility among WLAN devices. One of the specification is IEEE 802.11. The IEEE 802.11 is a set of media access control (MAC) and physical layer (PHY) specifications for implementing wireless local area network (WLAN) computer communication in the 900 MHz and 2.4, 3.6, 5, and 60 GHz frequency bands. This specification is created and maintained by Institute of Electrical and Electronics Engineers (IEEE) LAN/MAN Standards Committee (IEEE 802). The IEEE 802.11 was released in 1997 and since then there are several modifications, for instance, IEEE 802.11b and IEEE 802.11g. IEEE 802.11b is the most popular standard of all 802.11 families. IEEE 802.11g is just an extension of IEE 802.11b, supporting higher data rates.

The IEEE 802.11b standard employs Direct Sequence Spread Spectrum (DSSS) with Complementary Code Keying (CCK) as the modulation scheme and operates on the ISM 2.4 GHz frequency band. It has 22 MHz bandwidth and provides four different data rates: 1, 2, 5.5, and 11 MBit/s. IEEE 802.11b has an indoor range around 35 m and outdoor range around 140 m.

The IEEE 802.11b uses CSMA/CA technique for transmitting a packet and avoiding collision. It works by first sensing the presence of any signal in the channel, waiting for the channel to be empty before transmitting a packet. After successfully transmitted the packet, the transmitter will receive an acknowledgement from the receiver. However, if there is a collision, there will be no acknowledgement received and the transmitter assume the packet was lost. Then, the transmitter will do the same process, sensing the channel and retransmitting the packet whenever the channel is empty after a certain amount of time, which is called backoff time. Thus, it makes sense that the more users in a particular area, the greater the probability of occurrence of collisions, which may lead to the decreasing throughput.

As the IEEE 802.11b uses DSSS in which one frequency channel can be occupied by many users, power and interference become two dominant factors affecting the throughput [3]. For instance, a user that is located far away from an AP will get a lower signal level and probably get a lower throughput than a user that is located near AP. However, IEEE 802.11b provides advanced scheduling and traffic control [4]. Therefore, in this project, the density of users, location, and mobility that predominantly affect the througput will be investigated.

2

Project Description

2.1. Objective

In this project, the density of users, location, and mobility that predominantly affect the fairness of the throughput of the network will be investigated. Moreover, it is quite interesting to consider several network parameters regarding the flow of packets, such as data rate, and payload size. Probably, such kind of parameters may also affect the performance of the network.

2.2. Hypothesis

The increasing number of users or nodes, in any circumstances, will reduce the throughput and thus lower the performance of the network. The reason for this is that the probability of contention in available channels is increasing whenever the number of nodes is also increasing. Contention will add delay and thus lower the throughput. Theoretically, the maximum throughput will be achieved whenever there is only one node inside the network. Once new nodes are added to the network, the network capacity will be shared across all nodes, reducing the average throughput of the network.

3

Design and Implementation

In this project, the performance of IEEE 802.11b will be examined through simulation using NS-3. NS-3 is a discrete-event network simulator for Internet systems and is an open source software licensed under the GNU GPLv2 license [5]. NS-3 is programmed in C++, but it has a feature for binding with Python. We have created a Python script to plot the result from NS-3. The script along with the main NS-3 application is automatically triggered through a shell script we made. Thus, it makes the simulation setup faster.

The infrastructured WLAN network with one AP is selected for this simulation and N-number of nodes. The topology of this network is depicted in Figure 3.1.

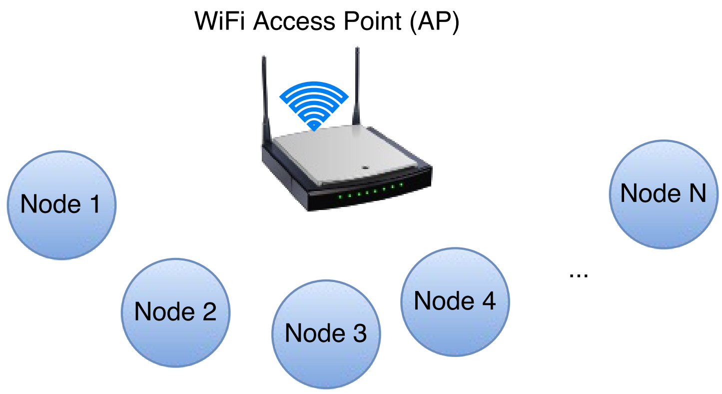

Figure 3.1: Infrastructured WLAN topology with 1 AP and N-clients

In order to build WiFi simulation in NS-3, first, we can make abstraction of the modules based on network layers just like in the OSI model. This is quite helpful when we start writing NS-3 program and also when we make a modification to the program. As a start, we define parameters of the simulation either define statically or dynamically by reading input arguments from users. Then, we specify the number of nodes (stations) and AP that will be used in the simulation. This can be done by using NodeContainer class.

After creating the containers, the physical layer (PHY) configuration has to be added to the containers. YansWifiPhyHelper and YansWifiChannelHelper are the NS-3 helpers that can be used to configure IEEE 802.11b standard. Here we can set the propagation loss model for the simulation.

For data link layer, the MAC parameter configurations can be added to select the data rate offered by the standard. The default value of this configuration is 11 Mbps with Constant Rate mode. We also specify SSID and MAC address of the nodes and AP in this step.

One of the most important configurations used in this project is the mobility configuration. There are several mode that can be selected for simulation, such as constant mobility model, which is static, and 2D random walk model, which is dynamic. Using constant mobility model, we have created two different scenarios for placing the nodes, one is uniformly distributed across a circular path, and one is randomly distributed within a disc shape. For the circular path, we use a custom function to generate exact location of the nodes so that the nodes have the same relative distance to AP. Note that the AP is placed in the center of the circle. For the randomly distributed disc, we use a built-in NS-3 class, i.e. RandomDiscPositionAllocator, that is used to allocate random positions within a disc according to a given distribution for the polar coordinates of each node with respect to the provided center of the disc [6].

The next step is configuring the network layer. We configure and assign IP addresses to the devices by using InternetStackHelper and Ipv4AddressHelper.

After configuring network layer, the next step is to configure transport layer. We use TCP for the simulation and set the AP as a transmission sink. Thus, all hosts try to send the data as much as possible to the sink during simulation time. Communication between AP and each node is established through a different port to prevent delay in case of the busy port. The traffics are generated with OnOff mechanism that follows On/Off pattern. The duration of ON state is equal to simulation time whereas the duration of OFF state is 0. This causes the packet generated constantly in each node.

It is interesting to see why TCP packets are used for this simulation. In contrast with UDP, TCP has a better mechanism to reduce the number of packet loss due to collision by performing three-way handshake. Thus, throughput calculation is more accurate than using UDP.

The effect of RTS/CTS mechanism is also evaluated in the simulation. By setting the RTS/CTS threshold value, we can either enable or disable the use of RTS/CTS packets. As the default RTS/CTS threshold value is set to 150 bytes and the payload size is 1024 bytes, RTS/CTS mechanism will be used. To disable this, the threshold value should be set above the payload size. For instance, the threshold value can be set to 2000 bytes.

To measure the throughput, FlowMonitor, which is a built-in class in NS-3, is used. Essentially, FlowMonitor is utilized to monitor packet flows in the simulation. The throughput itself is defined as the sum of data received by the sink, in this case, the AP, divided by transmission time. Note that transmission time is the difference between timestamp of last packet and timestamp of first packet. However, NS-3 simulator will give the same result once we run the same set of simulation. One of the solutions is by adding randomness using seed functions. This can be done by using the RngSeedManager class.

To recap, the scenario that will be evaluated in this project are:

1. Throughput vs Node placement and mobility
2. Throughput vs Payload size
3. Throughput vs the use of RTS/CTS

4

Results and Analysis

4.1. Throughput vs Node Placement and Mobility

As stated in previous chapter, we performs three simulations with different types of mobility, namely static circular location, static random distribution within a disc, and dynamic 2D random walk. The result from simulation scenario throughput vs node placement and mobility can be seen in Figure 4.1.

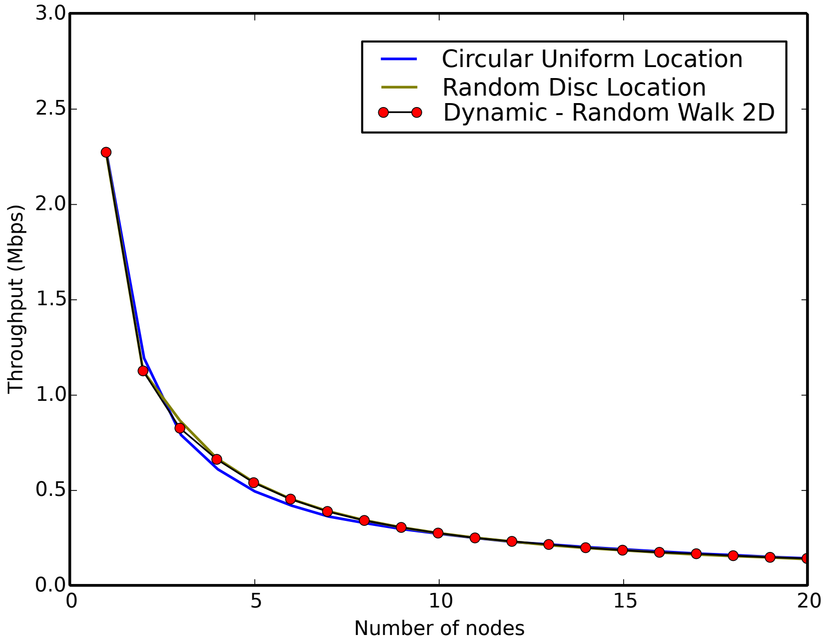

Figure 4.1: Average throughput on different types of mobility

From Figure 4.1, it is obvious that for all modes, the average throughput goes down as the number of nodes incerases. However, there are only slight differences among them. Thus, we need additional information in order to get the valuable results. The value of standard deviation from each of simulations can be used as a solution. Figure 4.2 depicts the standard deviation for three different modes of node mobility.

We can see from Figure 4.2 that the static circular placement of nodes gives low standard deviation for the total number of nodes below 10 whereas above 10, the standard deviation values are quite similar with the other modes. This is because for a small number of nodes, the circular placement is distributed uniformly so that the distance between AP and nodes are equal.

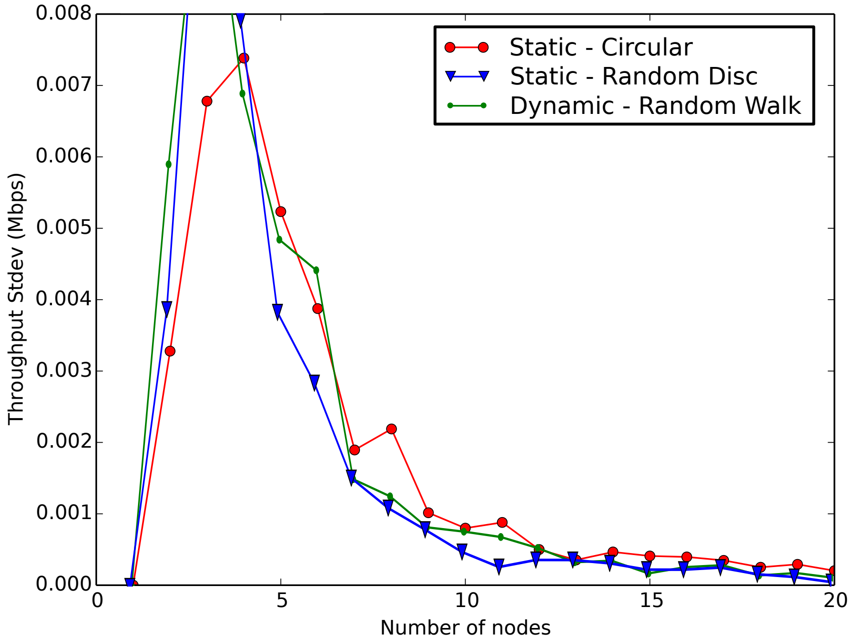

Figure 4.2: Standard deviation of throughput on different types of mobility

4.2. THROUGHPUT VS PAYLOAD SIZE

Figure 4.3 depicts the effect of payload size to the average throughput of the system.

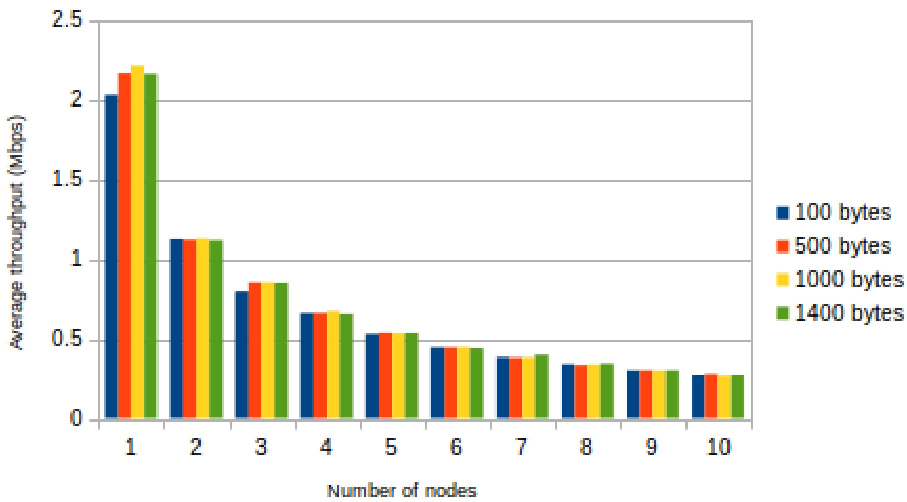

Figure 4.3: Throughput with and without using RTS/CTS

We can see from Figure 4.3 that for small number of nodes, the throughput is lower when we use 100 bytes as payloads. For large number of nodes, the higest payload gives the better results. This happen because when the payload size is decreased, the transmitted bytes (Tx) will also decrease, reducing the number of received bytes (Rx). Thus, as throughput depends on Rx bytes and the simulation time is made constant, the throughput is decreasing. To tacke this problem, we can use the maximum value of payload size.

4.3. THROUGHPUT VS THE USE OF RTS/CTS

Figure 4.4 shows the effect of RTS/CTS that reduces the throughput. This happens because RTS/CTS adds more overhead to the system, hence lowering the throughput.

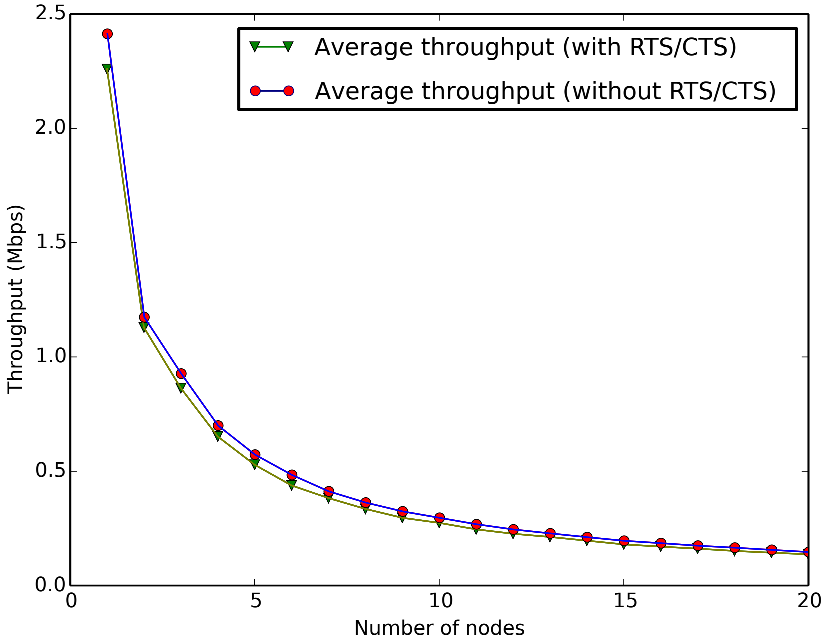

Figure 4.4: Throughput with and without using RTS/CTS

Basically, the use of RTS/CTS can increase the throughput. However, in this simulation, the throughput is lower than expected. There are two reasons that probably triggers this problem. First, TCP has three-way of handshaking mechanism to reduce collision. Thus, RTS/CTS does not give a big impact on the average of throughput for a system with a large number of nodes. Secondly, since only one AP is used in this simulation and such AP covers all of the nodes in the system, there is no hidden problem occurs. Also, the exposed node problem does not exist. Therefore, RTS/CTS does not give any advantage to the increasing number of throughput.

5

CONCLUSION

From the simulation results, we can conclude that in any circumstences, the average throughput of the system goes down as the number of users increases. This because the nature of shared medium, the more the user, the smaller average throughput. The use of RTS/CTS, which is part of IEEE DCF, is intended to solve hidden and exposed node. In a system where hidden and exposed node problem is non-exists, it gives the lower throughput instead of giving advantage to the performance of the system.

References

[1] A. Athanasopoulos, E. Topalis, C. Antonopoulos, and S. Koubias, *Evaluation analysis of the performance of ieee 802.11 b and ieee 802.11 g standards,* in *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on* (IEEE, 2006) pp. 141–141.

[2] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, *Ieee 802.11 wireless local area networks,* Communications Magazine, IEEE **35**, 116 (1997).

[3] V. P. Kemerlis, E. C. Stefanis, G. Xylomenos, and G. C. Polyzos, *Throughput unfairness in tcp over wifi,* in *WONS 2006: Third Annual Conference on Wireless On-demand Network Systems and Services* (2006) pp. 26–31.

[4] G. Bianchi, *Performance analysis of the ieee 802.11 distributed coordination function,* Selected Areas in Communications, IEEE Journal on **18**, 535 (2000).

[5] NS-3, *What is ns-3,* (2015).

[6] NS-3, *ns3::randomdiscpositionallocator class reference,* (2016).