Report Lab 1 Group C3

Niclas Scheuing and Vasileios Dimitrakis

October 13, 2015

1 Introduction

This lab is about studying the throughput using a 802.11b Wi-Fi network and different data rates, back-off policies or number of nodes. The IEEE 802.11[2] was originally defined in 1999 and uses DSSS on the license-free 2.4GHz band and offers 14 sub-bands of 5MHz called *channels*. The MAC layer of this standard provides functions for controlling the radio channel access. The DCF (Distributed Coordination Function[1]) is a fundamental MAC technique of IEEE 802.11 and employs a CSMA/CA with binary exponential back-off algorithm.

The theoretical throughput is 11MiB and the CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance) method is used for sharing the wireless medium, as we discussed before. The RTS/CTS scheme for avoiding the *hidden terminal problem* is not used in this lab but a random back-off time. This back-off time is a random time interval, bounded by the contention window size, a node has to wait between sensing that the channel is idle and starting its transmission. If during this back-off time another transmission takes place, the remaining back-off time is freezed, until this transmission is over and the ACK has been sent. When the node has waited for the full back-off time, it starts transmitting. In case a collision takes place the node waits for the *ACK-timeout* and gets a new back-off time with an increased contention window size.

2 Materials and Methods

Each student worked on PC equipped with a WiFi adapter and running an Arch Linux environment. For task 4.2.1 we were working in pairs and for the other tasks in groups of four. The following tools were used:

- *ifconfig*: Is used to configure the network interfaces.

- *iwconfig*: Is used to configure the *wireless* network interfaces.
- *iperf*: Network testing tool for creating data streams and measuring throughput.
- *Wireshark*: Packet analyzer.

3 Terminology

The machines serving as wireless nodes are denoted by *Node 1*, *Node 2* and *Node 3*. The fourth machine used as access point and server is called *local AP* in the following, while the access point used by the whole class is named *global AP*.

4 Results

In the following, we provide the results of the conducted simulations.

4.1 Task 4.1.10 Q/A

What are all the different modes that iwconfig offers and what do they do? Taken from iwconfig manual[3].

- *Ad-Hoc* Used for networks composed of multiple nodes without access point (AP).
- *Managed* Used for nodes connection to an AP in an environment that is using multiple APs.
- *Master* Acting as AP.
- *Repeater* Node forwards packets.
- *Secondary* Acts as backup master or repeater.
- *Monitor* Node is passive (not sending) and captures all packets received on the frequency used.
- *Auto* Magic!

Why did we set the channel in monitor mode? We need to select a specific channel (frequency range) that we want to observe since observing all channels is not possible or desired.

What's the difference when capturing in promiscuous mode and non-promiscuous mode? According to the wireshark manual page[4]:

- In promiscuous mode the MAC address filter is disabled and all packets of the currently joined 802.11 b network are captured.
- In non-promiscuous mode we see the traffic that this node is intended to receive only.

4.2 Task 4.2.1 Wireless ipserf server

Running the bandwidth benchmark on *Node 1* and *Node 2* using *global AP* as server in managed mode, resulted in both nodes getting almost the same throughput of $\sim$ 800KiB/s for the outgoing and incoming traffic. This was observed using *iptraf*. This implies a fair bandwidth sharing scheme.

4.3 Task 4.2.2 Different Channels

Node 1-3 performed the *iperf* benchmark using the *local AP* as AP and server and running in managed mode.

Hypothesis *We expected an equal distribution of the available bandwidth among all three nodes. Since only three nodes are competing for the bandwidth and not the whole class (12 nodes) as in subsection 4.2 the throughput for each node is expected to be higher.*

Observations The observed results are gathered in Table 1. Our hypothesis turned out to be correct. Changing the positions antennas by one meter did not have any observable influence. If *Node 1* was further away from the *local AP* than *Node 2*, both still got the same chance for using the medium because of waiting for the back-off time before sending introduces some randomness. The back-off time is much larger than the propagation delay resulting from the larger distance.

The total throughput was supposed to be 11MiB but we measured 7MiB. This is due to protocol overhead and obstacles, reflections and other environmental effects.

4.4 Task 4.2.3 Rate Changing

The transmission rate of *Node 1* was set to 1MiB, while the other two nodes used 11MiB and the same benchmark was executed.

	total	incoming	outgoing
Node 1	2500	1000	1500
Node 2	2500	900	1500
Node 3	2100	1000	1200
local AP	7000	4400	2300

Table 1: Results of measurement form Task 4.2.2. Units are KiB/s.

	total	incoming	outgoing
Node 1	900	300	600
Node 2	900	400	500
Node 3	900	400	500

Table 2: Results of measurement form Task 4.2.3. Units are KiB/s.

Hypothesis *We assumed* Node 1 *would be slower and* Node 2 *and* 3 *faster than before in subsection 4.3 because the* Node 1 *would free some bandwidth.*

Observation The observed results are gathered in Table 2. Our hypothesis turned out to be incorrect. All nodes were transmitting at the same slower speed if *Node 1* was sending. If *Node 1* was not sending *Node 2* and *3* were transmitting at full speed. This due to the fact that the *Node 1* needed more time for sending the same amount of data and thus allocated the medium for a longer period of time, slowing down the other two nodes as well.

When we reduced the transmission rate of all tree nodes, the result was the same as when the slower *Node 1* was sending in the previous setup.

4.5 Task 4.2.4 Selfish Back-off

Node 1 used a modified driver that set the back-off time to zero.

Hypothesis *Node 1 will get the whole bandwidth (up- and downstream) and the other nodes non.*

Observation The observed results are gathered in Table 3. Our hypothesis turned out to be partially correct. *Node 1* got a very high outgoing throughput, but almost no incoming. The other nodes got neither incoming nor outgoing throughput. Since *Node 1* allocated the medium without waiting for the back-off time, it was always the first to send not leaving any time

	total	incoming	outgoing
Node 1	7000	~ 0	7000
Node 2	~ 0	~ 0	~ 0
Node 3	~ 0	~ 0	~ 0

Table 3: Results of measurement form Task 4.2.4. Units are KiB/s.

slots for the other two nodes and the *local AP*. For that reason, *Node 1* did not get any data from the *local AP* and thus no incoming traffic. This scheme is not fair anymore and moving the antennas did not change anything.

Let's cheat all together When all the nodes used the modified driver without back-off time, the resulting throughput turned out to be unpredictable. Sometimes a node was sending at full speed and then not sending at all. The *local AP* did not send at all, since it used the back-off time.

Since none of the nodes was waiting before sending, a lot of collisions have happened, but we could not measure them.

Even when moving the antennas, we were not able to find any observable change, since it was still mostly random.

5 Analysis

References

[1] Wikipedia. The free encylopedia. Distributed coordination function. `https://en.wikipedia.org/wiki/Distributed_coordination_function`, 2015.

[2] Wikipedia. The free encylopedia. Ieee 802.11b-1999. `https://en.wikipedia.org/wiki/IEEE_802.11b-1999`, 2015.

[3] Linux Manual. iwconfig - linux man page. `http://linux.die.net/man/8/iwconfig`, 2015.

[4] The Wireshark team. Wireshark manual. `http://wiki.wireshark.org/CaptureSetup/wlan`, 2015.