ECE 6610: WIRELESS NETWORKS

EA1 Report

GROUP 1

<u>Team Members</u>:

UDAY KIRAN RAVURI

MRUNMAYI SHARAD PARANJAPE

PRAPURNA JAYAKRISHNA DUVVURI

RAKTIM PAL

Objectives:

1. Run the 802.11 MAC reference design on WARP and calculate throughput - from WARP and a laptop device
2. Run the modified 802.11 MAC design on WARP and calculate throughput - from WARP and the same laptop device
3. Compare the above obtained results

Devices Used:

1. WARP v3 board
2. 3 laptops (running 64-bit Windows 7 OS) having Ethernet Ports (one working as server, one for configuring WARP, and one for Wifi client):

A server (with a wired Ethernet connection to the access point) and listening on two ports, corresponding to two clients. This device is also used to configure the access point.

AP configuration parameters:

Channel used: 4

Mode: Wireless-g only

Putty terminal parameters:

COM port: COM3

Baud rate: 115200

The WARP board can be controlled dynamically and its output can be seen on this terminal (via the micro USB connected to the board). The JTAG is used to program the FPGA.

A client (having a wired connection to ETH A of the WARP board - which acts as a source or a sink for data packets) sending UDP packets to the server on one of the latter's listening ports. This device is also used to program the WARP's FPGA.

Another client which wirelessly transmits UDP data packets to the server (connected to the access point) on its (server's) other listening port. This client's wireless NIC implements 802.11-a/g standard design.

Router is Linksys router by Cisco. It is configured to work at 2.4GHz, with Channel 4.

Procedure:

1. Set up Network Topology for WIFI AP, WARP v3 and our laptop device (for wireless transmission)
2. Loaded 802.11 MAC Protocol in WARP v3 using Xilinx SDK 14.4
3. After setting up server and clients (WARP v3 and our device), using Jperf (Network performance and analysis tool - similar to iperf), throughput is measured at server side. This is measurement 1.
4. Now, after loading our code (programming the FPGA) into the WARP v3, using Jperf, throughput is again measured at server. This is measurement 2.
5. We then compare both measurements to show throughput improvement.

Our Design:

We have changed the unicast broadcast rate of the WARP device to 54 Mbps from its default value of 24 Mbps. The other changes made in order to modify the 802.11 design are as follows:

802.11 MAC protocol uses a binary exponential backoff mechanism. On every successful data transmission, the station increases its contention window by a factor of 2. The backoff timer is decremented from a chosen random value (in the range of 0 to CW) and when it gets to 0, the station transmits its packet on the channel.

Our laboratory setup is a situation where there is no hidden terminal problem. And also, there are only two stations (two clients - WARP device and laptop) which are contending for channel. In such a case, using a BE backoff is an overhead (the stations wait unnecessarily even when the channel is idle). So, we decided to decrease the backoff timer by changing the contention window limits, as shown in the following code snippet:

wlan_mac_sta.c | wlan_mac_low.c | wlan_mac_dcf.c

```c
 */
int wlan_mac_low_init(u32 type){
    u32 status;
    rx_frame_info* rx_mpdu;
    wlan_ipc_msg ipc_msg_to_high;

    mac_param_band = RC_24GHZ;
    mac_param_ctrl_tx_pow = 10;
    cpu_low_status = 0;

    //cw_exp_min = 4;
    cw_exp_min = 0;
    //cw_exp_max = 10;
    cw_exp_max = 0;

    //mac_param_rx_filter    = (RX_FILTER_FCS_ALL | RX_FILTER_HDR_ADDR_MATC
    mac_param_rx_filter    = (RX_FILTER_FCS_ALL | RX_FILTER_HDR_ALL);

    frame_rx_callback      = (function_ptr_t)nullCallback;
    frame_tx_callback      = (function_ptr_t)nullCallback;
    ipc_low_param_callback = (function_ptr_t)nullCallback;

    status = w3_node_init();

    if(status != 0) {
```

The above picture shows the change we made in wlan_mac_low.c code. [The wlan_mac_dcf handles the backoff timer functionalities. The arguments used in these functions obtain their values from the contention window exponents defined in the wlan_mac_low.c code]

As a result of this, the throughput has increased as shown in the measurements below.

Measurements:

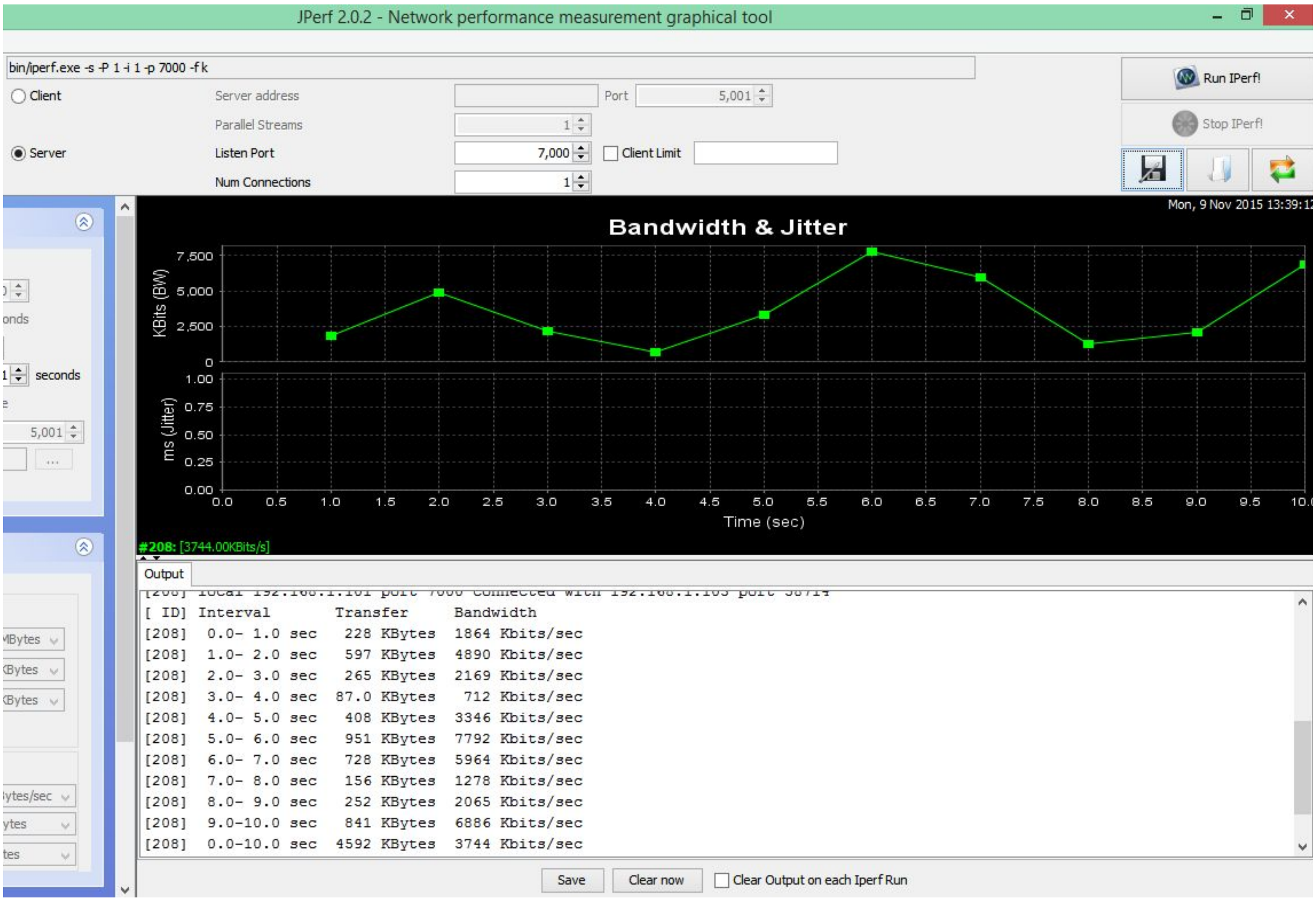


Figure 1: Throughput measurement of wireless transmission of UDP packets [during simultaneous transmission of data from wireless device (laptop) as well as from WARP v3 board] - using standard 802.11 design

Parameters used for transmission:

- UDP packets
- transmission rate = 54Mbps
- server listening on port 7000
- IP address of server (laptop to which AP is connected): 192.168.1.101

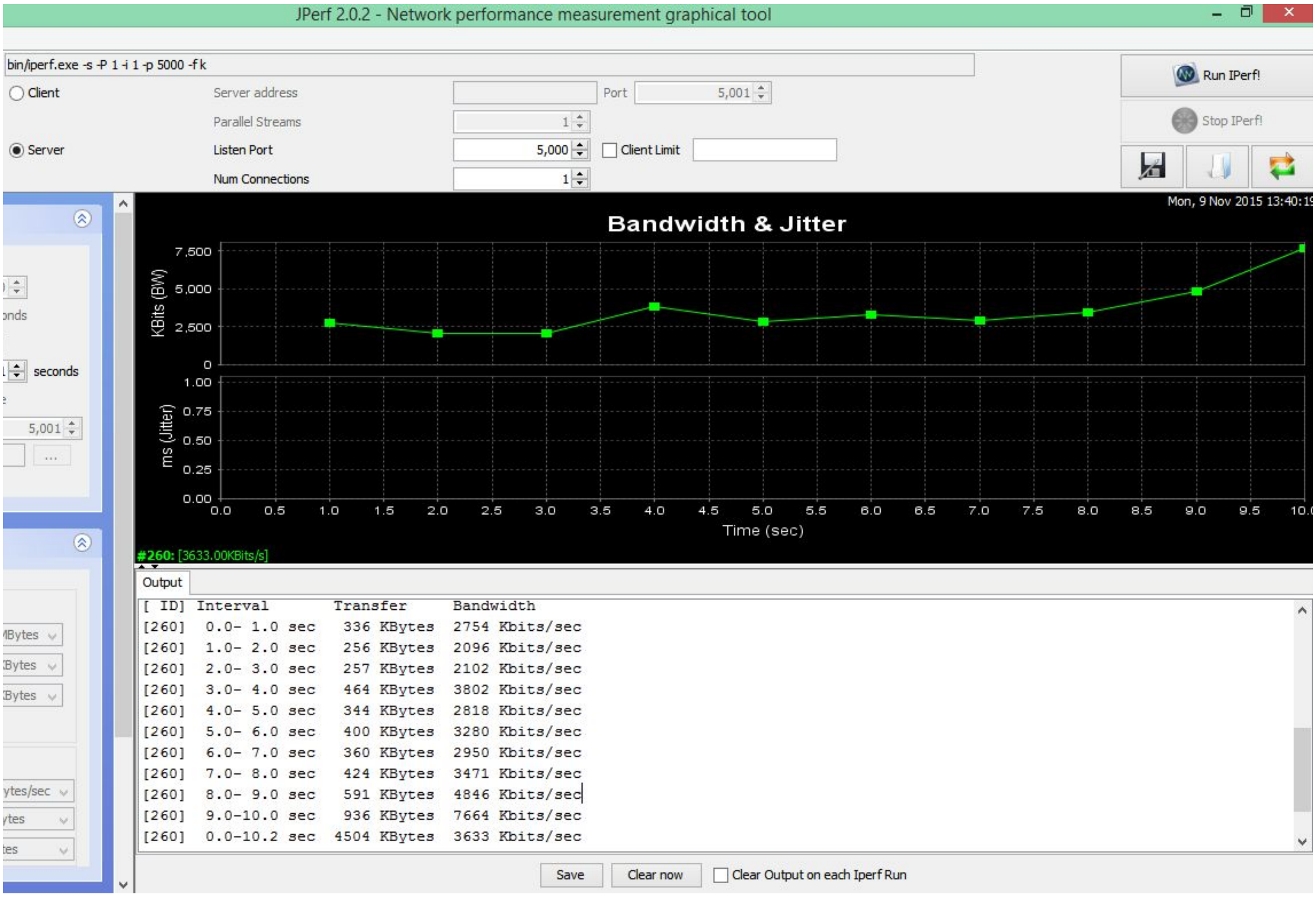


Figure 2: Throughput measurement of WARP transmission of UDP packets [during simultaneous transmission of data from wireless device (laptop) as well as from WARP v3 board] - using standard 802.11 design

Parameters used for transmission:

- UDP packets
- transmission rate = 54Mbps
- server listening on port 5000
- IP address of server (laptop to which AP is connected): 192.168.1.101

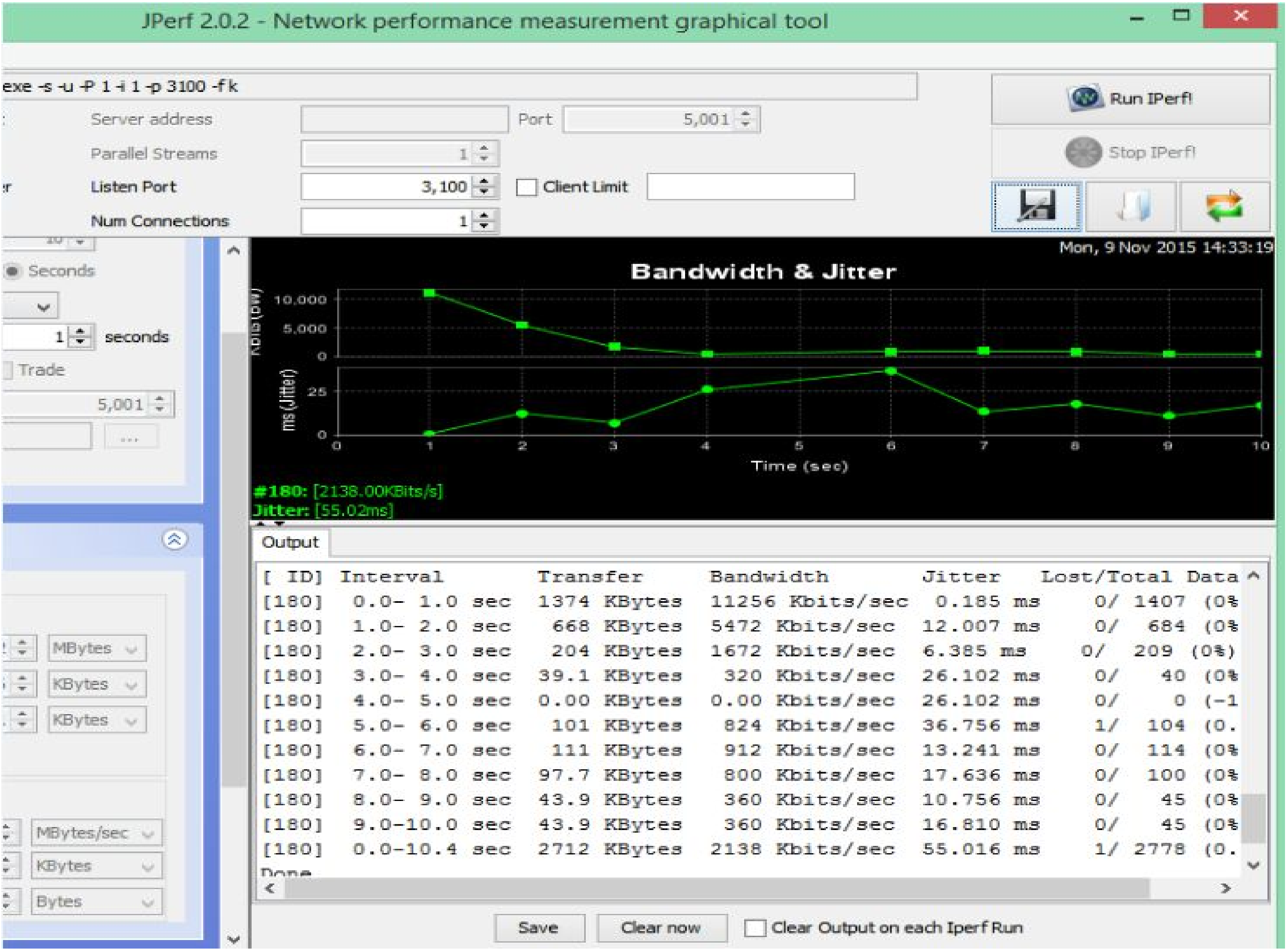


Figure 3: Throughput measurement of wireless transmission of UDP packets [during simultaneous transmission of data from wireless device (laptop) as well as from WARP v3 board] - using modified 802.11 design

Parameters used for transmission:

- UDP packets
- transmission rate = 54 Mbps
- server listening on port 3100
- IP address of server (laptop to which AP is connected): 192.168.1.101

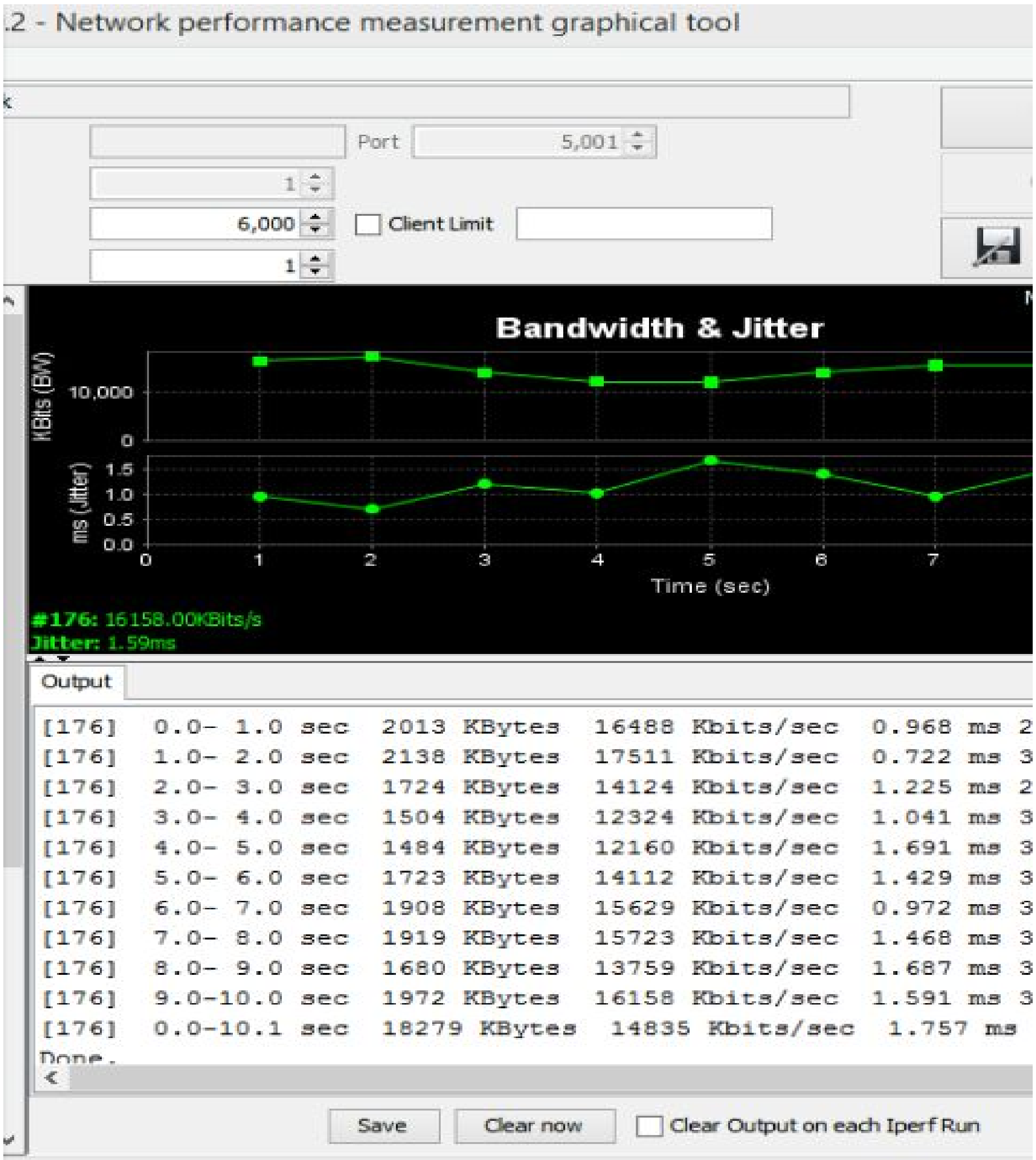


Figure 4: Throughput measurement of WARP transmission of UDP packets [during simultaneous transmission of data from wireless device (laptop) as well as from WARP v3 board] - using modified 802.11 design

Parameters used for transmission:

- UDP packets
- transmission rate = 54 Mbps
- server listening on port 6000
- IP address of server (laptop to which AP is connected): 192.168.1.101

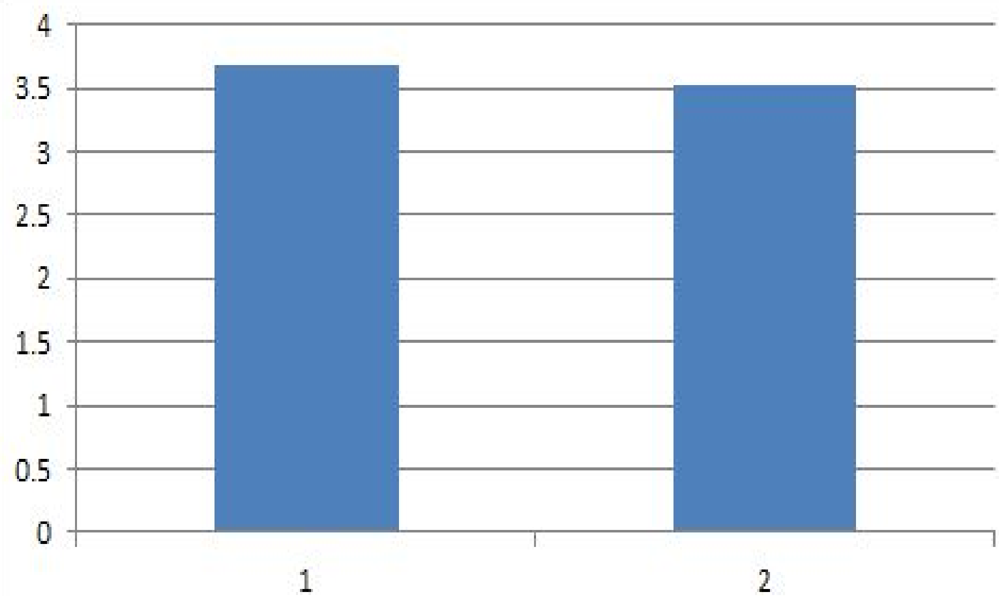
Results:

From the observations above, we calculate the throughputs obtained:

1. Using the standard 802.11 design:

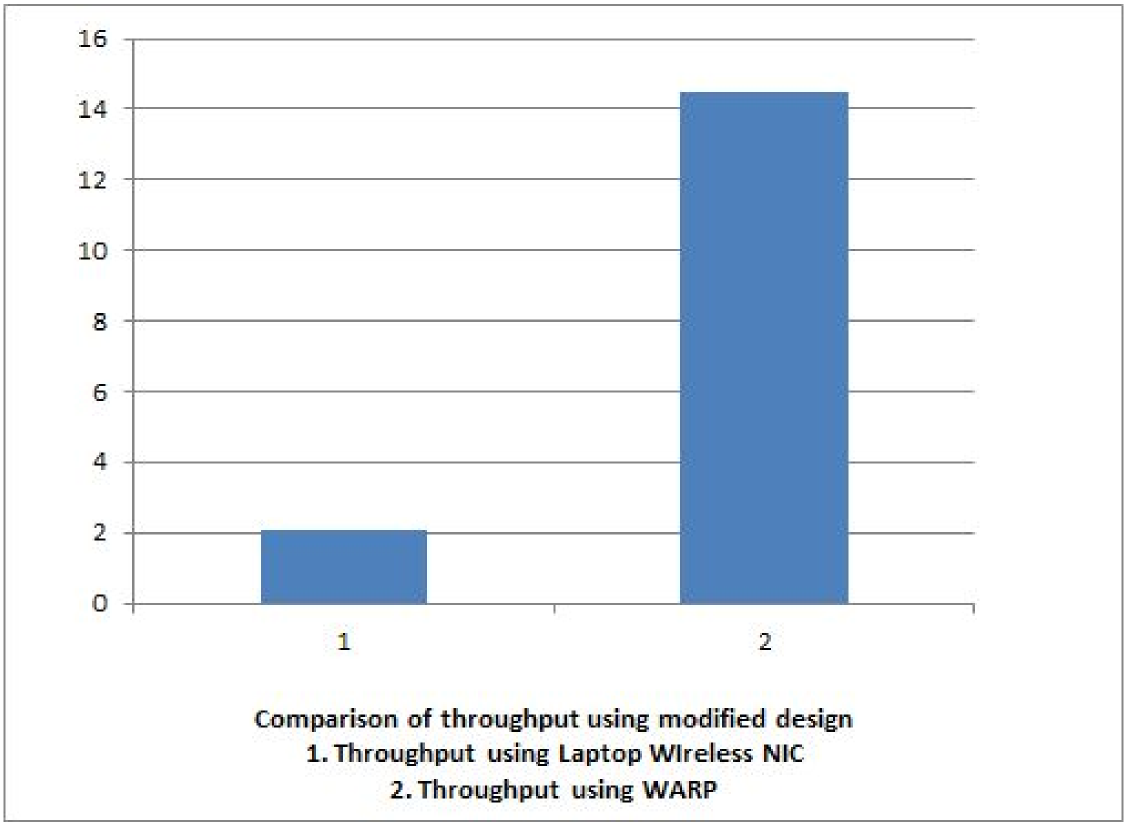
Wireless transmission throughput = 4.592 MB/10 sec = 3.6736 Mbps

WARP transmission throughput = 4.504 MB/10.2 sec = 3.5325 Mbps



Comparison of Throughput using standard design
1. Throughput using laptop wireless NIC
2. Throughput using WARP

2. Using the modified design:



Wireless transmission throughput = 2.086 Mbps

WARP transmission throughput = 14.478 Mbps

Throughput improvement (WARP over wireless) = 6.94 (approx. 7) times.

Conclusion:

In our design, we increased throughput by decreasing backoff timer (the contention window).

If we decrease backoff timer, by reducing contention window to 1, packets will be transmitted without backoff, after DIFS time after each successful transmission. It would act similar to 1-persistent CSMA.

This will give increased throughput as seen in results.