WirelessLab WS 2016/17

Assignment 3: Tools of the Trade - Data processing and Performance Evaluation

Group - 6
Gasper Kojek
Jens Klein

Question 1: Setup

(e) Provide the network and wireless configuration files of one node. How do you check that the nodes can exchange traffic with each other?

Node 6 (the one with two 802.11n cards)

Wireless config:

```
config wifi-device 'radio0'
        option type 'mac80211'
        option channel '11'
        option hwmode '11g'
        option path 'pci0000:00/0000:00:0c.0'
        option disabled '0'

config wifi-iface 'default_radio0'
        option device 'radio0'
        option mode 'ap'
        option encryption 'none'
        option ssid 'group06_ap'
        option network 'wlan0'
        option ifname 'wlan0'

config wifi-device 'radio1'
        option type 'mac80211'
        option channel '11'
        option hwmode '11g'
        option path 'pci0000:00/0000:00:0e.0'
        option disabled '0'
        option htmode 'HT20'

config wifi-iface 'default_radio1'
        option device 'radio1'
        option network 'lan'
        option mode 'monitor'
        option hidden '1'
```

Network config:

```
config interface 'loopback'
        option ifname 'lo'
        option proto 'static'
        option ipaddr '127.0.0.1'
        option netmask '255.0.0.0'

config interface 'lan'
        option type 'bridge'
        option ifname 'eth0'
        option proto 'dhcp'

config interface 'wlan0'
        option ipaddr '172.17.5.11'
        option netmask '255.255.255.0'
        option proto 'static'
        option dns '172.17.255.254'
```

Check inter-connectivity

(ST = Stepping Stone, N6 = Node06, N15 = Node15)

- get IPs for the nodes once more N6: `ifconfig wlan0 | grep 'inet addr' | awk '{print $2}'`
- Output: addr:172.17.5.10
- N15: `ifconfig wlan0 | grep 'inet addr' | awk '{print $2}'`
- Output: addr:172.17.5.11
- Check for reachability
- N6: `ping 172.17.5.11`
- Output (trunc): 64 bytes from 172.17.5.10: seq=1 ttl=64 time=0.188 ms
- N15: `ping 172.17.5.10`
- Output (trunc): 64 bytes from 172.17.5.10: seq=1 ttl=64 time=1.136 ms

The Nodes are connected.

Question 2: (30 Points) Passive Measurements and Difference of Management frames

a)

- Selected channels: 1, 6, 11 (should be most used ones)
- Wireless setup:
    - Channel setup:
    - `iw wlan1 set channel <cnumber>`
    - `iw wlan1 info` to check wlan1 monitoring interface status
- Logging command: We decided to stop logging after 5min (ST = Stepping Stone, N6 = Node 6)
    - Channel 1
    - ST: `nc -l -p 8080 > channel1.cap`
    - N6: `iw wlan1 set channel 1`
    - N6: `iw wlan1 info | grep channel`
    - Output: channel 1 (2412 MHz), width: 20 MHz (no HT), center1: 2412 MHz
    - N6: `tcpdump -i wlan1 -G 300 -w- | nc 172.17.3.1 8080`
    - Channel 6
    - ST: `nc -l -p 8080 > channel6.cap`
    - N6: `iw wlan1 set channel 6`
    - N6: `iw wlan1 info | grep channel`
    - Output: channel 6 (2437 MHz), width: 20 MHz (no HT), center1: 2437 MHz
    - N6: `tcpdump -i wlan1 -G 300 -w- | nc 172.17.3.1 8080`
    - Channel 11
    - ST: `nc -l -p 8080 > channel11.cap`
    - N6: `iw wlan1 set channel 11`
    - N6: `iw wlan1 info | grep channel`
    - Output: channel 11 (2462 MHz), width: 20 MHz (no HT), center1: 2462 MHz
    - N6: `tcpdump -i wlan1 -G 300 -w- | nc 172.17.3.1 8080`

tcpdump flags explained:

- `-i wlan1` sets it to listen on interface 1 (Monitor)
- `-G 300` sets it to listen for 300s or 5min
- *all traces are saved at ~/hw04/q2a/ on ST*

Networks

- Beacons on CH1:
- `tcpdump -r channel1.cap | grep Beacon | awk '{print $14}' | sed "s/[()]//g" | sort -u`
- Output:
- `Group01`
  `LEDE`
  `TUB-Guest`
  `TUB-intern`
  `WirelessLab-Group04`
  `eduroam`
  `group06_ap`

- Beacons on CH6:
- `tcpdump -r channel6.cap | grep Beacon | awk '{print $14}' | sed "s/[()]//g" | sort -u`
- Output:
- `Group01`
  `LEDE`
  `TUB-Guest`
  `TUB-intern`
  `WirelessLab-Group04`
  `eduroam`
  `group06_ap`

- Beacons on CH11:
- `tcpdump -r channel11.cap | grep Beacon | awk '{print $14}' | sed "s/[()]//g" | sort -u`
- Output:

- `FGINET`
  `Group01`
  `Group03`
  `LEDE`
  `MMS-TECHNIK`
  `TUB-Guest`
  `TUB-intern`
  `WirelessLab-Group04`
  `eduroam`
  `foobar`
  `group002`
  `group06_ap`

- Alternatively getting networks:
- N15: `iw wlan0 scan`
- Output not included because to big

b)

Beacon Frame (Management Frame, Subtype 8)

Beacon frames are emitted by Access Points (or STA in IBSS) to advertise the APs service with its characteristics. This mechanism is not only useful for already connected STAs but also for associated STA (Periodic Frames that hold information like RSS, etc). Discovering networks listening to beacons is called passive scanning

```
▶ IEEE 802.11 Beacon frame, Flags: ........C
▼ IEEE 802.11 wireless LAN management frame
  ▶ Fixed parameters (12 bytes)
  ▼ Tagged parameters (263 bytes)
    ▶ Tag: SSID parameter set: eduroam
    ▶ Tag: Supported Rates 18, 24(B), 36, 48, 54, [Mbit/sec]
    ▶ Tag: DS Parameter set: Current Channel: 1
    ▶ Tag: Traffic Indication Map (TIM): DTIM 0 of 0 bitmap
    ▶ Tag: Country Information: Country Code DE, Environment Any
    ▶ Tag: QBSS Load Element 802.11e CCA Version
    ▶ Tag: ERP Information
    ▶ Tag: HT Capabilities (802.11n D1.10)
    ▶ Tag: RSN Information
    ▶ Tag: HT Information (802.11n D1.10)
    ▶ Tag: Extended Capabilities (8 octets)
    ▶ Tag: Cisco CCX1 CKIP + Device Name
    ▶ Tag: Vendor Specific: Aironet: Aironet DTPC Powerlevel 0x0B
    ▶ Tag: Vendor Specific: Microsof: WMM/WME: Parameter Element
    ▶ Tag: Vendor Specific: Aironet: Aironet Unknown (1) (1)
    ▶ Tag: Vendor Specific: Aironet: Aironet CCX version = 5
    ▶ Tag: Vendor Specific: Aironet: Aironet Unknown (11) (11)
    ▶ Tag: Vendor Specific: Aironet: Aironet Unknown (19)
    ▶ Tag: Vendor Specific: Aironet: Aironet Client MFP Enabled
    ▶ Tag: Vendor Specific: Aironet: Aironet Unknown (12)
```

Example Beacon Frame from the trace of CH1

Probe Request (Management Frame, Subtype 4)

A Probe request is sent by a STA to ask for networks. This happens on every channel. An AP that is available for associating will reply with a probe response. Discovering networks by emitting probe request and awaiting response is called active scanning

```
▶ IEEE 802.11 Probe Request, Flags: ........C
▼ IEEE 802.11 wireless LAN management frame
  ▼ Tagged parameters (31 bytes)
    ▶ Tag: SSID parameter set: Broadcast
    ▶ Tag: Supported Rates 1, 2, 5.5, 11, 6, 9, 12, 18, [Mbit/sec]
    ▶ Tag: Extended Supported Rates 24, 36, 48, 54, [Mbit/sec]
    ▶ Tag: DS Parameter set: Current Channel: 1
    ▶ Tag: Extended Capabilities (8 octets)
```

Example Probe Request Frame from the trace of CH1

Differences of the two above frames

- SSID parameter set
- Usually this parameter is set to "0" (Broadcast) in Probe Request to address every APs whereas in Beacons the sending AP's SSID is supplied.
- The Beacon has additional fields like HT Capabilities, Country Information or Vendor Specific tags. Those Parameters would be sent as Probe Response of the Probe Request.

c)

The extended support rate tag is, as the name implies an extensions to the support rates tag. Per definition one flag has 8 octes of data it can hold. One octet for every supported rate. But since IEEE 802.11g there are more than 8 supported rates. That is why simply another tag was added to hold all possible values. When all rates fit into the Supported Rates field it must not actively add the extended support rates field.

Question 3:

(ST = Stepping Stone, N6 = Node 6, N15 Node 15)

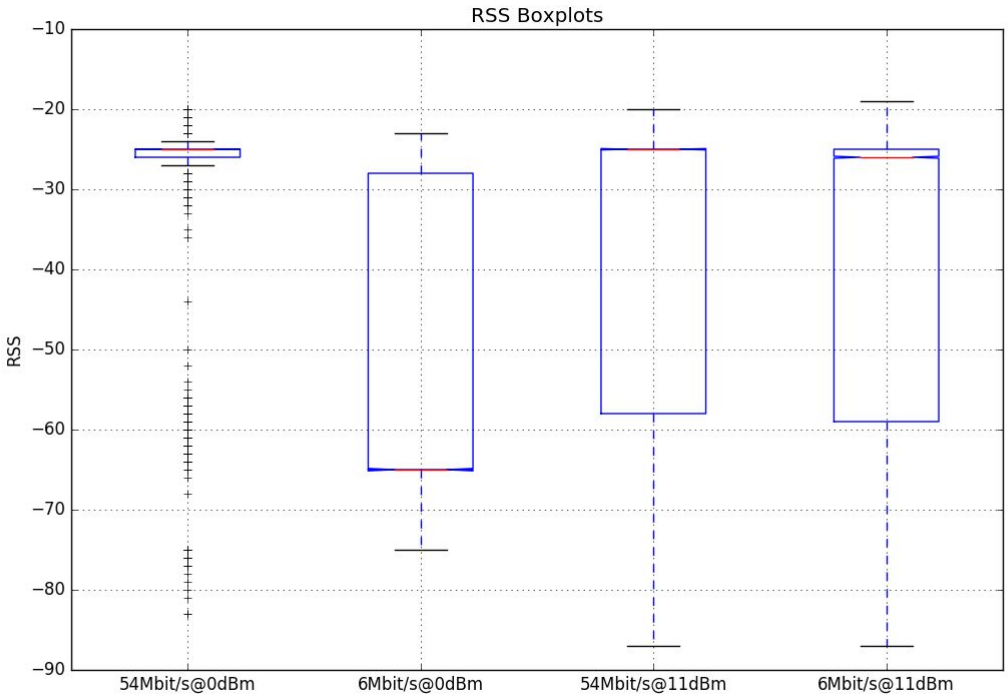
a)

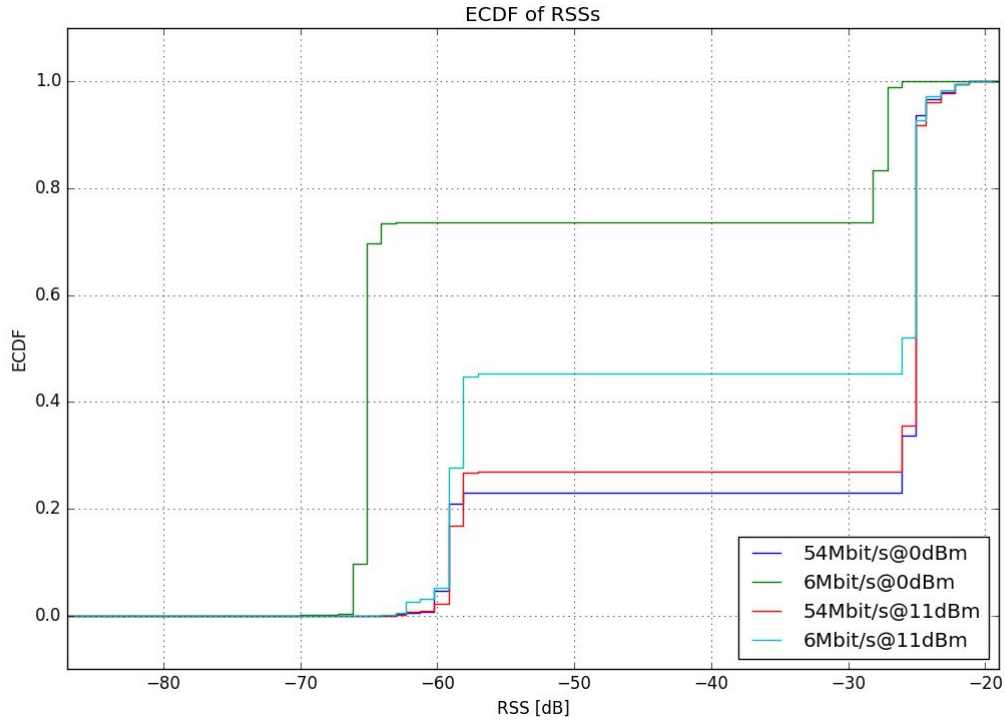
1. Check that all interfaces are on the same channel
2. N6 `iw wlan0 info | grep channel` Output: channel 11 [..]
3. N6 `iw wlan1 info | grep channel` Output: channel 11 [..]
4. N15 `iw wlan0 info | grep channel` Output: channel 11 [..]
5. Fine. All are on the same CH
6. check settings on N6
7. `iw wlan0 info | grep txpower`
8. Output: txpower 30.00 dBm
9. TX Power on Max
10. setup transmission power (tp) and transmission rate (tr) on N15
11. `iw wlan0 set txpower fixed 0`
12. `iw wlan0 info | grep txpower`
13. Output: txpower 0.00 dBm
14. `iw wlan0 set bitrates legacy-2.4 6`
15. `iw wlan0 station dump | grep 'tx bitrate'`
16. Output: tx bitrate: 6.0 MBit/s
17. iperf server command (N6):
18. `iperf -s -u`
    - -s = server mode
    - -u = udp packets
19. start nc (ST)
20. `nc -l -p 8080 > channel11-<tp>-<tr>.cap`
21. tcpdump on N6
22. `tcpdump -i wlan1 -w- | nc 172.17.3.1 8080`
23. iperf client command (N15):
24.
```
for i in `seq 12`; do
    iperf -c 172.17.5.10 -u -b 7M -t 30 -l 1024
    sleep 2s
done
```

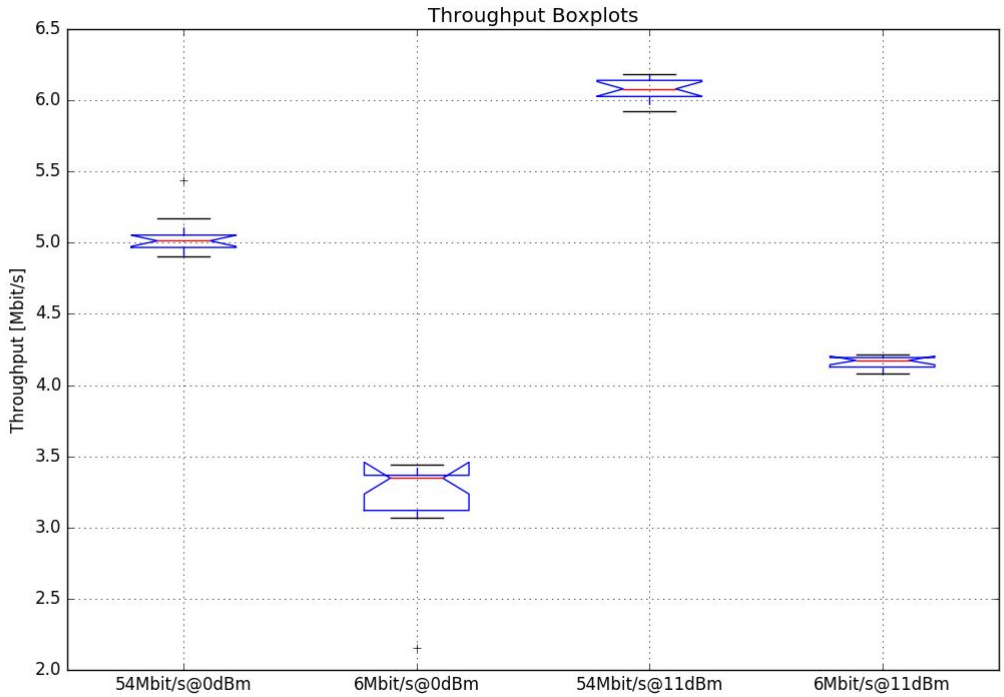
    - -c = client mode (server ip)
    - -u = upd packets
    - -b = bandwidth #
    - -t = test interval #

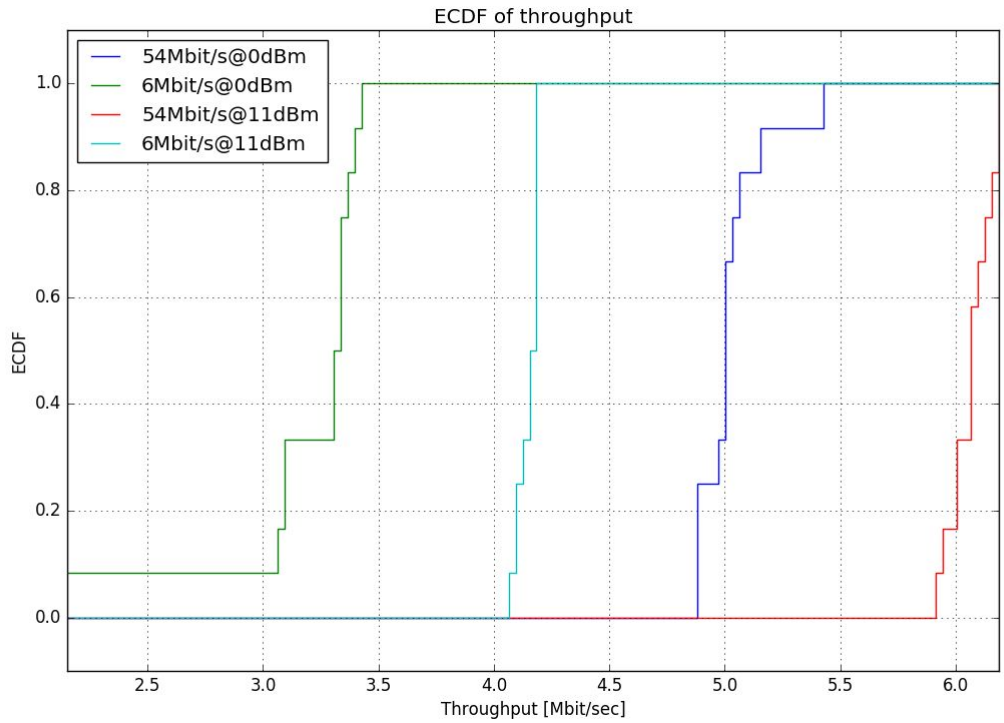
    - -l = upd packet size #
    - sleep 2s, wait for connection to be closed
25. Copy iperf Server output to `iperf-<tp>-<tr>.out`
26. Repeat Step 3, 5, 6, 7, 8 for remaining values
27. TP: 0 dBm, TR: 54mbit/s
28. NP15 iperf cmd: `iperf -c 172.17.5.10 -u -b 56M -t 30 -l 1024`
29. NP15 tx pwr: `iw wlan0 info | grep txpower` Output: txpower 0.00 dBm
30. NP15 tr : `iw wlan0 link | grep rate` Output: tx bitrate: 54.0 MBit/s
31. TP: 11 dBm, TR: 54mbit/s
32. NP15 iperf cmd: `iperf -c 172.17.5.10 -u -b 56M -t 30 -l 1024`
33. NP15 tx pwr: `iw wlan0 info | grep txpower` Output: txpower 11.00 dBm
34. NP15 tr : `iw wlan0 link | grep rate` Output: tx bitrate: 54.0 MBit/s
35. TP: 11dBm, TR: 6mbit/s
36. NP15 iperf cmd: `iperf -c 172.17.5.10 -u -b 7M -t 30 -l 1024`
37. NP15 tx pwr: `iw wlan0 info | grep txpower` Output: txpower 11.00 dBm
38. NP15 tr : `iw wlan0 link | grep rate` Output: tx bitrate: 6.0 MBit/s
39. TP: 0 dBm, TR: 6mbit/s
40. NP15 iperf cmd: `iperf -c 172.17.5.10 -u -b 7M -t 30 -l 1024`
41. NP15 tx pwr: `iw wlan0 info | grep txpower` Output: txpower 0.00 dBm
42. NP15 tr : `iw wlan0 link | grep rate` Output: tx bitrate: 6.0 MBit/s

b)


RSS Boxplots
RSS
−10
−20
−30
−40
−50
−60
−70
−80
−90
54Mbit/s@0dBm
6Mbit/s@0dBm
54Mbit/s@11dBm
6Mbit/s@11dBm


ECDF of RSSs
ECDF
1.0
0.8
0.6
0.4
0.2
0.0
−80
−70
−60
−50
−40
−30
−20
RSS [dB]
54Mbit/s@0dBm
6Mbit/s@0dBm
54Mbit/s@11dBm
6Mbit/s@11dBm


Throughput Boxplots
Throughput [Mbit/s]
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
54Mbit/s@0dBm
6Mbit/s@0dBm
54Mbit/s@11dBm
6Mbit/s@11dBm


ECDF of throughput
54Mbit/s@0dBm
6Mbit/s@0dBm
54Mbit/s@11dBm
6Mbit/s@11dBm
ECDF
1.0
0.8
0.6
0.4
0.2
0.0
2.5
3.0
3.5
4.0
4.5
5.0
5.5
6.0
Throughput [Mbit/sec]

- Medians:

```
54Mbit/s@0dBm: median = 5.013333Mbit/s
6Mbit/s@0dBm: median = 3.360000Mbit/s
54Mbit/s@11dBm: median = 6.093333Mbit/s
6Mbit/s@11dBm: median = 4.186667Mbit/s
```

```
54Mbit/s@0dBm: median = -25dBm
6Mbit/s@0dBm: median = -65dBm
54Mbit/s@11dBm: median = -25dBm
6Mbit/s@11dBm: median = -26dBm
```

- From RSS plots we can conclude that there was some kind of interference when capturing scenario 2 (6Mbit/s, 0dBm), which produced so much readings at -65dBm to push the median to -65dBm, as opposed to other scenario medians, which are around -25dBm. We see no other reason for this scenario to be that much worse than the others. Although it is true that with wour settings (7Mbit/s on 6Mbit/s medium and 56Mbit/s on 54Mbit/s medium) this is expected to be the worst run, as it's at the lower power and 17% oversaturated we don't think this would give such a drastic difference when looking at RSS value. What is also interesting is that 54Mbit/s at 0dBm power is the best performing run, instead of 54Mbit/s at 11dBm. On ewould expect the latter one to perform the best, as it's saturated only by 4% and has the highest transmit power, although saturation shouldn't have much impact on received strength.
- When we look at throughput everything is following expectations, as all the runs are lower than the max throughput medium supports because of medium saturation. We also expected to see a better throughput with higher transmit power. We were quite surprised by how much the throughput has fallen of maximum for 54Mbit/s medium, but we do acknowledge that we used Channel 11, which is probably most used in this environment. When we look at boxplot of scenario 6Mbit/s at 0dBm we again see something interesting, as the 75% quartile is lower than higher confidence interval of median, which tells us that the median is quite unconfident with our dataset of this scenario.