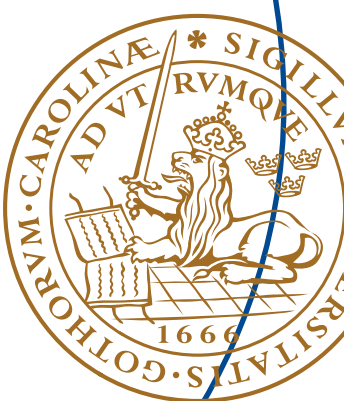
Master's Thesis

Cooperative Wireless Network Configuration Protocol Design

Simon Davani
Malda Said



Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, December 2013.

Master's Thesis

Cooperative Wireless Network Configuration Protocol Design

By

Simon Davani and Malda Said

Department of Electrical and Information Technology
Faculty of Engineering, LTH, Lund University
SE-221 00 Lund, Sweden

Abstract

With the popularity of the IEEE 802.11 standard ever increasing, scanning and finding an access point is not an issue in medium to large cities as access points are deployed densely. Usually when searching for local access points in an area, it results in a number of available access points. This dense deployment of access points causes problems with just three non-overlapping frequency channels in the unlicensed 2.4 GHz band, especially in uncoordinated deployments. For example, residential users usually place their access points (routers) on locations according to their own contentment, and configure them without taking their neighbouring access points' configuration into consideration. As a consequence the signal coverage areas of the different access points overlap which results in interference if they are running on the same channel. Therefore, we present an inter-access point protocol, working with the help of newly constructed text-based frames, which enables cooperation between access points and handles different functions to mitigate interference and improve the network performance. The protocol functions that have been considered and included in the protocol are; channel assignment, virtual TDMA between access points, balancing the load between access points and switching off interfering access points. Simulations were performed in ns-3, and it was concluded that by introducing the protocol functions the overall network throughput is enhanced.

Acknowledgments

We would like to show our gratitude to everyone who helped and supported us with our thesis, especially our examiner Ulf Körner and supervisor Björn Landfeldt. We also want to thank our families and friends who have been supportive and patient when we were busy writing this thesis. Last but not least, we would like to thank each other for the amazing cooperation and achieved results!

Table of Contents

List of Figures

List of Tables

CHAPTER 1

1 Introduction

Today, the IEEE 802.11 standard is the most common standard for Wireless Local Area Networks (WLANs) due to its low cost and easy deployment. Scanning and finding an IEEE 802.11 Access Point[1] (AP) in medium to large cities usually local results in a number of available APs. APs can be deployed with and without coordination. They can be found in coordinated deployments such as universities, airports, large enterprises and uncoordinated deployments such as in residential buildings (private APs) etc.[1] The majority of users usually demands better performance in any kind of technology they are using, everything from cameras to laptops, and this is also the case in WLANs. With the growing number of WLAN users there is a higher user-demand. Since the IEEE 802.11 standard works on the unlicensed frequencies [2, pp. 207-239], which offers limited bandwidth, this increased usage becomes a problem. Moreover, as mentioned in [2, pp. 232-233] the number of non-overlapping channels in the unlicensed frequency bands is also limited. The combination of many users and limited resources results in unsatisfying performance. One major factor for this unsatisfying performance is interference which can occur when there are many nearby WLAN APs using the same frequency channel at the same time. This situation usually appears in areas where the APs have been deployed without coordination, typically happening in e.g. private residents, where users buy their own private APs (routers) and configure them without considering neighbouring routers' configuration. [3][11] This is why a new protocol is introduced in this thesis that involves

[1] Area where you can find connection to an IEEE 802.11 WLAN

the exchange of information between WLAN APs and cooperative behaviour. The goal of this cooperation is to achieve better network performance for all. Imagine self-organizing APs communicating and agreeing with each other to find the best channel selection, to schedule the medium access of neighbouring APs using the same channel, to balance the load between APs and to switch off interfering APs.

As mentioned in [4], the 802.11f standard was proposed. It defined inter-access point communication but with a different purpose. The purpose was that the APs would exchange information about roaming stations through a common backbone network. It did not, however, define any functions regarding channel assignment, load balancing etc. Many algorithms have been proposed for handling channel selection, interference mitigation and load balancing, but they have all been studied and defined separately. To our knowledge, there is no other protocol that fully offers wireless inter-access point communication for handling all four functions mentioned above. The thesis will mainly focus on how the APs communicate and exchange information in order to make it possible for the functions to work, and not on the algorithms behind the functions. It will only focus on one of the IEEE 802.11 wireless network modes, namely the infrastructure mode. The infrastructure mode is the most common mode of operation in 802.11 and that is why it seems reasonable to focus on it. With the help of simulations in the network simulator ns-3 [6], we show how the network performance changes when using the different functions of the protocol. The simulation results will be presented using plots and tables, followed by a discussion. Using topology figures helps the reader to get an intuitive illustration of the protocol functions. Flow charts are used to describe the process upon receiving a frame. This gives an intuitive demonstration of the decoding of a packet, and the actions followed.

1.1 Reader's guideline

The remaining part of the thesis is structured as follows; Chapter 2 gives a theoretical background to help the reader to understand the basic principles of the IEEE 802.11 standard. In the following chapter, Chapter 3, the protocol is presented and explained. Chapter 4 demonstrates the simulations of the protocol functions and the results are discussed. The conclusion is given in Chapter 5, followed by future work proposals in Chapter 6.

CHAPTER 2

2 Theoretical Background

2.1 IEEE 802.11 Infrastructure and Ad-hoc Networks

There are two different types of IEEE 802.11 architectures; infrastructure and ad-hoc. In infrastructure mode, there is one access point (AP) and several stations (laptops, smartphones, tablets etc.) affiliated to it. All the communication is done through the AP. The AP and its stations together create a Basic Service Set (BSS) and it is identified by its Service Set Identification (SSID). Via a so-called distribution system (DS), which is a type of backbone network, several BSSs can be connected together to extend the wireless coverage area. The DS and BSSs together create an Extended Service Set (ESS) and each ESS has its own Extended Service Set Identification (ESSID) to separate the different ESS networks (see Figure 1).

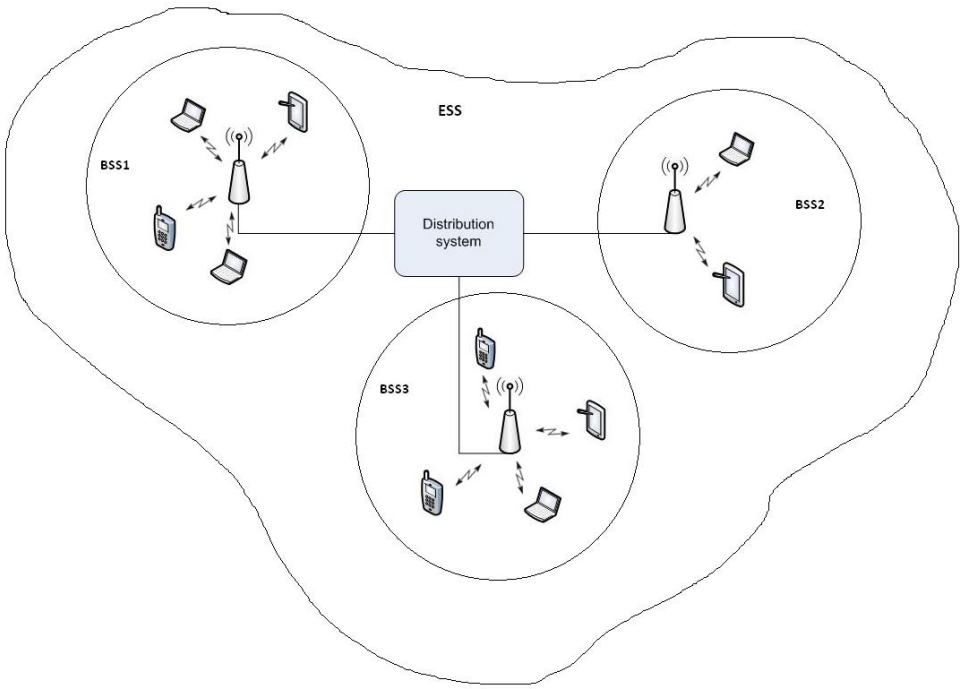


Figure 1. Architecture of an infrastructure-based IEEE 802.11 network

The ad-hoc mode works in a different way. Stations communicate directly with each other in ad-hoc mode, assuming that they are within transmission range and use the same radio frequency channel, without the need of an AP. Together the stations create an Independent Basic Service Set (IBSS). Figure 2 demonstrates this type of network.

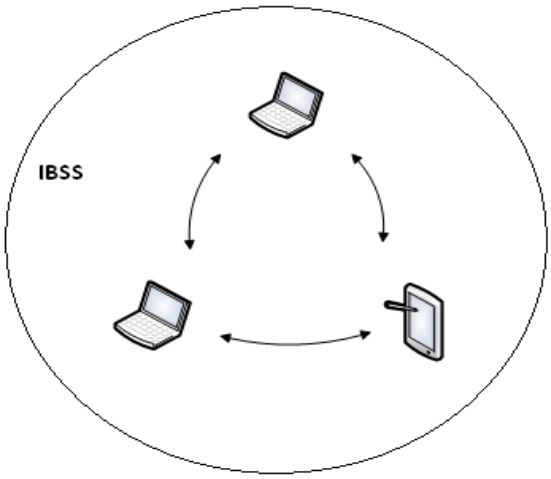


Figure 2. Architecture of ad-hoc based IEEE 802.11 network

2.2 The Beacon Frame in IEEE 802.11

The IEEE 802.11 standard uses management, control and data frames for administrative functions, control functions and data transmission respectively. In infrastructure mode, a management frame is periodically broadcasted from the AP to its associated stations. This management frame is called beacon frame in IEEE 802.11 and is necessary to establish connection between AP and stations and also to maintain the connection. [2, pp. 205-207, 226][10, pp. 404-437] [11] Figure 3 shows how a beacon frame is constructed.

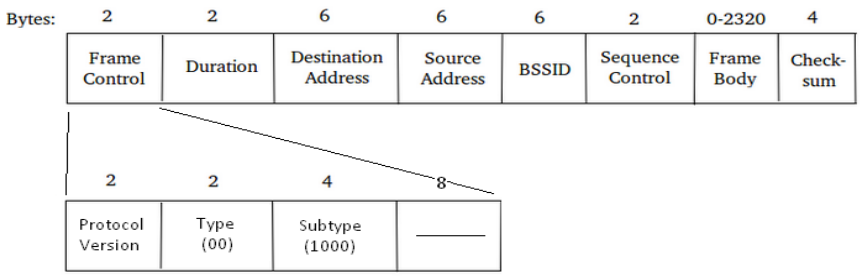


Figure 3. Structure of the IEEE 802.11 beacon frame

The Frame Control field contains information concerning the frame type. Within the Frame Control field, there are for instance a Type field and Subtype field. The Type field indicates if it is a management, control or data frame. The Subtype field specifies what type of management, control or data frame it is. For example, for a beacon frame the Type field is set to 0 and the Subtype field is set to 8. The Duration field carries information that is used for medium access. The Destination Address is the address of the receivers. This field is set to all ones in the beacon frame, which means that it is broadcasted to all stations. The Source Address is the address of the transmitter. In the beacon frame, it is the address of the transmitting AP. The BSSID field is the address of the current BSSID. The Sequence Control field is used by the receiver to prevent it from getting duplicated frames. The Frame Body field includes information that is varying depending on what type of management, control or data frame it is. The Checksum field provides error detection capability. [7, pp. 20-22][10][12]

Stations choose an AP to connect to based on the received signal strength of the beacon frame. [5] If a station is already connected to an AP, but suddenly receives a stronger signal from another AP, it will terminate its connection to the first AP and try to associate with the new AP. This function is called roaming. [2, p. 230]

In ad-hoc mode, the beacon frame is broadcasted from the stations. Infrastructure-based networks are usually less complex than ad-hoc-based networks. The complexity of every station in ad-hoc networks is higher since they all have to implement mechanisms to handle the medium access, whereas in infrastructure-based networks this functionality is controlled solely by the AP. This results in simpler stations and collision free transmissions within a BSS, which ensures a better overall network performance. [2, pp. 226-227]

2.3 Frequency of operation

The IEEE 802.11 standards work on the unlicensed 2.4 GHz and 5 GHz frequency bands, depending on the version. In the US/Canada region, there are 11 channels on the 2.4 GHz band, and in Europe there are 13 channels available on the same frequency band. Each channel occupies a bandwidth of 22 MHz and the spacing between the center frequencies is 5 MHz. There are three non-overlapping channels, namely channel 1, 6 and 11 in the US/Canada region, and 1, 7 and 13 in Europe. Figure 4 depicts the three non-overlapping channels and their center frequencies in Europe.

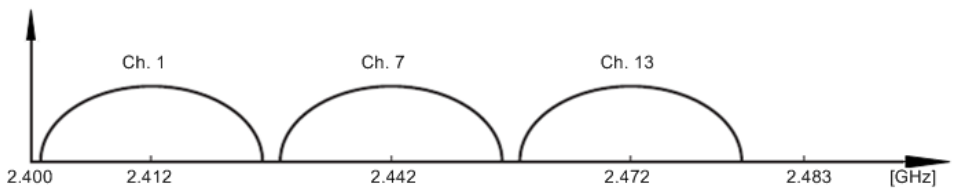


Figure 4. The non-overlapping channels in the unlicensed 2.4 GHz band

These non-overlapping channels are typically used by APs to decrease interference. The 5 GHz band offers more channels and the number of channels available depends on the country since countries have their own

regulations. There are also non-overlapping channels in the 5 GHz band and for the same reason as in the 2.4 GHz band; they are assigned to APs to decrease the interference. [2, pp. 232-233][8]

Because of the limited number of available channels, they need to be reused. Two APs that are positioned far enough away from each other should use the same channel if the interference level detected by the AP is lower than a certain threshold. In practice, interference means that APs and stations need to transmit frames more than once to increase the probability of successful transmission. For example, if data is transmitted at 10 Mbps and one frame is sent successfully, the effective throughput is 10 Mbps. However, if interference is introduced and the efficiency drops to 50 %, the effective throughput will be 5 Mbps since the frame needs to be sent twice. [9]

2.4 Medium Access

The IEEE 802.11 standard only defines the physical (PHY) layer and the medium access control (MAC) layer. The obligatory medium access mechanism used in IEEE 802.11 is based on carrier sense multiple access with collision avoidance (CSMA/CA). CSMA/CA is a scheme where stations sense the carrier and randomly access the medium whenever it is idle. To avoid collisions from occuring, a random backoff time is introduced. If the medium is free for a specific time period (DIFS, Distributed Inter Frame Space), any station can use it directly for transmission. If, however, the medium is sensed busy, stations need to wait for this DIFS plus the random backoff time that postpones the medium access. The random backoff time will prevent stations from transmitting at the same time, therefore avoiding collisions. After getting a random backoff time, a station will still sense the medium, and if it is busy, it lost its opportunity to access the medium in that cycle and needs to remain on hold until the channel is free again for at least DIFS. When the random backoff

time is over and the medium is free, the station can use the channel straight away. This mechanism only helps to prevent collisions, but does not provide any fairness among the competing stations. To solve this, a station stops its backoff time if it does not get access to the medium in the first cycle. In the next cycle (after waiting period DIFS), instead of getting a new backoff time, the station starts its old backoff counter again, and once it reaches zero it can access the channel. This will be beneficial for stations that have waited longer for accessing the medium. [2, pp. 215-216] [11]

2.5 The IEEE 802.11 standards

The IEEE 802.11 standard comes in different versions, and what differentiates them is their PHY layer. The most common and successful version is the IEEE 802.11b, operating at the unlicensed 2.4 GHz band. It uses Direct Sequence Spread Spectrum (DSSS) for transmission and offers a maximum data rate of 11 Mbps. IEEE 802.11g also operates on the 2.4 GHz band, but offers a maximum data rate of 54 Mbps by using Orthogonal Frequency-division Multiplexing (OFDM) for transmission. 802.11g is fully backward compatible with 802.11b. 802.11a also uses OFDM for transmission and the achievable data rate is 54 Mbps, but compared to *b* and *g*, *a* operates on the unlicensed 5 GHz band. 802.11n improves data rates and coverage by adding multiple-input multiple-output (MIMO) antennas. It can operate on both 2.4 GHz and 5 GHz. Aside from the standards mentioned above, there are also amendments to the different standards that each handles a very specific purpose, but they do not specify any change in the PHY layer. [1] Table 1 gives an overview of the different versions.

Table 1. Overview of the different IEEE 802.11 versions

Version	Freq. (GHz)	Trans. Technique	Max. data rate (Mbps)
802.11a	5	OFDM	54
802.11b	2.4	DSSS	11
802.11g	2.4	OFDM	54
802.11n	2.4 & 5	OFDM	72.2 & 150

CHAPTER 3

3 The Cooperative Inter-Access Point Protocol

Before moving on to the design, some assumptions need to be taken into consideration for the protocol to work. First, the APs need to directly communicate with each other to share information. In practice, this is possible if APs are within communication range of each other i.e. their coverage areas should overlap. [3] Therefore, this will be assumed henceforth. With this assumption in mind, the protocol is suitable for medium to big city environments where many APs are deployed densely. The protocol is mainly aimed for uncoordinated environments since there is no centralized unit to control the network. The intention of the protocol is to improve the network performance so that all involved parties benefit from it. Thus, it is assumed that APs are cooperative and reliable, and that privacy and authentication issues are solved. These factors will not be further investigated and are beyond the scope of this thesis. It is furthermore assumed that the underlying functionalities of the IEEE 802.11 infrastructure mode are present, and that the protocol will add additional functions to this already existing standard. As mentioned in [13, p. 131], text-based protocols are flexible and easy to extend with new features. Therefore, it seems appropriate that the suggested protocol is text-based. To highlight the protocol frames, they are written with different font and capital letters.

3.1 Inter-Access Point Information Exchange

Sharing information between neighbouring APs is vital for the protocol functions to work. The idea is to create new text-based frames that contain information that is necessary for the APs to cooperate with each other. Sending out the standard beacon frame from AP to stations is simple because the stations operate on the same frequency channel as their associated AP. However, adjacent APs might work on different frequency channels, making it more difficult to exchange the new frames between the APs. One solution is to use one channel for exchanging the new frames, which means that all APs need to switch to a common channel to send and receive the frames. The major disadvantage with this method is that the APs and their affiliated stations face interruption every time they change the channel to send and receive frames. Delay-sensitive data would be affected the most because of data loss during the channel switches. Therefore, another suggestion is to have a *coordinator-AP* that handles the new frames by sending and receiving the frames on all channels and hence leaving the other APs on their operating channel. This will save resources in terms of saving the time it takes for the channel switching and reconnections, and also regarding the data losses that would occur because of the channel switching. The coordinator-AP will switch and scan every channel and collect frames from each neighbouring AP. It then uses the information in these frames to create a frame that is sent back to the APs on the different channels. New frames are created for establishing connection and sharing information between APs, handling the protocol functions and controlling the frame flow. In Section 3.3, the format of these new frames is presented, the frame processing at the receiver side is shown using flow charts, and detailed examples of how the protocol works is described and illustrated.

The reader might wonder which AP becomes the coordinator-AP. It is appropriate that the coordinator-AP is selected based on the number of surrounding neighbours. The idea is to select the AP with most neighbours as the coordinator-AP so that more APs will be covered and hence included

in the protocol. Virtual cluster networks are created by having coordinator-APs, and the coordinator-AP in each cluster will share the important information that is necessary for the protocol and its functions (Figure 5). In order for the APs to know how many neighbours each one has, a `COORDINATOR_ELECTION` frame of subtype `NEIGHBOURS` is exchanged between them. Once a coordinator-AP is chosen, it announces the selection with a `COORDINATOR_ELECTION` frame of subtype `COORDINATOR` to the cluster members. The frames for finding coordinator-AP are further explained in Section 3.3.2.1 and 3.3.3.1.

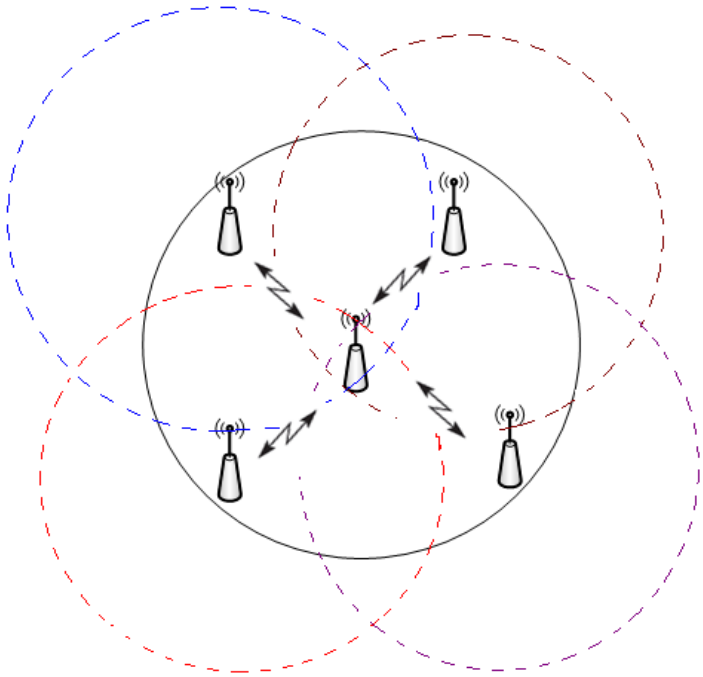


Figure 5. An illustration of a cluster with the coordinator-AP in the center

Any AP can take the role as coordinator-AP, and whenever there is a change in the topology a new AP can be selected as coordinator-AP. To see if there is any change in the topology, the NEIGHBOURS frame needs to be sent out periodically. Unlike stations, APs are usually stationary, leaving the AP topology the same for longer periods. Thus, there is no need for very frequent sharing of the NEIGHBOURS frame. For example, the standard beacon frame is sent out periodically every 100 ms [14]. To periodically transmit the NEIGHBOURS frame in millisecond basis would be considered a waste of resources. Hence, it is convenient to share the NEIGHBOURS frame on second basis.

One possible scenario could be that one AP is in range of two coordinator-APs. The AP will belong to the coordinator-AP it first receives a connection establishment frame from. If the AP receives a second connection establishment frame from another coordinator-AP, it will simply ignore this frame. The second coordinator-AP then excludes the AP from its schedulings.

3.2 The Protocol Functions

3.2.1 Channel Assignment

One of the most important properties of the IEEE 802.11 standard is channel selection. Having a proper channel selection for neighbouring APs is important in terms of reducing co-channel interference and maximizing throughput. [3][15] Therefore, the channel assignment mechanism will be one part of the proposed protocol. As many changes occur in a wireless environment, with stations acting in a stochastic manner regarding traffic load and distribution, it is important that the channel assignment is performed dynamically.

As mentioned earlier, the dense deployment of APs causes problems with only three non-overlapping frequency channels available, especially in uncoordinated deployments without placement planning of the APs. The APs should always seek for the channel with lowest traffic load and lowest interference level**.** Channel allocation is a well-studied topic and there are many proposed algorithms for channel selection, such as in [3], [16] and [17]. In [3] one AP takes the role as the *initiator*. The initiator collects channel information (traffic load, interference level etc.) from other APs using a backbone network or stations that are deployed in overlapping areas of APs. It then proposes a channel assignment and broadcasts this to the APs. The APs can decide whether to accept this proposal or not. The initiator then asks the APs to follow this proposal, or it suggests another proposal.

The initiator in [3] has been an inspiration for the coordinator-AP suggested in the previous section. The coordinator-AP periodically collects information such as traffic load and interference level from the cluster member via an `INFORMATION` frame of subtype `UPDATE-INFORMATION`. It then processes this information and sends back useful information, such as channel assignment, to the cluster members in a `MANAGEMENT` frame of subtype `PARAMETERS` on all channels. It is appropriate to exchange the `INFORMATION` and `MANAGEMENT` frame periodically on second basis, but more frequent than the `NEIGHBOURS` frame since many changes occur in a cell because of moving stations. In Figure 6, the coordinator-AP in the center transmits the channel assignment to its cluster members. The APs with overlapping coverage areas are assigned different channels to avoid co-channel interference. Once an AP receives its scheduling, it has the option to either use the proposed channel, or reject the proposal. If it accepts the suggested channel assignment, it enters a test period which is given in the received `MANAGEMENT` frame. During the test period, the AP uses the proposed channel in order to determine if the capacity is satisfying.

If the capacity is sufficient, the AP stays on the proposed channel until it receives another MANAGEMENT frame containing a new channel assignment from the coordinator-AP. On the other hand, if the AP senses that the capacity is too low on the proposed channel, the AP selects another channel based on new measurements on all channels.

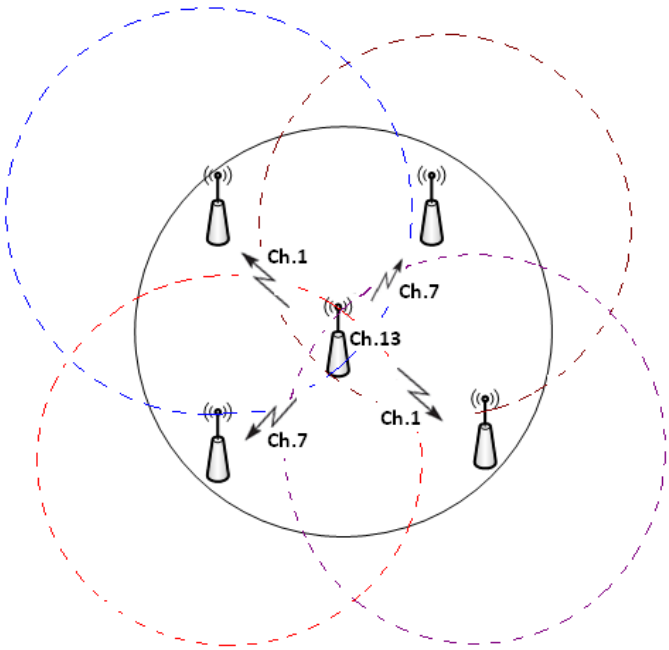


Figure 6. APs receiving the channel assignment from their coordinator-AP

3.2.2 Virtual TDMA

Neighbouring APs operating on the same channel will, as mentioned earlier, experience interference and degradation in performance. Hence, it is proposed to change the medium access mechanism between stations connected to neighbouring APs running on the same channel. In the

ordinary case, two or more neighbouring APs and their stations compete for medium access in CSMA/CA fashion, and co-channel interference is then introduced. Instead, it is suggested to assign each station associated to the neighbouring AP a virtual timeslot where they can use the medium in a scheduled fashion in order to better utilize the channel and mitigate co-channel interference. This time-scheduled mechanism will further on be referred as virtual TDMA. For virtual TDMA to work, the APs need to agree on the timeslot assignment in a distributed manner, and then inform their stations about the scheduling. In order to change to virtual TDMA, one of the APs in the shared channel issues a `VIRTUAL_TDMA` frame of subtype `REQUEST` to the other AP or APs (Figure 7).

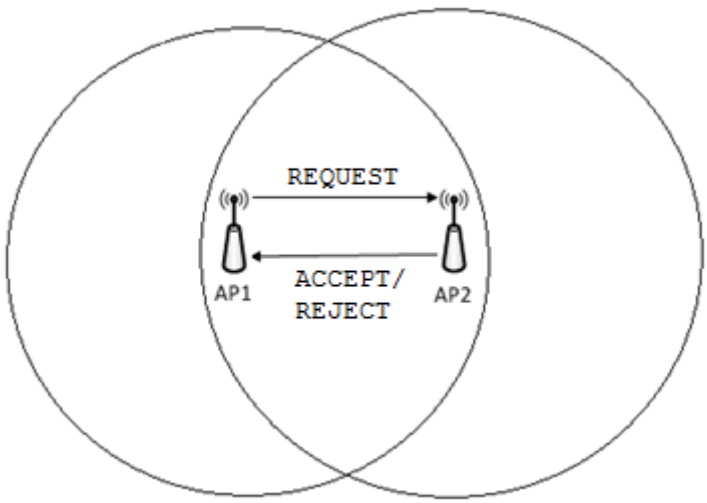


Figure 7. Virtual TDMA with request and response

When an AP receives a `REQUEST` frame, it sends back a response frame containing either `ACCEPT` or `REJECT` for this request. Once APs agree on having virtual TDMA, they need to agree on a timeslot scheduling. The schedulings for virtual TDMA are included in the `SCHEDULING` frame, and

the scheduling information for the stations is included in the STATIONS frame. In Figure 8 and 9, AP1 and AP2 have established a virtual TDMA network and their stations have been assigned timeslots, with guard time intervals in between. Simulations in Section 4.2 show that combining stations running on virtual TDMA with non-scheduled stations have severe impact on the throughput. Therefore, if an AP denies a virtual TDMA request, it would be more efficient for the AP to change channel. To improve efficiency, the timeslot scheduling could be adapted based on e.g. the traffic load and type of data in a cell.

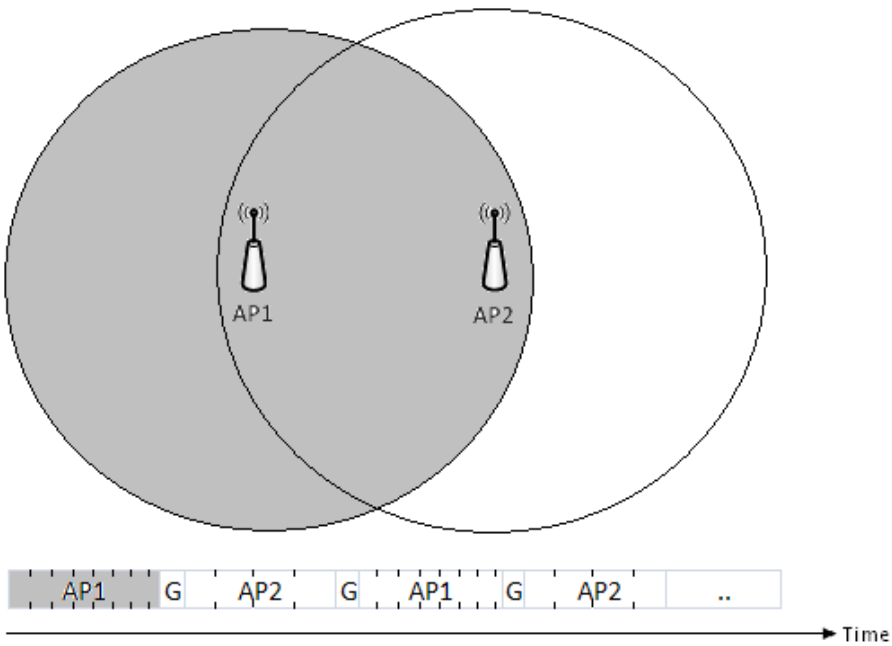


Figure 8. AP1 using its allocated timeslot

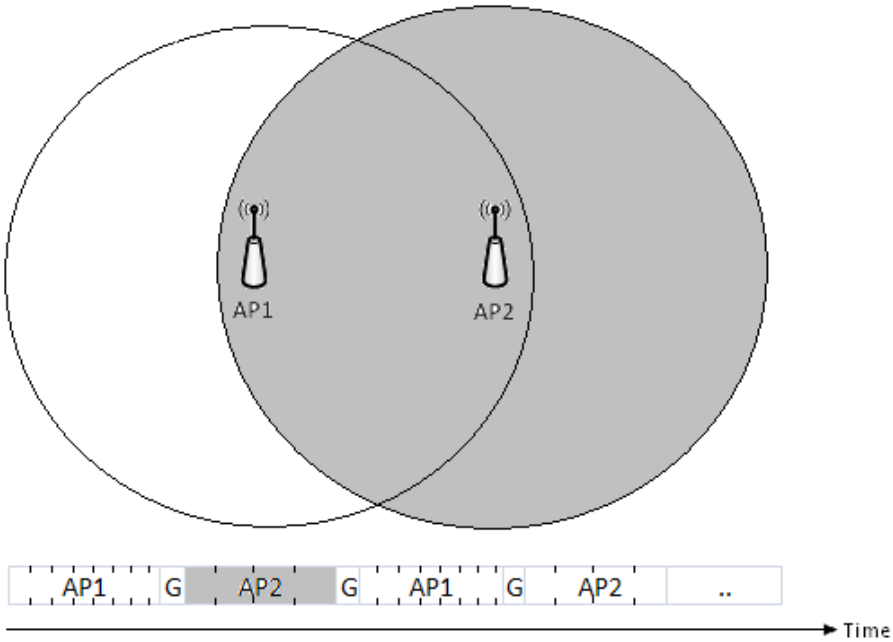


Figure 9. AP2 using its allocated timeslot

If an AP leaves the virtual TDMA network, it creates unused timeslots in the virtual TDMA frame i.e. resources are wasted. The other APs will notice the absence of an AP because they will not hear anything from it any longer. To solve this problem, the remaining APs need to agree on a new virtual TDMA scheduling after a period of time. There will be a timestamp field in the `SCHEDULING` frame, and it is used for synchronization between the involved APs to avoid timing mismatch. To further improve efficiency, stations of APs that are beyond interference range of each other should use the same timeslot. Figure 10 and 11 shows one scenario where this could occur. AP1 and AP2 have already established a virtual TDMA network. A new AP, AP3, is introduced and it is operating on the same channel as AP1 and AP2. AP3 issues a request to AP2, and AP2 responds with a `SCHEDULING` frame with new virtual TDMA parameters that is sent to both

AP3 and AP1. In the new timeslot scheduling, the stations of AP1 and AP3 are assigned the same timeslot.

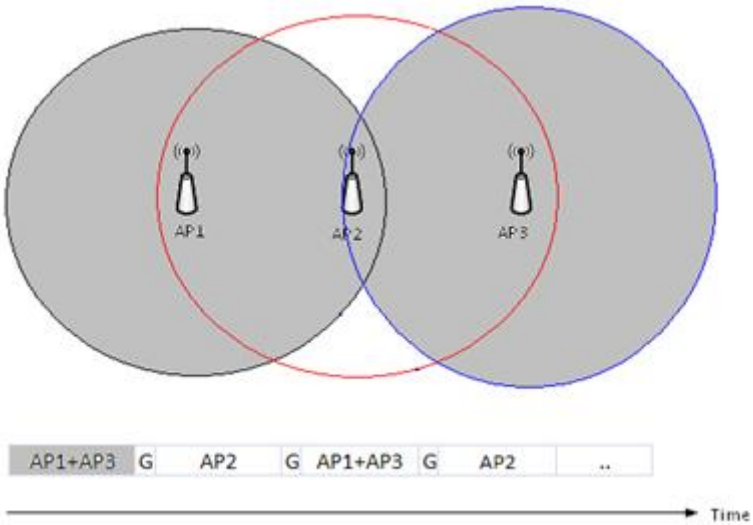


Figure 10. AP1 and AP3 using their allocated timeslot

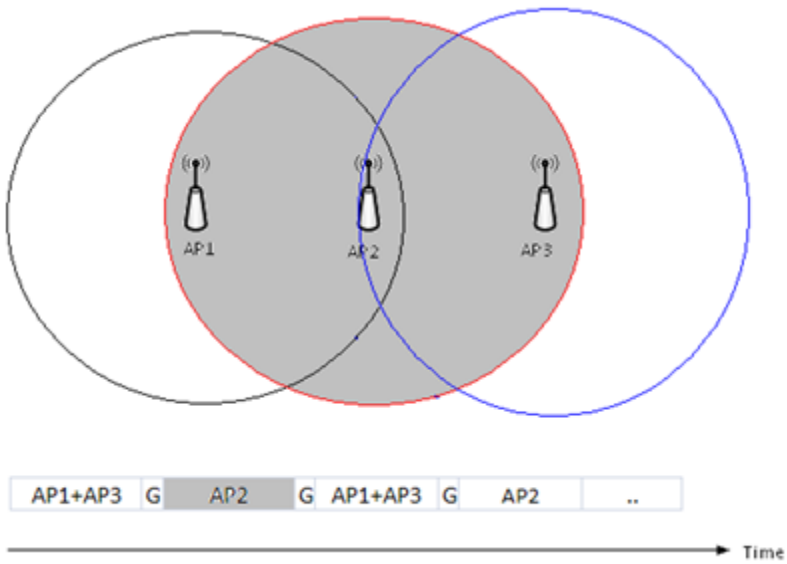


Figure 11. AP2 using its allocated timeslot

3.2.3 Switching off APs

When having many APs deployed close to each other with overlapping coverage areas, their associated stations will get interfered from other BSSs if they are running on the same channel. This situation is shown in Figure 12, where station 1 and 2 receives packets from AP1, and at the same time AP2 want to transmit packets to station 3 and 4. Since AP1 and station 1 and 2 are already using the medium, AP2 will not be able to send any data to station 3 and 4 because the medium is sensed busy. A suggestion to manage this interference problem is to switch off one of the APs, in this case AP2, and move the left stations to the surrounding APs, station 3 to AP1 and station 4 to AP3. Figure 13 illustrates this situation. Similar scenario is simulated in Section 4.2. To handle this switch-off method, a new frame, `SWITCH_OFF`, has been created to put an AP in stand-by mode. Another frame, `SWITCH_ON`, has been created to return an AP from stand-by mode.

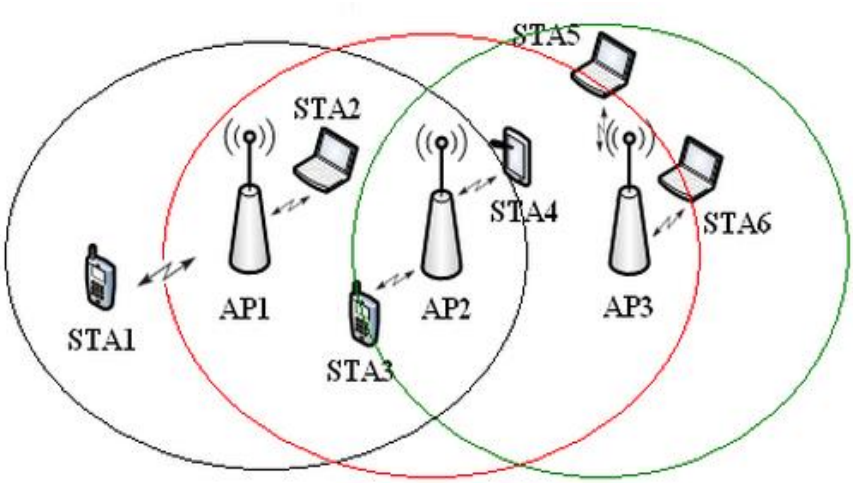


Figure 12. Three interfering APs

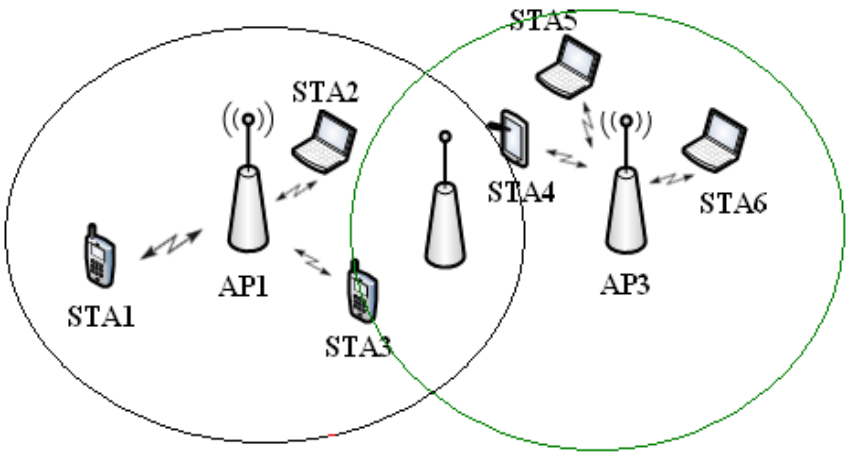


Figure 13. Shutting down AP2 and handing over the left stations

3.2.4 Load Balancing

It is uncommon that the load is evenly distributed among APs. One AP could have many mobile stations associated with it and experience heavy load, while its neighbouring AP could have connection with few mobile stations and thus experience light load. The mobile stations that are affiliated with the heavily loaded AP will suffer from many packet collisions and large delays when competing for medium access, which degrades the overall network performance. Redistributing some stations to adjacent APs can help to avoid this degradation from happening since the load in the network will be more balanced among the APs, resulting in an increase of overall network throughput and decrease of contention between the stations. This strategy is called load balancing and there are many different ways to perform it. [18] The authors in [18] propose a technique called cell-breathing. This implies that when an AP experiences heavy load,

it reduces its transmit power in order to decrease the coverage area so that less stations are connected to it. At the same time, the neighbouring APs would need to increase their transmit power to cover the left stations and coverage holes. For that reason, this method is not considered to be efficient for the protocol. All neighbours constantly need to be informed about all power changes occurring, and one power change in an AP would not only change its neighbours' power level, but also neighbours neighbours' power level. This creates a chain of power level changes involving many APs.

There is another method proposed in [19] where each AP is equipped with a so-called Load Balancing Agent (LBA) that periodically broadcasts the traffic load to the shared Ethernet backbone (distribution system). Each LBA uses these reports to decide if the traffic load is balanced among neighbouring APs. Depending on the load level, an AP can be in three possible states; under-loaded, balanced and over-loaded. To determine the load-state, each AP compares its load with the average load of the whole WLAN. If the load of an AP exceeds the average load by some specific value, the AP is recognized as over-loaded. If the load exceeds the average load by less than the specific value, it is considered balanced. If the load is below the average load, the AP is declared under-loaded. APs that are declared as over-loaded will force the handover of some stations to balance the load. This is done by terminating the association of these stations with a disassociation message sent out to them. Once disassociated, the stations need to find another AP to connect to, and only under-loaded APs accept new roaming stations. Balanced APs only accept new stations joining the network. To reduce the number of transferred stations from APs, [19] proposes a policy that selects the stations that have a throughput close to the difference between the average load and the load of the AP.

It is considered that a modification of the method used in [19] is suitable for the cooperative inter-access point protocol. Instead of having the traffic load information in the LBA and shared through the Ethernet backbone,

each AP measures its load and includes it in the INFORMATION frame which is sent to the coordinator-AP. The coordinator-AP extracts this information and uses it to calculate average load of its cluster. This average load is included in the MANAGEMENT frame, sent from coordinator-AP to the cluster members. Each AP can then compare the received average load with its own load in order to determine its state. Once the state is decided for each AP, the disassociation and handover of a station will work as proposed in [19] regarding which stations to handover. A new frame has been created to terminate a connection with a station, the LOAD_BALANCING frame of subtype TERMINATION. The LOAD_BALANCING frame is sent between APs and stations, compared to the other newly created frames that are sent between APs. Figure 14 demonstrates two APs, where AP1 is more heavily loaded than AP2. This is illustrated by having more stations connected to AP1. AP1 is declared to be over-loaded after comparing its load with the average load received from the coordinator-AP. It then chooses to terminate the association with two of its belonging stations by sending a TERMINATION frame, depicted in Figure 15. The two left stations then associate with AP2, shown in Figure 16.

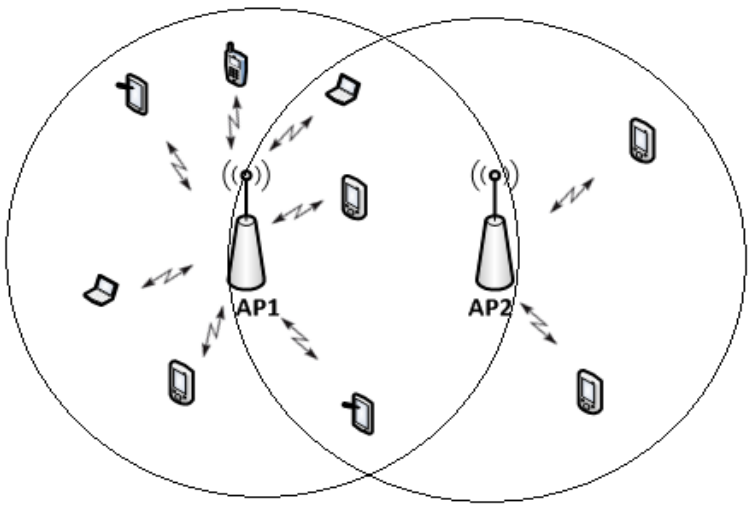


Figure 14. Load balancing - more stations connected to AP1

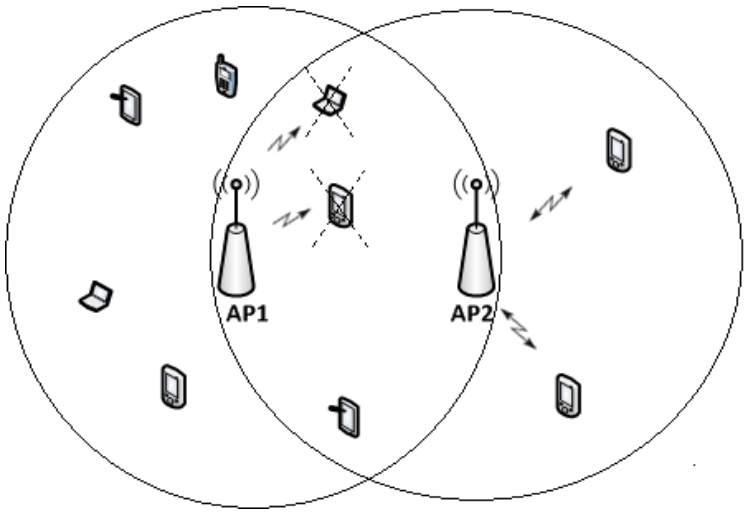


Figure 15. AP1 sending out `TERMINATION` frame to two stations

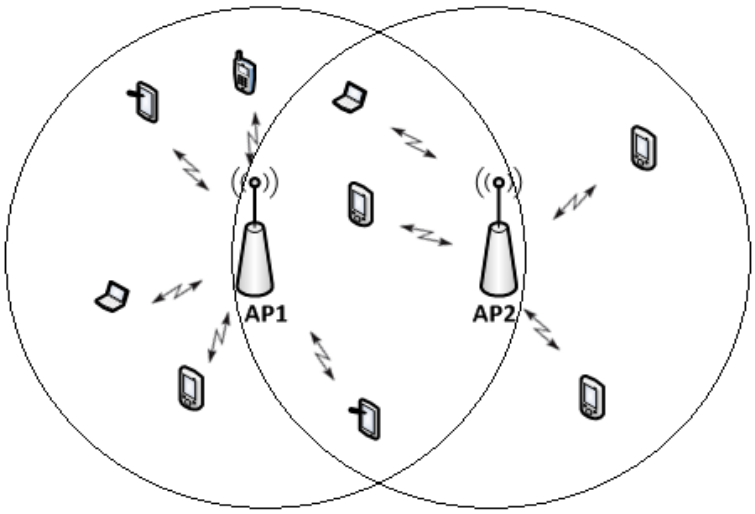


Figure 16. Load balancing completed

3.3 Protocol Operation

This section starts with a general description of the frame format followed by flow-charts illustrating how the different frames are used by the protocol. After this, detailed demonstrations of the packet exchanges are shown and described for the different protocol functions.

3.3.1 General Frame Format

All the frames in the protocol will follow the same general frame structure with the following fields. The fields are separated by the character ";" so that the APs know when a new field is expected.

VERSION

TYPE

SUBTYPE

SEQUENCE

DESTINATION

SOURCE

The first field, `VERSION`, indicates the version of the protocol and it contains two elements: the protocol name followed by the protocol version. The current protocol version is written as `CCP/1.0 (Cooperative Communication Protocol/1.0)`. In case of protocol changes, extended features or additional fields in the future, the version number will change. When an AP receives a frame it checks what is available before the first ";", in this case the `VERSION`. Depending on the `VERSION` the remaining frame could have different structure and fields, so when the `VERSION` is determined the receiving AP knows how to decode the rest of the frame.

After the VERSION field, the next field to decode is the TYPE field. This field states the type of the frame, and it contains no arguments. When an AP reads the TYPE, it will know what SUBTYPE to expect. Depending on the TYPE, different SUBTYPEs are possible. The SUBTYPE field further specifies the frame type, and it can hold parameters. The structure of the SUBTYPE field is written as follows, where the values are separated by ",":

```
SUBTYPE: value, value, ..., value
```

Depending on the SUBTYPE, different values are possible. It is not necessary for all SUBTYPEs to carry values. The reader might ask why there is a need for a single SUBTYPE field in some of the frames. The reason for this is to keep a general frame structure that works for all functions, and to make the protocol more flexible for future changes. The SEQUENCE field is used to keep track on the frame flow. The confirmation frames (ACK, OK, ACCEPT and REJECT) use the sequence number included in the SEQUENCE field and the SUBTYPE to acknowledge or deny the received frames. The DESTINATION field holds the IP address of the receiving AP, and the SOURCE field holds the IP address of the sending AP. Table 2 shows the defined TYPE frames and their SUBTYPEs. The different frames are explained in the two following sections.

Table 2. The defined protocol frames

TYPE	SUBTYPE	PURPOSE
COORDINATOR_ELECTION	NEIGHBOURS COORDINATOR	For finding and announcing the coordinator-AP. Parameters for synchronization also included.
CONTROL	ACK NACK OK	For confirming and controlling the frame flow.
INFORMATION	UPDATE-INFORMATION	For sharing information such as traffic load and interference level. Sent from cluster members to coordinator-AP
MANAGEMENT	PARAMETERS	For sharing information such as timestamp, channel number and avg. load. Sent from coordinator-AP to cluster members.
VIRTUAL_TDMA	REQUEST ACCEPT REJECT SCHEDULING STATIONS	For establishing a virtual TDMA network and sharing the necessary virtual TDMA parameters. Sent between APs operating on the same channel and between AP and stations.
SWITCH_OFF	STAND-BY	For switching off an AP when the interference level is too high. Sent from coordinator-AP to cluster members.
SWITCH_ON	WAKE-UP	For switching on an AP from stand-by mode. Sent from coordinator-AP to cluster members.
LOAD_BALANCING	TERMINATION	For terminating connection with a station to balance the load. Sent from APs to stations.

3.3.2 Frame Processing

In this section, the process upon receiving a packet is demonstrated using flow charts. This gives an intuitive illustration of the decoding of a packet, and the actions followed.

3.3.2.1 COORDINATOR_ELECTION

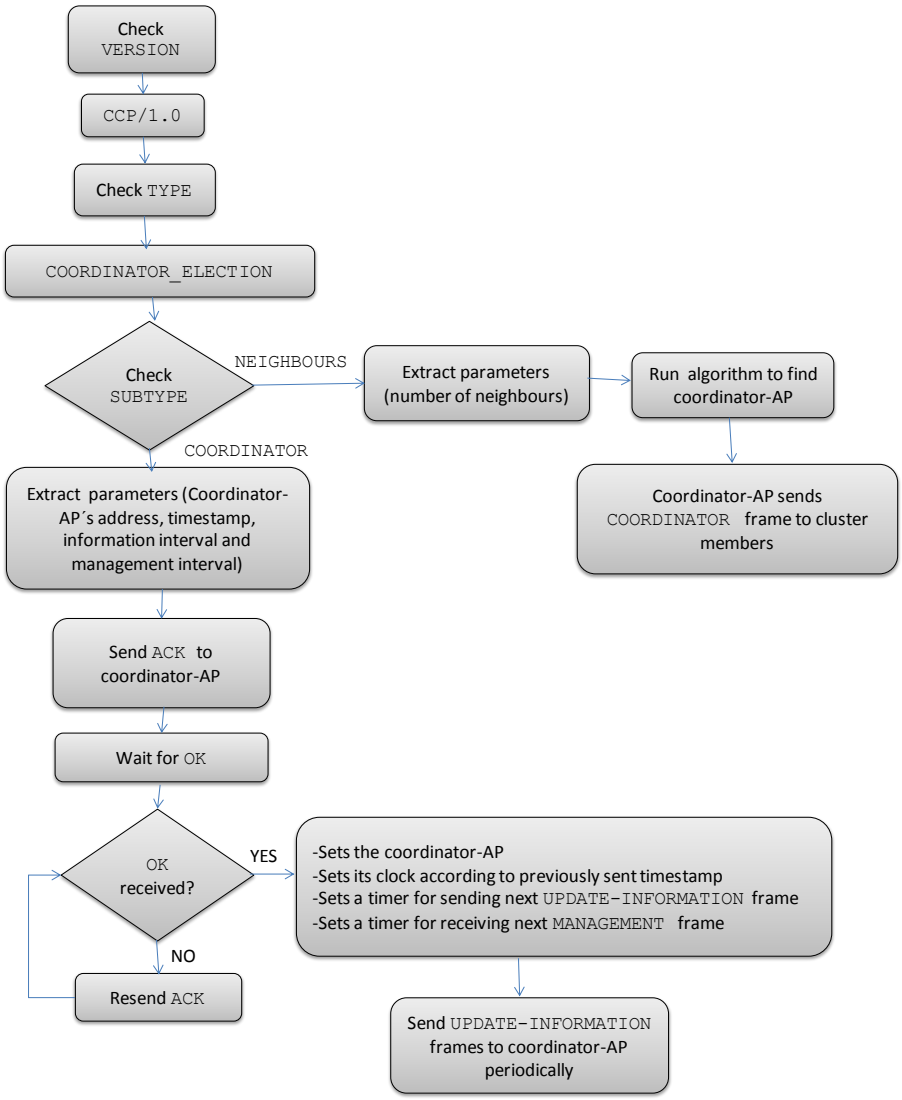


Figure 17. Actions when receiving COORDINATOR_ELECTION frame

3.3.2.2 CONTROL

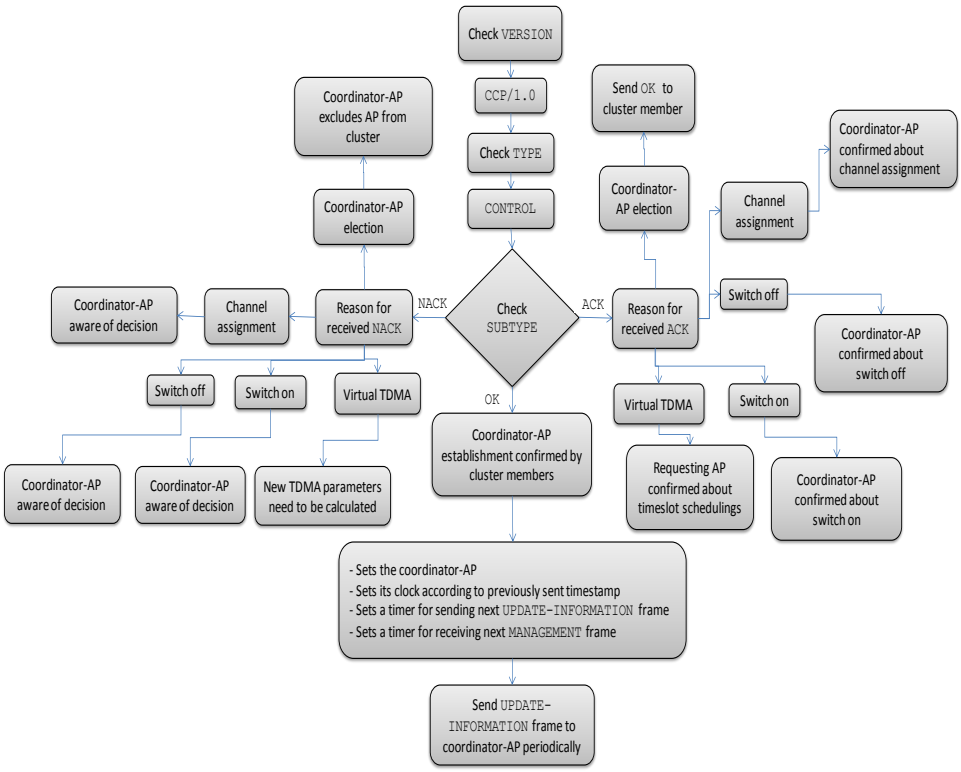


Figure 18. Actions when receiving CONTROL frame

3.3.2.3 INFORMATION

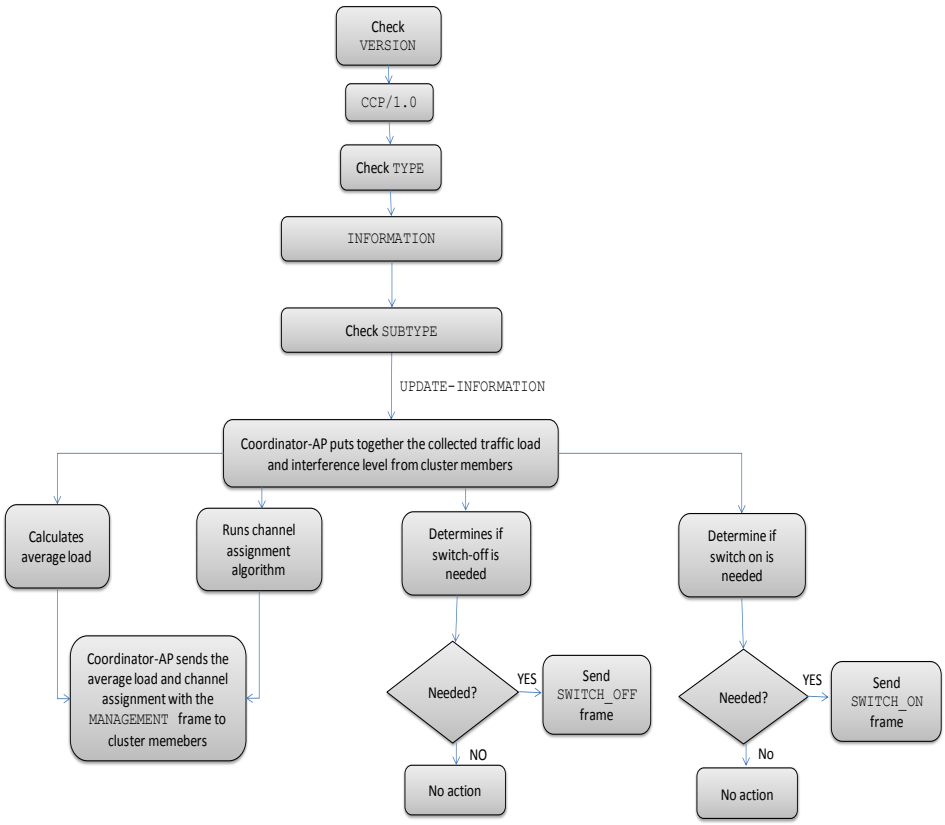


Figure 19. Actions when receiving INFORMATION frame

3.3.2.4 MANAGEMENT

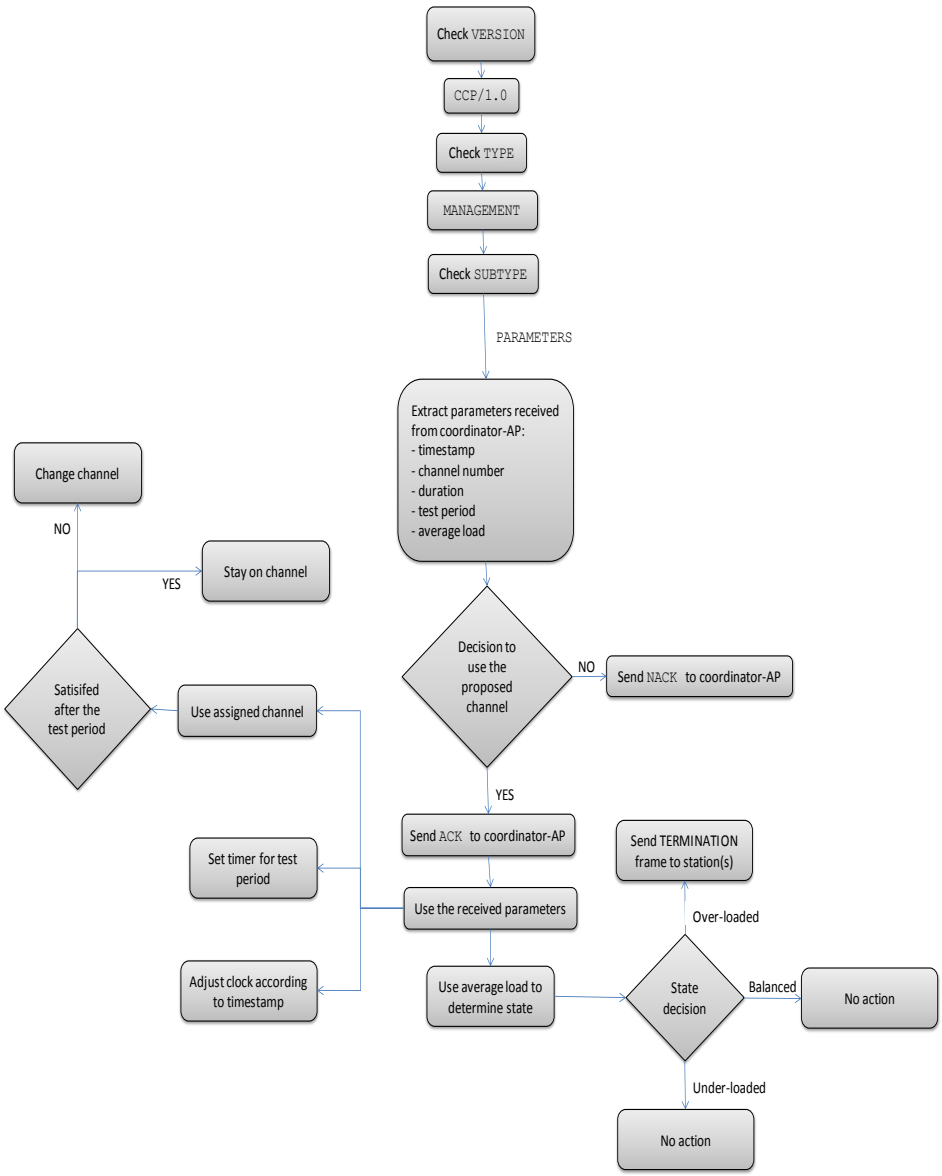


Figure 20. Actions when receiving MANAGEMENT frame

3.3.2.5 VIRTUAL_TDMA

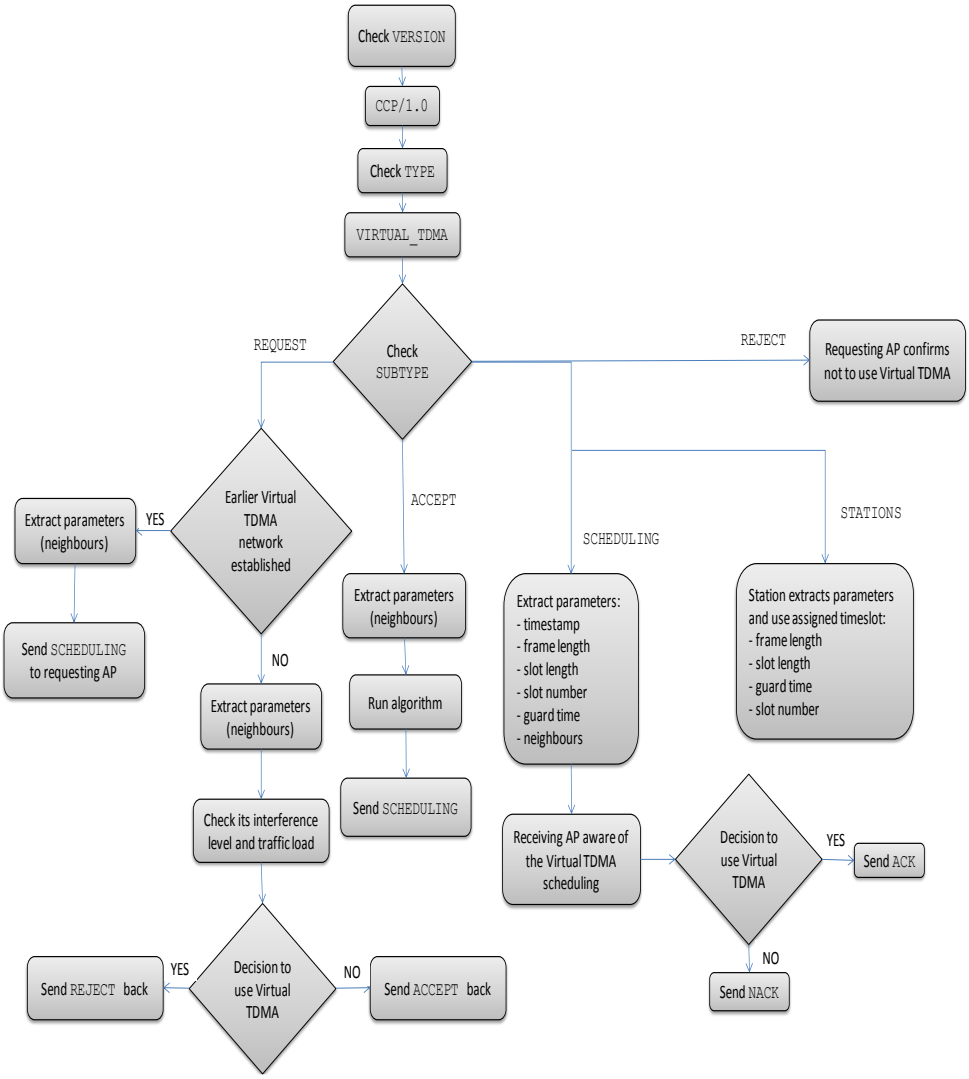


Figure 21. Actions when receiving VIRTUAL_TDMA frame

3.3.2.6 SWITCH_OFF & SWITCH_ON

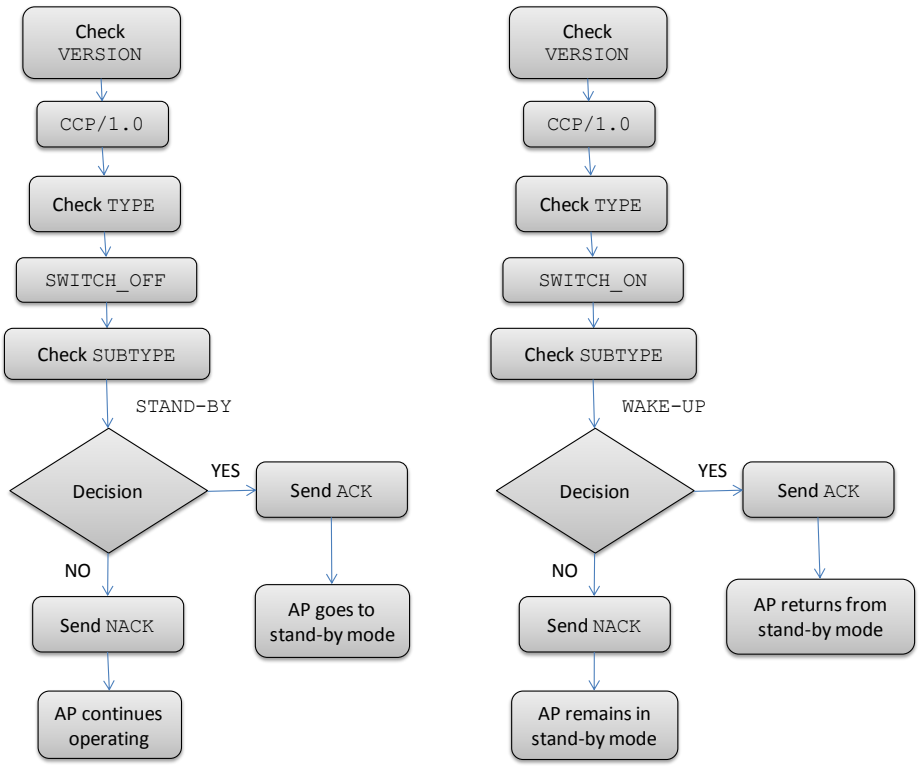


Figure 22. Actions when receiving SWITCH_OFF and SWITCH_ON frames

3.3.2.7 LOAD_BALANCING

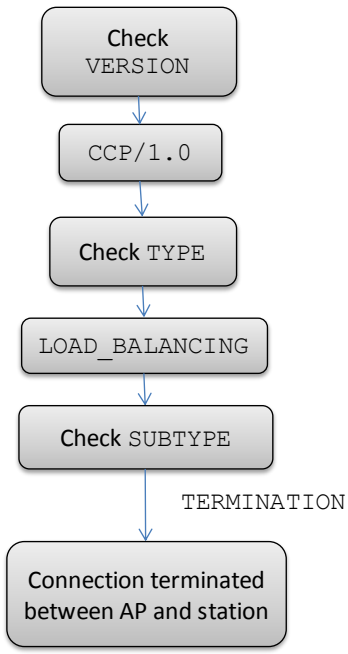


Figure 23. Actions when receiving LOAD_BALANCING frame

3.3.3 Function Specific Examples

In the following section, detailed descriptions of the protocol functions are presented. The packet exchanges to obtain a certain function are demonstrated and the information in the packets is explained. The units of the values are included in the frame demonstration for clarity, but not in the actual frames. The required overhead for a specific protocol frame is first presented separately, and then at the end of each section the total overhead that includes all frames needed for obtaining a protocol function is

calculated. In the overhead calculations, each character corresponds to 8 bits and maximum frame length for each function is assumed.

3.3.3.1 Finding coordinator-AP:

1) To find the coordinator-AP, the APs in the surrounding area broadcast the number of neighbours to all neighbouring APs on all channels. The APs send a `COORDINATOR_ELECTION` frame to notify other APs that a coordinator-AP is to be decided. Moreover, by using the subtype `NEIGHBOURS`, the APs specify how many neighbouring APs they have. This is necessary since the algorithm for choosing the coordinator-AP is based on the number of neighbours each AP has. The number is saved as a String. Required overhead: 642 bits.

```
VERSION: CCP/1.0
COORDINATOR_ELECTION
NEIGHBOURS: number (String)
SEQUENCE: number (String)
DESTINATION: Broadcast (address)
SOURCE: AP (address)
```

2) All involved APs run an algorithm that finds the optimal AP as coordinator-AP based on number of neighbours. This algorithm works in a distributed fashion and is not further explained since it is beyond the scope of the thesis.

3) When the coordinator-AP is determined, it announces to every cluster member that it is the coordinator-AP by sending a COORDINATOR_ELECTION frame of subtype COORDINATOR. This field holds four values. The first value is the address of the coordinator-AP, and each cluster member saves this address for later transmissions to the coordinator-AP. The second value is the current time of the coordinator-AP. It is important that this timestamp is included so that synchronization is maintained within the cluster. Furthermore, the information-interval tells the cluster members when to send the INFORMATION frame to the coordinator-AP. The management-interval informs the cluster members when to receive the MANAGEMENT frame from the coordinator-AP. Required overhead: 914 bits.

VERSION: CCP/1.0
COORDINATOR_ELECTION
COORDINATOR: coordinator-AP (address), timestamp (clock), information-interval (ms), management-interval (ms)
SEQUENCE: number (String)
DESTINATION: AP(s) (address)
SOURCE: Coordinator-AP (address)

4) To confirm the coordinator-AP, the cluster members send a CONTROL frame containing an ACK back to the coordinator-AP. The ACK field includes the sequence number and the subtype of the received frame that is to be acknowledged, in this case the COORDINATOR frame. It is important for the coordinator-AP to receive the ACK because it will handle many functions in the future. Cluster members also have the possibility to reject coordinator-AP with the NACK frame. Required overhead: 586 and 594 bits respectively.

```
VERSION: CCP/1.0
CONTROL
ACK: number (String), COORDINATOR
SEQUENCE: number (String)
DESTINATION: Coordinator-AP (address)
SOURCE: AP (address)
```

```
VERSION: CCP/1.0
CONTROL
NACK: number (String), COORDINATOR
SEQUENCE: number (String)
DESTINATION: Coordinator-AP (address)
SOURCE: AP (address)
```

5) The coordinator-AP sends a CONTROL frame containing an OK back to each cluster member. The OK field contains the sequence number and the subtype of the frame it is confirming, in this case the ACK frame. This three-way handshake makes sure that connection between coordinator-AP and APs has been established. Required overhead: 514 bits.

```
VERSION: CCP/1.0
CONTROL
OK: number (String), ACK
SEQUENCE: number (String)
DESTINATION: AP(s) (address)
SOURCE: Coordinator-AP (address)
```

Total overhead: 642+914+594+514=2664 bits.

3.3.3.2 Channel Assignment

1) The cluster members send the interference level and traffic load in their cell to the coordinator-AP with the INFORMATION frame. The saved address from the COORDINATOR frame is used as destination address. Required overhead: 754 bits.

```
VERSION: CCP/1.0
INFORMATION
UPDATE-INFORMATION: traffic load (bytes), interference
level (dBm)
SEQUENCE: number (String)
DESTINATION: Coordinator-AP (address)
SOURCE: AP(s) (address)
```

2) The coordinator-AP runs an algorithm to find the best and optimal channel assignment. This algorithm is not further explained since it is beyond the scope of the thesis.

3) The coordinator-AP transmits a MANAGEMENT frame on all channels containing information such as the timestamp, channel number, duration and test period for each AP. The duration value is the time the AP will stay on the proposed channel to transmit and receive data. The timestamp is once again used for synchronization within the cluster. The average load of the cluster is included for later use in the Load Balancing function (Section 3.3.3.6). Required overhead: 826 bits.

```
VERSION: CCP/1.0
MANAGEMENT
```

```
PARAMETERS: timestamp (clock of coordinator-AP),
channel number (String), duration (ms), test period
(ms), average load (bytes)
SEQUENCE: number (String)
Destination: AP(s) address
Source: Coordinator-AP (address)
```

4) The cluster members send an ACK back to the coordinator-AP for confirmation about the channel assignment received in the PARAMETERS frame. The cluster members can also send back NACK to reject the channel assignment. Required overhead: 578 and 586 bits respectively.

```
VERSION: CCP/1.0
CONTROL
ACK: number (String), PARAMETERS
SEQUENCE: number (String)
DESTINATION: Coordinator-AP (address)
SOURCE: AP(s) (address)
```

```
VERSION: CCP/1.0
CONTROL
NACK: number (String), PARAMETERS
SEQUENCE: number (String)
DESTINATION: Coordinator-AP (address)
SOURCE: AP(s) (address)
```

5) After sending the ACK frame, the cluster members use the assigned channels at least for the test period value given in the MANAGEMENT frame. The cluster members can then either choose to stay or change the channel.

Total overhead: 754+826+586=2166 bits.

3.3.3.3 Virtual TDMA

There are two different cases for establishing a virtual TDMA network; when no previous network has been established, and when there is already a network established. These cases are explained separately in the following sections.

3.3.3.3.1 No previously established virtual TDMA network

1) The APs measure the interference level and traffic load in their cell, and if it is not satisfactory, a virtual TDMA request is issued to neighbouring APs on the same channel. The REQUEST subtype frame contains the addresses of the neighbours operating on the same channel. This is needed for having knowledge of each other's neighbours, which is utilized for later timeslot-assignments. Required overhead: 13202 bits.

```
VERSION: CCP/1.0
VIRTUAL_TDMA
REQUEST: neighbour1 (address), neighbour2 (address),…,
SEQUENCE: number (String)
DESTINATION: AP1 (address)
SOURCE: AP2 (address)
```

2) The AP receiving the REQUEST checks its interference and traffic load level. If the AP evaluates that virtual TDMA will improve its performance,

it replies with an ACCEPT, otherwise it sends a REJECT to the requesting AP. If ACCEPT is sent, the AP's neighbours are included in the frame. Required overhead: 13282 and 618 bits respectively.

```
VERSION: CCP/1.0
VIRTUAL_TDMA
ACCEPT: number (String), REQUEST, neighbour1
(address), neighbour2 (address), …,
SEQUENCE: number (String)
DESTINATION: AP2 (address)
SOURCE: AP1 (address)
```

```
VERSION: CCP/1.0
VIRTUAL_TDMA
REJECT: number (String), REQUEST
SEQUENCE: number (String)
DESTINATION: AP2 (address)
SOURCE: AP1 (address)
```

3) If ACCEPT is received by the requesting AP, it sends back a SCHEDULING frame with all the necessary parameters to establish and preserve a virtual TDMA network. The timestamp is used for synchronization. The APs in the virtual TDMA network know when to use the channel with the given frame length, slot length, slot number and guard time. Required overhead: 13960 bits.

```
VERSION: CCP/1.0
VIRTUAL_TDMA
```

```
SCHEDULING: timestamp(s), frame length (ms), slot
length (ms), slot number (String), guard time (ms),
neighbour1 (address), neighbour2 (address), …,
SEQUENCE: number (String)
DESTINATION: AP1 (address)
SOURCE: AP2 (address)
```

4) To confirm that an AP has received the TDMA parameters for the schedulings, an ACK message is sent back. If the AP is not satisfied with the timeslot scheduling, it has the possibility to send a NACK frame. Required overhead: 578 and 586 bits respectively.

```
VERSION: CCP/1.0
CONTROL
ACK: number (String), SCHEDULING
SEQUENCE: number (String)
DESTINATION: AP2 (address)
SOURCE: AP1 (address)
```

```
VERSION: CCP/1.0
CONTROL
NACK: number (String), SCHEDULING
SEQUENCE: number (String)
DESTINATION: AP2 (address)
SOURCE: AP1 (address)
```

5) To get the stations scheduled in each AP, an additional frame is broadcasted from the AP to its stations containing the slots for each station in a vector. Required overhead: 3058 bits .

```
VERSION: CCP/1.0
VIRTUAL_TDMA
STATIONS: frame length (ms), slot length (ms), guard
time (ms), {slot number1 (String) , slot number2
(String), slot number3 (String), …, }
SEQUENCE: number (String)
DESTINATION: Broadcast (address)
SOURCE: AP (address)
```

Total overhead: 13202+13282+13960+586+3058 =44088 bits.

3.3.3.3.2 Already established virtual TDMA network

1) As in previous section, an AP issues a REQUEST frame to adjacent APs on the same channel if the interference or traffic level is not satisfactory in their cell. Required overhead: 13162 bits.

```
VERSION: CCP/1.0
VIRTUAL_TDMA
REQUEST: neighbour1 (address), neighbour2 (address),…,
SEQUENCE: number (String)
DESTINATION: AP1 (address)
SOURCE: AP2 (address)
```

2) If the AP receiving the request is already included in a virtual TDMA network, it does not reply with an ACCEPT or REJECT as in 3.3.2.3.1. Instead, it directly replies with a SCHEDULING frame containing all the TDMA-parameters. Required overhead: 13384 bits.

```
VERSION: CCP/1.0
VIRTUAL_TDMA
SCHEDULING: timestamp(s), frame length (ms), slot
length (ms), slot number (String), guard time (ms),
neighbour1 (address), neighbour2 (address), …,
SEQUENCE: number (String)
DESTINATION: AP2 (address)
SOURCE: AP1 (address)
```

3) To confirm that an AP has received the TDMA parameters for the schedulings, an ACK message is sent back. If the AP is not satisfied with the timeslot scheduling, it can send a NACK frame back. Required overhead: 578 and 586 bits respectively.

```
VERSION: CCP/1.0
CONTROL
ACK: number (String), SCHEDULING
SEQUENCE: number (String)
DESTINATION: AP1 (address)
SOURCE: AP2 (address)
```

```
VERSION: CCP/1.0
```

```
CONTROL
NACK: number (String), SCHEDULING
SEQUENCE: number (String)
DESTINATION: AP1 (address)
SOURCE: AP2 (address)
```

5) To get the stations scheduled in each AP, an additional frame is broadcasted from the AP to its stations containing the slots for each station in a vector. Required overhead: 3058 bits.

```
VERSION: CCP/1.0
VIRTUAL_TDMA
STATIONS: frame length (ms), slot length (ms), guard time (ms), {slot number1 (String) , slot number2 (String), slot number3 (String), …, }
SEQUENCE: number (String)
DESTINATION: Broadcast (address)
SOURCE: AP (address)
```

Total overhead: 13162+13384+586+3058=30190 bits.

3.3.3.4 Switching off APs:

1) The cluster members send the interference level and traffic load in their cell to the coordinator-AP with the INFORMATION frame. The saved coordinator-AP address obtained from the COORDINATOR subtype frame is used as destination address. Required overhead: 754 bits.

```
VERSION: CCP/1.0
INFORMATION
UPDATE-INFORMATION: traffic load (bytes), interference
level (dBm)
SEQUENCE: number (String)
DESTINATION: Coordinator-AP (address)
SOURCE: AP (address)
```

2) The coordinator-AP extracts the received information from the APs and, through an algorithm, determines the suitable AP to shut down. This algorithm is not further explained since it is beyond the scope of the thesis.

3) The coordinator-AP sends a SWITCH_OFF frame that tells the chosen AP to enter a stand-by mode for a specific duration that is given in the frame. Required overhead: 554 bits.

```
VERSION: CCP/1.0
SWITCH_OFF
STAND-BY: duration (ms)
SEQUENCE: number (String)
DESTINATION: AP (address)
SOURCE: Coordinator-AP (address)
```

4) Before the AP goes to stand-by mode, it sends an ACK message to the coordinator-AP. There is a possibility to reject the switch-off by sending back an NACK. Required overhead: 562 and 570 bits respectively.

```
Version: CCP/1.0
CONTROL
```

ACK: number (String), STAND-BY

SEQUENCE: number (String)

DESTINATION: Coordinator-AP (address)

SOURCE: AP (address)

Version: CCP/1.0

CONTROL

NACK: number (String), STAND-BY

SEQUENCE: number (String)

DESTINATION: Coordinator-AP (address)

SOURCE: AP (address)

Total overhead: 754+554+570=1878 bits.

3.3.3.5 Turn on a switched off AP:

1) The cluster members share their interference level and traffic load to the coordinator-AP with the INFORMATION frame. The destination address is obtained from the COORDINATOR subtype frame that is previously received from the coordinator-AP. Required overhead: 754 bits.

VERSION: CCP/1.0

INFORMATION

UPDATE-INFORMATION: traffic load (bytes), interference level (dBm)

SEQUENCE: number (String)

DESTINATION: Coordinator-AP (address)

SOURCE: AP (address)

2) The coordinator-AP uses an algorithm to determine the suitable AP to wake up from stand-by mode. This algorithm is not further explained since it is beyond the scope of the thesis.

3) The coordinator-AP sends a SWITCH_ON frame that tells the chosen AP to wake up from stand-by mode. Having SUBTYPE for SWITCH_ON makes it flexible for additional subtypes that could be implemented in the future. Another possible SUBTYPE could be more time-specific in case the coordinator-AP wants one AP to wake-up for a particular time. Required overhead: 506 bits.

```
VERSION: CCP/1.0
SWITCH_ON
WAKE-UP
SEQUENCE: number (String)
DESTINATION: AP (address)
SOURCE: Coordinator-AP (address)
```

4) The chosen AP sends back an ACK to the coordinator-AP to confirm the activation, or NACK to reject the WAKE-UP frame. Required overhead: 554 and 562 bits respectively.

```
VERSION: CCP/1.0
CONTROL
ACK: number (String), WAKE-UP
SEQUENCE: number (String)
DESTINATION: Coordinator-AP (address)
SOURCE: AP (address)
```

```
VERSION: CCP/1.0
CONTROL
NACK: number (String), WAKE-UP
SEQUENCE: number (String)
DESTINATION: Coordinator-AP (address)
SOURCE: AP (address)
```

Total overhead: 754+506+562=1822 bits.

3.3.3.6 Load Balancing:

1) The cluster members transmit the interference level and traffic load in their cell to their coordinator-AP with the INFORMATION frame. The acquired coordinator-AP address from the COORDINATOR subtype frame field is used as destination address. Required overhead: 754 bits.

```
VERSION: CCP/1.0
INFORMATION
UPDATE-INFORMATION: traffic load (bytes), interference
level (dBm)
SEQUENCE: number (String)
DESTINATION: Coordinator-AP (address)
SOURCE: AP (address)
```

2) Based on the information received from all cluster members, the coordinator-AP calculates the average load of the cluster and includes it in the MANAGEMENT frame. Required overhead: 826 bits.

```
VERSION: CCP/1.0
MANAGEMENT
PARAMETERS: timestamp (clock of coordinator-AP), channel number (String), duration (ms), test period (ms), average load (bytes)
SEQUENCE: number (String)
DESTINATION: AP (address)
SOURCE: Coordinator-AP (address)
```

3) Each AP compares its load with the received average load and determines its state (under, balanced or over-loaded).

4) If the AP is over-loaded, it terminates the connection with one or more of the affiliated stations with a `LOAD_BALANCING` frame. The `SUBTYPE` field is also left here in case of any additional features in the future, e.g. handing over stations to a specific AP. Required overhead: 578 bits.

```
VERSION: CCP/1.0
LOAD_BALANCING
TERMINATION
SEQUENCE: number (String)
DESTINATION: Station(s) (address)
SOURCE: AP (address)
```

5) The chosen stations are then handed over to the surrounding APs.

Total overhead: 754+826+578=2158 bits.

3.4 Future Protocol Functions

Having different versions of 802.11 in one network could degrade the overall performance. [20] shows that the throughput of a station with high transmission rate (for example 802.11n) will suffer when having a low rate station (for example 802.11b) connected to the same network. This is due to the CSMA/CA medium access mechanism which guarantees that the long term medium access probability is the same for all stations. When a low rate station gets access to the channel, it will occupy the channel for a long time and this in turn penalizes other stations using higher rates. Making the APs cooperate with each other to handle this problem could be something to consider in the future to further increase the network performance. One suggestion is to, if possible, move low-rate stations to neighbouring APs with more low-rate stations, and vice versa. The APs would have to keep track on the 802.11 version each station is running on in the network, and share this information to adjacent APs. The APs could then compare this information and agree on how to distribute the stations among them.

CHAPTER 4

4 Simulations, Results & Discussion

The network simulator ns-3 was used for the simulations [6] and the common parameters for all simulations can be seen in Table 3.

Table 3. Common parameters for all simulations

Parameters	**Values**
IEEE 802.11 version	802.11b
Operating frequency	2.4 GHz
Transmission Technique	DSSS
Data rate	1 Mbps
Transmission protocol	UDP
Received signal strength threshold	-80 dBm
Propagation loss model	Log distance
Propagation delay model	Constant speed
Positioning model for APs	Constant positions
Positioning model for stations	Uniform and random positions
AP radius for station allocation	50 m
Speed of stations	2 m/s
Movement direction of stations	Random
Simulation time	100 s

Each simulation was run 10 times, and the final result is the mean of these runs. To validate the simulation results, the 95 % confidence interval is calculated and included in the plots and summary tables. The general topology in the simulations is depicted in Figure 24. What will vary in the different simulations is the number of stations and APs, channel of operation, timeslot length for each station and the amount of data sent.

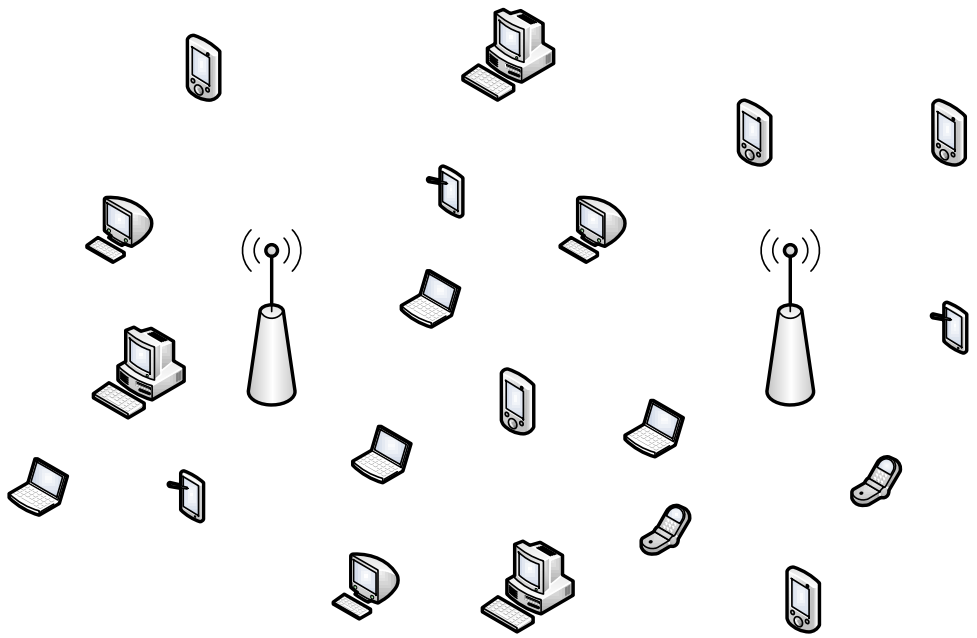


Figure 24. General simulation topology

4.1 Interference-free vs interference-present transmission

In the first case, a comparison is made between an interference-free and interference-present transmission. Two APs, operating on different channels, are deployed 5 m away from each other in the interference-free case. The stations send UDP packets of size 2000 bytes on a radio channel with the capacity at 1 Mbps to their AP, and the number of associated stations varies for each simulation. To introduce interference, both APs are switched to the same channel, making them both contend for medium access. The interference-free transmission can be seen as a result of using the proposed channel assignment function, explained in Section 3.2.1. Recall from Section 3.3.3.2 that the frames needed for obtaining a channel assignment consumes 2166 bits in total. A proposal is to perform the channel assignment function periodically every 10 seconds, and since the simulation time is 100 seconds, it will be performed 10 times resulting in:

$$2166 * 10 = 21.66 \text{ kb}/100\text{s}$$

This means that the channel assignment function consumes 0.217 kbps. This overhead is therefore subtracted from the final mean result of the interference-free case. Figure 25 shows the results of the two cases, with the throughput measured in average received bits per second for one station.

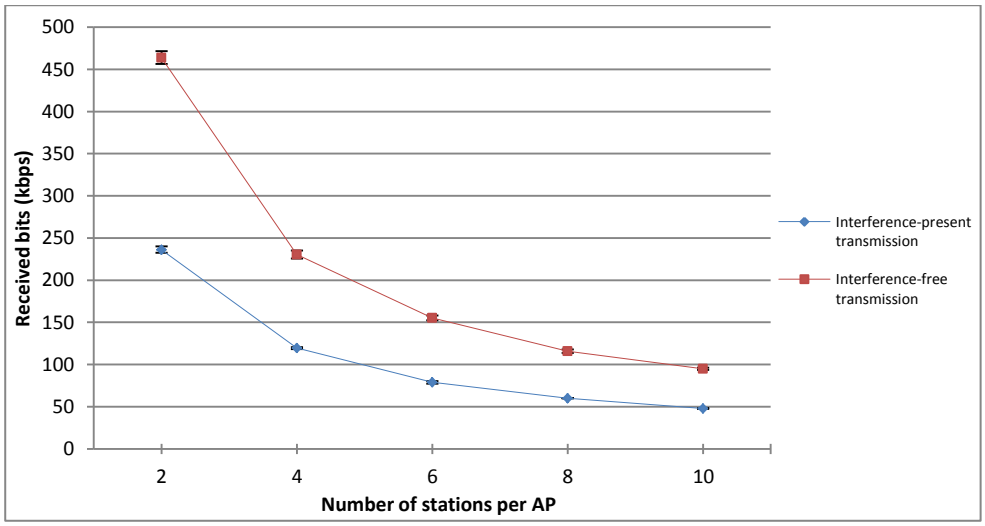


Figure 25. Interference-free vs interference-present transmission

The contention for medium access increases with higher station densities, which results in a lower throughput for both cases, and this can be seen as both curves start to deteriorate. As expected, introducing interference degrades the overall performance for all station densities. When having two APs on the same channel, they are considered to be on one network and all stations will contend for medium access under the same network. To clarify this, one could compare the interference-present case with 4 stations per AP i.e. 8 stations in total in the network, with the case of interference-free transmission with 4 stations per AP and network. As can be seen in Figure 25, the two cases show approximately 120 kbps and 240 kbps respectively, which means that the stations in the first network only reach half the throughput. This is why it is desired to have two neighbouring APs on different channels. Even with the protocol overhead included in the results, the interference-free transmission shows much better performance. For this reason, it is necessary for the protocol to include a channel assignment function that aims for mitigating interference between APs and decreasing contention between APs and stations.

Similar situation is simulated once again (Figure 26), but the stations send UDP packets of varying size to their AP, ranging between 1-2000 bytes. This means that for every transmission, a random number between 1 and 2000 is selected for packet size length. The time between the transmissions is not reduced when the packets size decreases. The behavior of the curves is comparable to the curves in Figure 25, but the throughput is lower in both cases due to each station not using the maximum possible packet transfer. This is also the case when making the transmission from each station random but with maximum packet size length, which basically means that every time a station is to transmit data, there is a 50% probability that no packet will be sent. Hence this results in a total traffic less than 1 Mbps.

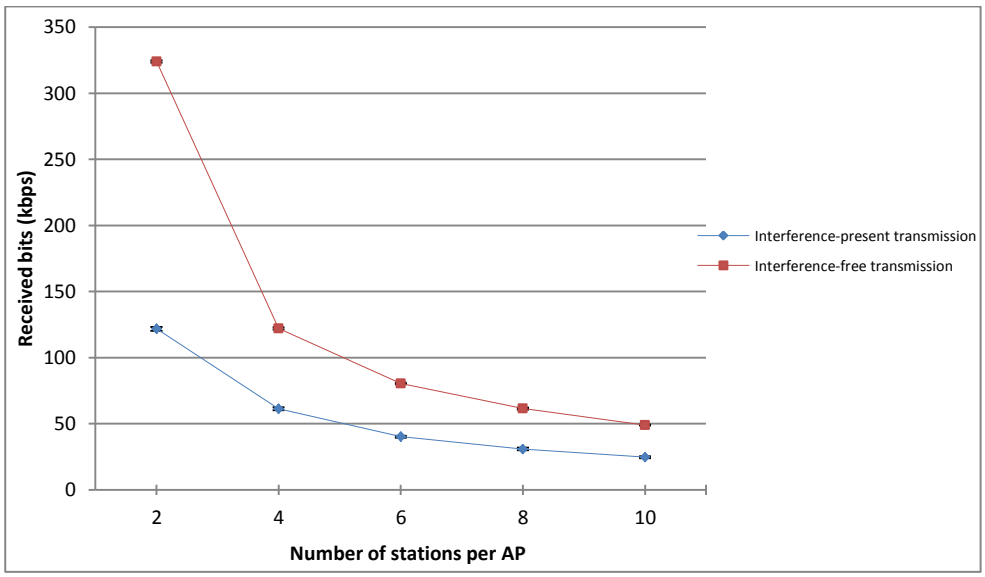


Figure 26. Interference-free vs interference-present transmission- varying packet size

4.2 Virtual TDMA

Next, the virtual TDMA method is simulated and examined, having one AP with varying number of stations. To eliminate the contention between the stations, the stations are assigned virtual timeslots where they can use the channel undisturbed. During each timeslot a station transmits UDP packets of size 2000 bytes continuously to the AP, at a data rate of 1 Mbps. The guard time between each timeslot is set to 17 ms based in the following calculations; the time it takes to send one UDP packet of size 2000 bytes at a rate of 1 Mbps is $\frac{2000*8}{1*10^6} = 0.016$ s. To compensate for any propagation delays, a margin of 0.001 s is added, and this results in 17 ms. Recall from Section 3.3.3.3 that the STATIONS frame is sent between AP and stations for timeslot scheduling. Since in the simulation, only one AP is used, the previous frame exchanges (step 1-4) of the virtual TDMA function are neglected. As mentioned in Section 3.3.3.3, the size of the STATIONS frame is calculated to 3058 bits, assuming maximum size of associated stations (99 stations). In the simulations, the number of associated stations used is 2, 4, 6, 8 and 10, resulting in an overhead of 0.067, 0.070, 0.073, 0.076 and 0.079 kbps respectively if assuming that the STATIONS frame is sent periodically every 10 seconds. This overhead is therefore subtracted from the final results of the virtual TDMA function. For each station density, different timeslot lengths are simulated in order to analyze how the performance changes. In Figure 26, the average received bits per second from one station after a simulation time of 100 seconds is presented.

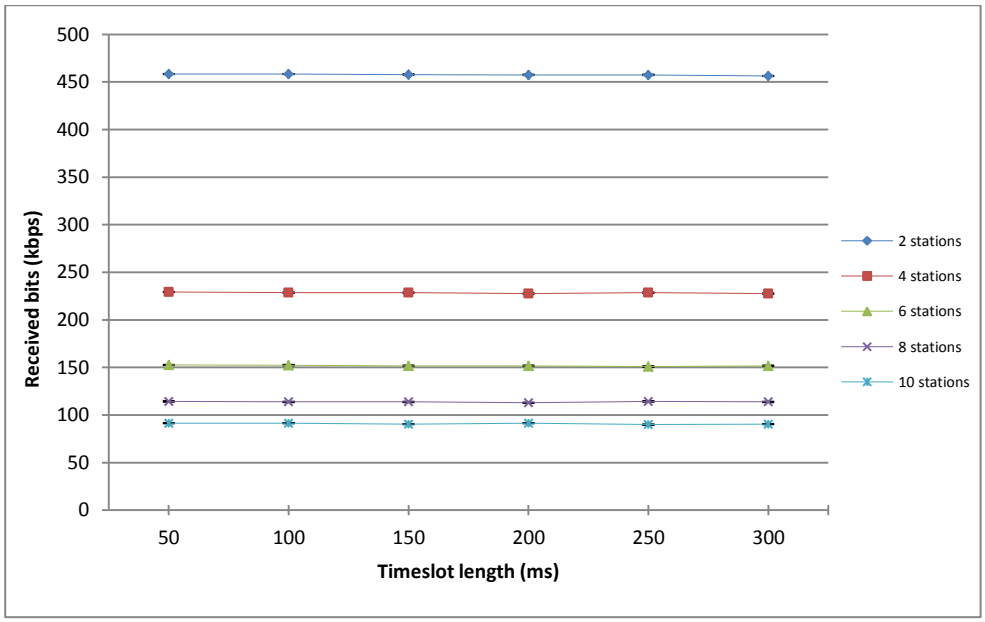


Figure 27. Virtual TDMA between stations

As expected, the more stations in the system, the lower throughput one station reaches because of the longer interruptions between transmissions, and this is seen by looking at the curves for each station density, with lowest station density at the top and highest station density at the bottom. According to the figure, the timeslot length is not affecting the throughput for one station. It can therefore be concluded that the throughput is independent of the timeslot length. However, long waiting time, i.e. long timeslot length, is not good for delay sensitive services due to the increase in interupptions. Because of the scheduled transmission, the confidence interval is kept very small. The interesting part is now to compare the results of the virtual TDMA with regular contention between stations. The same simulation is run for the different station densities, but without any scheduling, and the result is presented in Figure 27.

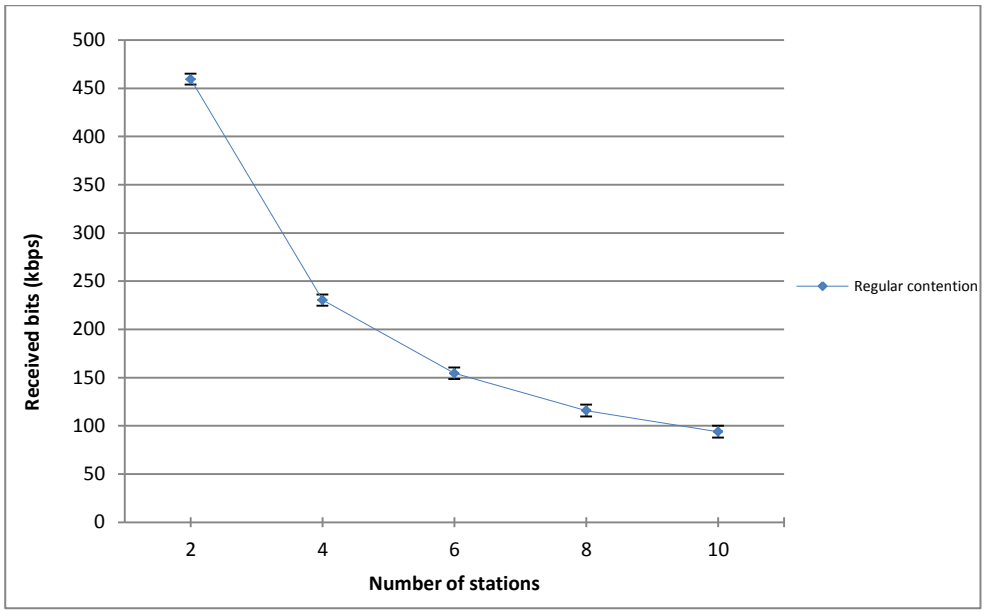


Figure 28. Regular contention between stations

Once again, with more stations in the system, the lower throughput one station reaches due to the contention between the stations. The confidence interval is higher in this case, compared to the previous virtual TDMA simulations. To get a better overview and for easier comparison, Table 4 summarizes the results of the virtual TDMA simulation (Figure 26) and the contention simulation (Figure 27) with the confidence interval for each station density also included. Since the throughput is similar for all timeslot lengths, the one giving highest results is chosen for the comparison, and that is timeslot length 50 ms.

Table 4. Summary of the virtual TDMA and regular contention simulations

Station density	Virtual TDMA (kbps)	Contention (kbps)
2	458.9 ± 0.1	459.5 ± 5.6
4	229.7 ± 0.1	230.4 ± 5.8
6	153.3 ± 0.1	154.5 ± 5.9
8	114.9 ± 0.1	115.8 ± 6.0
10	92.0 ± 0.1	93.8 ± 6.1

From the table it can be seen that the mean results for all station densities are similar for virtual TDMA and contention, but the confidence interval is higher in the contention case. This means that the throughput for a station can vary more with contention as medium access, compared to having scheduled medium access between stations. So for one simulation run, one station could have a higher throughput than the other stations, but in next run the same station could get lower throughput than the others. Whereas in the virtual TDMA case, the same station shows a stable throughput for any simulation, hence the low confidence interval. The conclusion that can be drawn from this is that virtual TDMA provides a more stable throughput and fair distribution of medium access among the stations.

Now it is investigated how the virtual TDMA and scheduled stations are affected by non-scheduled stations i.e. contending stations. In the first scenario, there is one AP and four stations; station 1, station 2, station 3 and station 4. Station 1, 2 and 3 are assigned virtual timeslots of 0.1 s each, while station 4 is assigned the same timeslot as station 1, i.e. station 4 contends for medium access with station 1 in regular CSMA/CA fashion. All of the stations transmit UDP packets of size 2000 bytes continously during their allotted timeslot to their AP, at a rate of 1 Mbps. In the second scenario, station 1, 2 and 3 are assigned timeslots of 0.1 s, and station 4 contends for medium access in any of the three timeslots whenever

the channel is sensed free. In the third scenario, all of the stations are assigned virtual timeslots of 0.1 s each. The guard interval is kept to 17 ms in all scenarios. It is now interesting to investigate how station 1 is affected in all of the above mentioned scenarios. Table 5 summarizes the results of the different scenarios, where the average received bits per second from station 1 is presented after a simulation time of 100 s.

Table 5. Summary of virtual TDMA combined with contention – 1 AP

Scenario	Throughput (kbps)
Station 1 and 4 contending under same timeslot	154.7
Station 4 contending under any of the timeslots	197.2
All four stations assigned timeslots	229.2

As can be seen from the table, the first scenario gives the worst performance, where station 4 contends for medium access only during one timeslot, specifically the same timeslot as station 1. When station 4 contends with all the other stations, the throughput of station 1 is not affected in the same way. This is reasonable since station 1 now is less interfered by station 4. If all the stations are assigned timeslots so that no contention is present, the amount of average received data bits per second from station 1 is increased. Station 4 is no longer interfering with the scheduled stations which results in undisturbed transmission for all stations.

4.3 Switch-off method

In the next case the switch-off method is simulated to see if it improves the performance. The number of associated stations varies for each simulation. The APs transmit UDP packets of randomly selected size between 1 and 2000 bytes at 1 Mbps to their stations. To test the switch-off method, one of the APs is shut down, and the left stations are handed over to the other AP. So, for example, if both APs have 10 stations each before the switch-off, one AP will have 20 stations after. From Section 3.3.3.4, the UPDATE-INFORMATION frame is used by the coordinator-AP to determine if switch-off is needed based on the interference level sent from the cluster members, and the overhead for this frame is 754 bits. If assuming a periodic transmission of this frame every 10 seconds, and furthermore including SWITCH-OFF and CONTROL frame once, it results in a total overhead of $\frac{8634}{100} = 0.086$ kbps for a simulation time of 100 seconds. The final result after the switch-off includes the calculated overhead. The total average received bits per second before and after the switch-off is presented in Figure 29.

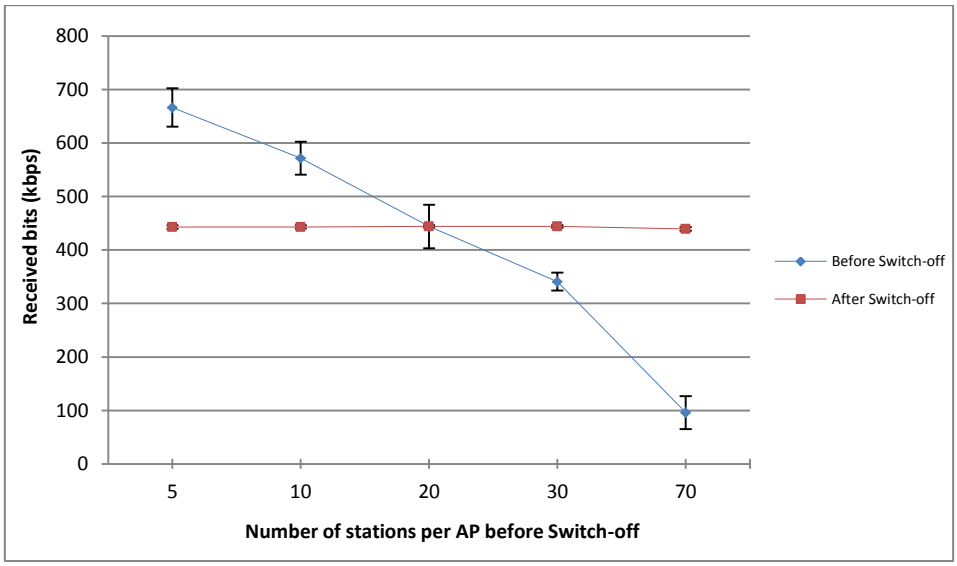


Figure 29. Switch-off method

As can be seen in the figure, the switch-off method shows no improvement in the overall throughput for station densities below approximately 20 stations. The APs seems to handle the co-channel interference fairly well with low station densities, and the overall throughput is better before switching off one AP. When the station density increases, the co-channel interference starts degrading the overall throughput due to heavy contention and packet loss. With 70 stations connected to each AP, the overall received bits per second reaches 100 kbps before the switch-off method is executed, and nearly 450 kbps after. The results show that it is better for one AP to handle a large number of stations e.g. 140 stations, compared to having two APs with 70 stations each with co-channel interference present. Therefore, it is concluded that it is more beneficial to use the switch-off method for higher station densities. Based on the traffic load and interference level information from cluster members, the coordinator-AP could send out `SWITCH_OFF` frames to improve the performance, and this is especially appropriate when the station density and the interference level is high. Depending on the station density, one could switch between the

switch-off method and virtual TDMA i.e. when the station density is low, it could be better to use virtual TDMA. On the other hand, if the density is high, it is preferable to use the switch-off method.

4.4 Load balancing

Load balancing is tested in this section. Two APs operating on different channels are deployed with a distance of 5 m from each other. The number of associated stations varies for each simulation. The stations send UDP packets of size 2000 bytes to their AP, and the APs do not interfere with each other since they are running on different channels. To execute the load balancing method, some of the stations from the more heavily loaded AP are handed over to the less loaded AP to create a more balanced station distribution. For example, if one AP has 10 stations and the other one has 30 stations, 10 stations are handed over and both APs will have 20 stations after the load balancing method is performed. Recall from Section 3.3.3.6 that in order to perform the load balancing method, the `UPDATE-INFORMATION` frame is shared to the coordinator-AP, along with the `MANAGEMENT` and `LOAD_BALANCING` frame. Assuming that the `UPDATE-INFORMATION` and `MANAGEMENT` frames are periodically shared every 10 seconds, and `LOAD_BALANCING` shared once, the overhead per second is $\frac{16378}{100} = 0.310\,\text{kbps}$. The final result after the load balancing method is performed includes this overhead. Figure 29 illustrates the throughput before and after the load balancing, measured in total average received bits per second for one station in each AP network.

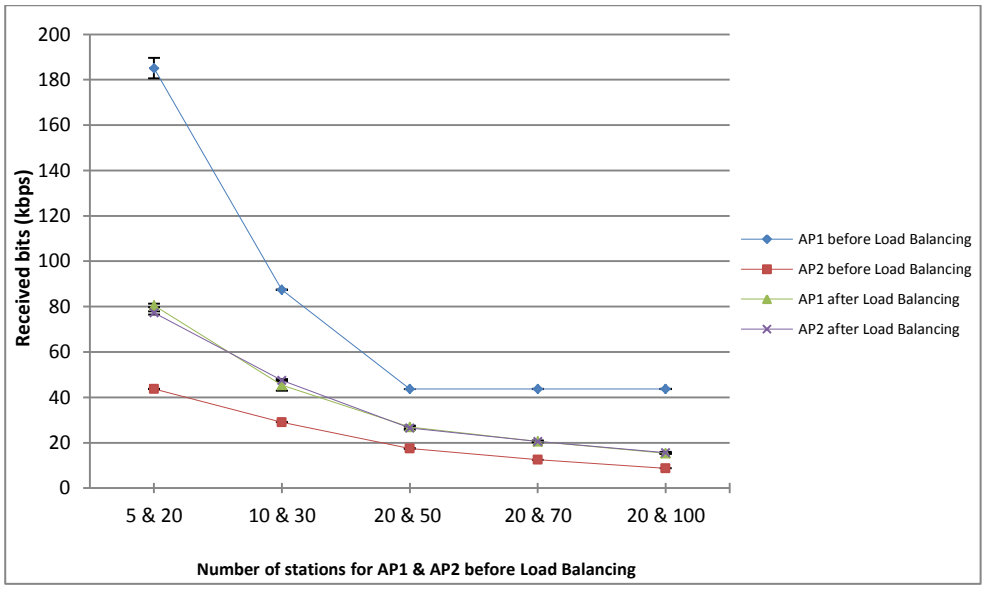


Figure 30. Load balancing method

The figure shows that the load balancing function results in a more balanced throughput among the stations connected to the different APs. So when an AP is overloaded, and to increase the fairness in nearby networks, handing over some stations is preferred. This is why it is suitable that all APs in an area always keep track on the traffic load around them, so that they can try to distribute the stations in a more balanced fashion. As mentioned earlier in Chapter 3, the average load extracted from the `MANAGEMENT` frame from the coordinator-AP is used to determine if the load balancing is needed. Making the APs well aware of the load around them is therefore of importance, and the `MANAGEMENT` frame could be shared in intervals short enough for the APs to evaluate the need of load balancing. In the same way load balancing could be implemented when neighbouring APs operate on the same channel, and it could then be combined with virtual TDMA to get the best results.

CHAPTER 5

5 Conclusion

In this thesis we designed a text-based protocol for inter-access point communication. The protocol is intended to give better network performance for all involved parties. It consists of a number of functions; channel assignment, virtual TDMA, load balancing and switching off APs. New frames were proposed and constructed for the protocol; `COORDINATOR_ELECTION`, `CONTROL`, `INFORMATION`, `MANAGEMENT`, `VIRTUAL_TDMA`, `SWITCH_OFF`, `SWITCH_ON` and `LOAD_BALANCING`. The coordinator-AP is introduced and shares important data among neighbouring APs with the `MANAGEMENT` frame. The data in the `MANAGEMENT` frame is based on the `INFORMATION` frames that are collected by the coordinator-AP from the cluster members. The `MANAGEMENT` frame includes parameters that are later used by the APs for the different protocol functions, e.g. channel assignment and load balancing. Flow charts have been presented to give an intuitive demonstration of the actions performed when an AP receives a frame. Simulations in ns-3 were performed to evaluate the functions of the protocol. The performance of regular contention between APs begins to suffer significantly when the number of stations increases, hence the proposed channel assignment function. It was concluded that virtual TDMA stabilizes throughput for stations, and provides a fair distribution of medium access among them, compared to having regular contention. It is also concluded that the troughput is independent of the timeslot length. Combining virtual TDMA-scheduled stations with non-scheduled stations degrades the performance for stations, therefore it was concluded that all stations should be included in a virtual

TDMA network. Simulations showed that the switch-off method only pays off when the station density is large. Then it is better to switch off one AP and hand over the stations to adjacent APs, since the co-channel interference with a large number of stations greatly degrades the network throughput. The simulations also showed that balancing the distribution of stations i.e. traffic load between APs results in a more balanced throughput among the stations connected to different APs with. All simulation results included the protocol frame overhead, and it was concluded that even with this overhead, the functions still show improvement in terms of throughput.

The number of Internet-users is still increasing, as well as the popularity of WLANs. Moreover, the user-demand is also getting higher, while the bandwidth remains a scarce resource. Making APs self-organized and cooperative, and at the same time using the proposed protocol functions, is an interesting field of research, and could be a step towards better network performance, especially in uncoordinated deployments.

CHAPTER 6

6 Possible Improvements and Future Work

The next step would be to continue the simulations but with TCP traffic instead of UDP traffic. With TCP, the same simulations could show different results due to retransmissions, ACK, SYN etc. Therefore it is of interest to simulate and evaluate the performance using TCP.

An interesting case to study and simulate is to assign the same timeslot to more than one station when the station density is high. Then the stations contend for medium access only during this timeslot and they do not need to wait longer periods as in the case of one timeslot for each station.

Designing the protocol for the ad-hoc mode can also be something to consider in the future. Having different IBSS in the same area communicating with each other and exchanging information could also result in better performance. The functions proposed in this paper would need to be adapted to the ad-hoc mode, and new functions could be added. The same goes for infrastructure mode. Adding new functions or improving the already proposed functions could be investigated.

More experienced ns-3 users could implement the whole protocol with all functions and simulate it in order to get a complete result and better overall picture of the protocol.

References

[1] J. Farooq and B. Rauf, "An Overview of Wireless LAN Standards IEEE 802.11 and IEEE 802.11e," *Department of Computing Science,* Umeå University, 2006.

[2] J. Schiller, *Mobile Communications,* Addison-Wesley, 2nd Edition, 2003.

[3] M. Abusubaih, J. Gross and A. Wolisz, "An Inter-Access Point Coordination Protocol for Dynamic Channel Selection in IEEE 802.11 Wireless LANs," *Telecommunication Networks Group.*

[4] IEEE, "Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation," *IEEE Standard 802.11f,* July 2003.

[5] M. Abusubaih, J. Gross, S. Wiethoelter and A. Wolisz, "On Access Point Selection In IEEE 802.11 Wireless Local Area Networks," *Local Computer Networks, Proceedings 2006 31st IEEE Conference on,* pp. 879-886, 14-16 November 2006.

[6] ns-3, 2008, [ONLINE] Available at: http://www.nsnam.org/. [Accessed 20 March 2013].

[7] M. Ergen, "IEEE 802.11 Tutorial," *Department of Electrical Engineering and Computer Science,* University of California Berkeley, June 2002.

[8] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan and W. Arbaugh, "A Client-driven Approach for Channel Management in Wireless LANs," *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings,* pp. 1-12, April 2006

[9] M. Boulmalf, T. Aouam and H. Harroud, "Dynamic Channel Assignment in IEEE 802.11," in *Wireless Communications and Mobile*

Computing Conference, 2008. IWCMC '08. International, pp. 864-868, 6-8 August 2008.

[10] IEEE Std. 802.11, *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, The Institute of Electrical and Electronics Engineers, 29 March 2012.

[11] B. P. Crow, I. Widjaja, J. G. Kim and P. T. Sakai, "IEEE 802.11 Wireless Local Area Networks, " Communications Magazine, IEEE (Volume:35 , Issue: 9), pp. 116-126, September 1997.

[12] 802.11 frames: A starter guide to learn wireless sniffer traces - Cisco Support Community, 2013, [ONLINE] Available at: https://supportforums.cisco.com/docs/DOC-13664. [Accessed 7 May 2013].

[13] G. Camarillo, *SIP Demystified*, McGraw-Hill TELECOM, 2002.

[14] DotEleven, 2011, Chapter 4 - 802.11 Management frames. [ONLINE] Available at: http://dot11.info/index.php?title=Chapter_4_-_802.11_Management_frames. [Accessed 21 May 2013]

[15] A. Mishra, V. Shrivastava, D. Agarwal, S. Banerjee and S. Ganguly, "Distributed Channel Management in Uncoordinated Wireless Environments", *Proceeding MobiCom '06 Proceedings of the 12th annual international conference on Mobile computing and networking*, pp. 170-181, 2006.

[16] H. Luo and N. K. Shankaranarayanan, “A Distributed Dynamic Channel Allocation Technique for Throughput Improvement in a Dense WLAN Environment,” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on (Volume:5)*, pp 345-348, 17-21 May 2004.

[17] M. W. R. da Silva and J. F. de Rezende, “A Dynamic Channel Allocation Mechanism for IEEE 802.11 Networks, ” *Telecommunications Symposium, 2006 International,* pp. 225-230, 3-6 September 2006

[18] O. Brickley, S. Rea and D. Pesch, “Load Balancing for QoS Enhancement in IEEE 802.11e WLANs Using Cell Breathing Techniques,“ Centre for Adaptive Wireless Systems, Department of Electronic Engineering Cork Institute of Technology.

[19] H. Velayos, V. Aleo and G. Karlsson, “Load Balancing in Overlapping Wireless LAN Cells,“ in *Communications, 2004 IEEE International Conference on (Volume:7),* pp 3833 - 3836, 20-24 June 2004.

[20] M. Heusse, F. Rousseau, G. Berger-Sabbatel and A. Duda, “Performance Anomaly of 802.11b,” *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies (Volume:2),* pp 836-843, 30 March-3 April 2003.

LUND
UNIVERSITY

http://www.eit.lth.se