University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science
Chair for Design and Analysis of Communication Systems

Roaming on a wireless campus network

Bachelor assignment Telematics

Committee:
P.T. de Boer
A. Pras
G.J. Heijenk

I.F.R. Noppen
8/14/10

Abstract

Roaming on wireless networks is an aspect of wireless networks which is not researched extensively yet. In this report we present the results of analysis and visualization of roaming on the wireless campus of the University of Twente. The University has provided us with a large dataset (~2 months), which we could use for our analysis.

We found that there is a fairly small share of roaming taking place on the network (17.6% of the sessions), but that almost 80% of the clients roamed at least once in the total dataset. Regarding unnecessary roaming we found that we could account 31.3% of the associations to this category, which is a fairly high, but pessimistic estimate. We also developed a method to represent the roaming data in a short movie that can be interpreted without too much detailed knowledge of roaming.

Acknowledgements

First and foremost I want to thank Pieter-Tjerk de Boer for his supervision and inspiring words at times I was uninspired or needed another perspective.

Another crucial role was reserved for Jeroen van Ingen, network administrator of the ICT Service Centre of the University of Twente. He supplied all the network data and answered all my questions regarding gaps, unknown phenomena in the data and the meaning of all the columns in the trace files.

Table of Contents

1 Introduction

An ever-growing number of people use wireless networks every day. After revolutionizing the way we connect to our home and work networks, wireless networks are used more and more with mobile devices as well. People can now connect anywhere, anytime with their favourite web services, read their email and update their status on social media.

1.1 Wireless network

In June 2003 the "wireless campus" was founded at the University of Twente, enabling Internet access to staff and students anywhere on the campus without the hassle of finding wall sockets and connecting cables. (Diepenhuis, 2003) The campus of the University of Twente is about 140 hectare and is located between the cities of Enschede and Hengelo. The campus contains educational, sports and leisure buildings as well as staff- and student housing. With over 800 access points, the campus is essentially one big hotspot. Through the years, the network has been extended to include some access points in the nearby "Business and Science park" and in the city centre of Enschede. The wireless campus has also joined the eduroam initiative, which enables students from participating educational institutions to connect to each other's wireless networks with the credentials from their own universities.

Roaming

Because one access point can just service a limited area depending on obstacles which block or weaken the wireless signal, when a wireless network is developed on a large area, we need to place multiple access points to service that area. To maintain a connection when we move through the area with our mobile device, we need to connect to multiple access points sequentially. When people move with a wireless enabled device outside the coverage area of one access point, the device has to connect to another access point to be able to offer continued connectivity. Also, if an access point is connected to a failing power supply and the access point stops transmitting, all previously connected devices need to reconnect to another access point. In afore mentioned cases, the wireless device needs to find another signal. *The seamless switching to the new access point is called Roaming.*

When we turn on our laptops, the laptop begins to look for wireless networks in its vicinity. Most of the times nowadays, the laptop will find multiple networks with different names and lookup these names in its history of known networks. A single network can consist of multiple access points which are in reach (have a strong enough signal at the location) and the wireless chip will probably choose to connect to the access point with the greatest signal strength. Now, dependent on the configuration of the network, the wireless chip will try to authenticate to the network and setup a connection. This should result in a working wireless (internet) connection with afore mentioned wireless network.

As long as the signal strength and the position of the user remains the same, the user will be connected to the same access point. If we move the user however, the relative signal strengths of the access points will change. When the strength of the access point, which the user is currently connected to, drops below a certain threshold, the wireless chipset will again search for access points related to the same network. If there is an access point in the vicinity with greater signal strength than the current one, the chipset will disconnect from the current access point and negotiate a connection with the access point which has greater signal strength.

What we defined in the last paragraph is the most traditional form of roaming: roaming within a network of a single operator. Furthermore, there exists roaming between different network operators and technologies (i.e. from WLAN to 3G), but that is outside the scope of this assignment. In this paper, we focus on roaming within the wireless network on the University of Twente.

1.2 Goals

The goals of this assignment are the analysis of roaming on the wireless network at the University of Twente and the visualization of roaming.

1.2.1 Roaming analysis

Because there have not been that much earlier studies on the subject of roaming on a large wireless network, we hope to extract a few interesting facts about roaming from the dataset. A lot of people have WLAN enabled devices nowadays like laptops, PDA's and smartphones. But what we don't know is if they use that functionality at all and if they stay connected with the wireless network while wandering around the campus. So our first question is how much roaming actually occurs on the wireless network of the University of Twente.

We also suspect that there is a lot of roaming going on that could be considered as unnecessary. With unnecessary we mean that a user with a mobile device is on the edge of two or more coverage areas of multiple access points. This could possibly mean that the device is constantly switching between these access points. Therefore, we would like to know how much this "unnecessary" roaming occurs.

Another interesting aspect of wireless devices is the current uprising in the possession and use of WLAN-enabled smartphones and PDA's. It is easier than ever before to check your email, read the newspaper, post a message to twitter or Facebook whilst walking to the next lecture or whilst riding your bike around the campus. Since the portability of these devices is even higher than the relatively stationary laptops and notebooks that are usually switched off when they are carried around, we suspect that the amount of roaming is increasing because of this development. Therefore, we would like to know if the amount of roaming is actually increasing over time.

1.2.2 Roaming visualization

Nobody really knows when, where and how much roaming is taking place on the campus. All these facts could be represented in graphs, figures and data, but these can only tell us about the amount of roaming and the time at which the roaming occurs. What cannot be represented in a simple and insightful way through graphs, figures and data, is where the roaming is taking place on the campus. Furthermore, studies suggest that people are better at pattern recognition with visualizations than they are with a list of data. Therefore, we asked ourselves how we could visualize the roaming paths on the wireless network of the University of Twente in a concise, insightful way. From this visualization we would like to be able to see all three aspects (when, where and how much), while preserving the ability to interpret the visualization without detailed knowledge of roaming.

1.3 Approach

To be able to analyse and visualize data about the wireless network of the University of Twente, we first needed a dataset. Fortunately the ICT service centre of the University keeps their log files for a few months. Therefore it was no problem to acquire this data. With this data at hand, we were able

to analyse it and prepare it for visualization. To provide ourselves an easy way to query the data we converted the log files to tables in a MySQL database.

1.4 Structure

This report will begin at base level with the description and selection of the available measurement data and build on that to explain our analysis of the data and the difficulties we had with the data. A big part of the analysis of the data comprises of the pre-processing of the data to transform it into suitable data for visualization. After analysis of the data comes the visualization part with the problems we walked into there. Finally this report will conclude with the conclusions and some recommendations for future research.

2 Background

2.1 Available measurement data

The ICT service centre of the University of Twente supplied a variety of data about the wireless network. This data comes from all kinds of logging facilities and all encompass approximately the same window of August 6 to October 15 2009.

SNMP access point data

The SNMP access point data is a log file in which results from SNMP read-outs are combined. For this log file, a SNMP management server requests SNMP data every three minutes from each access point. This data includes the client mac-address, ip of the access point, timestamp, uptime of the client, interface to which the client is connected and other fields which are not interesting in this assignment.

Furthermore, another log file shows the more access point related statistics (number of associations, name of access point, timestamp of read-out, interface, ip address of the access point), which are more suitable in a research about access point usage.

Accounting data

The RADIUS accounting data log file contains everything the RADIUS server receives about accounting on the wireless network. Information about accounting sessions can be found in this log file. This log file is actually quite useful since every access point authenticates every client when it begins an association with this client. RADIUS will be explained later in section 2.2.

Access point data

This data file contains a list of all the access points in use for the wireless network of the University of Twente, combined with information about each of the access points. This information contains descriptions, signal strength, mac address, ip, geographical location in “Rijksdriehoekstelsel” (RD) etc. Unfortunately not all information is available for every access point. The names of the access points are relatively descriptive though and it’s possible to find out the approximate whereabouts of the access point based on the name.

2.2 Authorization, authentication and accounting

Wireless networks are prone to unauthorized and possibly illegal use because of their very nature. You don’t have to find a wall socket and connect a cable anymore to try to get access to the network.

Wireless networks are opportunities for criminals to attack other computers from a location that is not traceable to them or can be used to hack into the systems of the provider of the wireless network if the network is not properly secured. To prevent usage of the wireless network by unauthorized users, the University of Twente uses 802.1X and RADIUS for authorization, authentication and accounting. IEEE 802.1X is a standard for network access control. The protocol provides an authentication method to devices that want to connect to a wired- or wireless network. RADIUS is a protocol, which provides centralized authentication, authorization and account management. It was developed in 1991 and submitted to the IETF in April 1994.

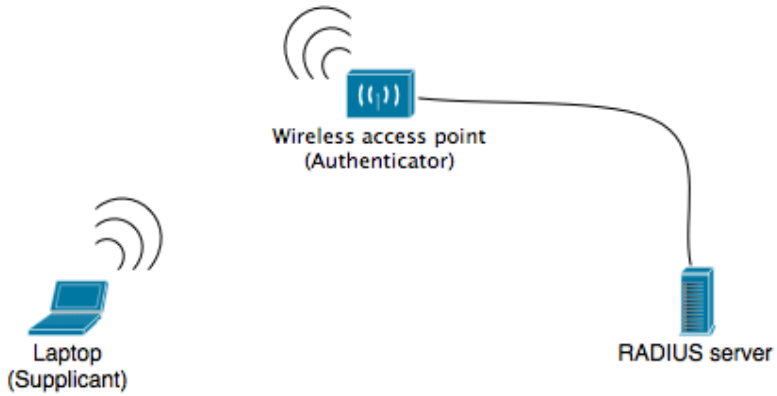

Figure 2.1 Diagram of authentication with a RADIUS server through a wireless access point

Authentication with 802.1X involves three parties: a supplicant, an authenticator and an authentication server. With the supplicant, we mean a device (laptop, pda, smartphone) wishing to join the network. The term supplicant however, is also widely used as the piece of software, which supplies credentials to the authenticator. The authenticator is a network device. In the scope of this paper, this is always a wireless access point. The authentication server is generally a server running RADIUS and/or other authentication protocols. The University of Twente is no exception in this. The authenticator acts like a security guard; it protects the inner network against unauthorized supplicants. With 802.1X, the supplicant provides credentials to the authenticator (username, password and/or digital certificate). The authenticator forwards these credentials to the authentication server for verification and allows the supplicant access if the authentication server validates the credentials. Schematically, this is depicted in Figure 2.1.

When a supplicant wants to connect to the network, the authenticator sends an Access-Request message to the RADIUS server containing the credentials, which the supplicant entered. The RADIUS server now returns one of three responses back to the authenticator: Access-Reject, Access-Challenge or Access-Accept. When an Access-Reject is received, the credentials failed validation and the authenticator should deny access to the supplicant. When an Access-Accept response is received, the authenticator should allow access to the supplicant. If however an Access-Challenge response is received, the authenticator has to request additional information from the supplicant such as a secondary password, token or card.

When network access is granted to a supplicant by the authenticator, the accounting phase begins. This phase can yield three different status messages, which describe the current state of the accounting session. These three types of messages and when they are sent are outlined below.

- An Accounting-Request with an Acct-Status-Type attribute with the value “start” is sent by the authenticator to the RADIUS server to signal the start of the supplicant’s network access. These “start” records typically contain user identification, network address and data about the wireless access point as well as a unique accounting session identifier.
- Periodically, “keep-alive” messages are sent when the supplicant is still attached to the access point. These messages typically contain data like number of packets / bytes sent and received and session duration.
- When the connection to the supplicant is lost or terminated, the authenticator sends a “stop” message to the RADIUS server providing final information on usage like time, data transfer and the reason for the disconnection.

In general, the primary purpose of accounting is for billing the user according to usage or time connected, but in the wireless network of the University of Twente, this data is mainly used for statistical purposes and network monitoring. The full process for an association with an access point with RADIUS authorization and accounting is depicted in Figure 2.2.

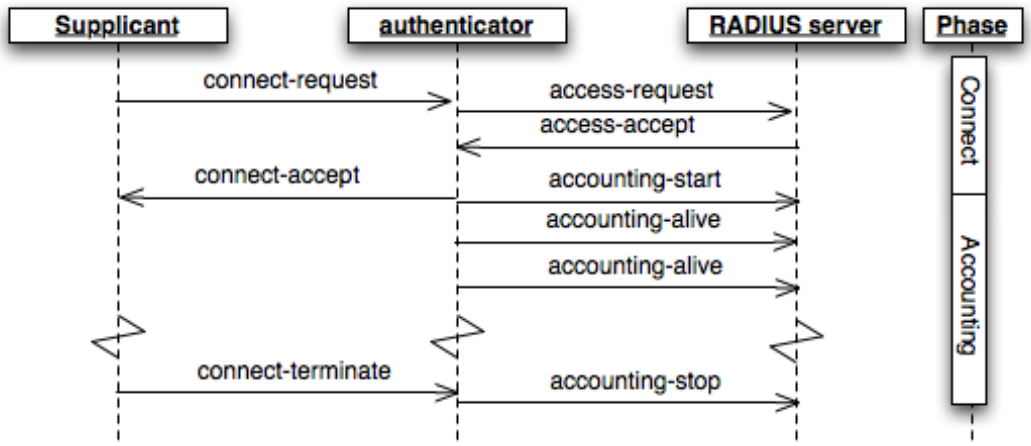

Figure 2.2 Exchange of messages between a supplicant, authenticator and RADIUS server when the credentials are valid and termination happens from the supplicant. Alongside with the naming of the phases.

As we have explained, the exchange of messages between the three parties present in Figure 2.2 happens when a client wants to connect to the wireless network. This is true for each association the client negotiates with an access point. So when a client roams from one access point to another, this whole sequence of messages between the three parties will be repeated.

The accounting data we discussed before in section 2.1 contains all the data collected by the RADIUS server in the accounting phase. This data therefore consists of “start”, “alive” and “stop” records. Additional data recorded in this dataset contains the timestamp, the accounting delay (the delay in seconds before the message could be sent to the radius server), the mac address of the client, the mac address and IP address of the access point and other fields which are of no interest for the scope of this assignment.

3 Roaming analysis

3.1 Roaming data selection and preparation

To be able to analyse data about roaming actions on the wireless network of the University of Twente, we need data that provides a reliable picture of these actions.

3.1.1 SNMP vs. RADIUS data

Both the SNMP access point data and the RADIUS accounting data can provide us with an overview of the clients which are connected to an access point. The SNMP data is polled by a management server every three minutes and gives us details about the clients which are active at the time of polling. The RADIUS accounting data comes from the RADIUS server. This log file encompasses all data about the accounting of the associations on the access points.

The SNMP data, has a window of three minutes in which a management server queries for active clients on each access point. Therefore, this data is basically just a snapshot of the clients connected to the access points at the time of polling by the management server. This implies that it is very well possible that the usage of this data misses clients which are connected to a given access point just between two polling intervals. Furthermore, the exact time of a roaming action could not be determined within these three minutes intervals with this data. Since we would like to analyse the whole spectrum of roaming at the campus if possible, and as accurately as possible, we weren't able to use this data source.

The "Accounting data" from the RADIUS server contains, as we explained before, start, stop and keep-alive records. For every client, an accounting session is started when he has been authenticated by the access point. When the client now roams to another access point, this access point authenticates the client and a new accounting session starts. Amongst others, the RADIUS server stores the client mac address, the access point mac address, the timestamp of the message, the delay of the message before it was sent to the RADIUS server and the status message. Combining the data from different records leads to the data we are trying to assemble without any gaps like the SNMP data. Therefore, the RADIUS accounting log file contains exactly the data we are looking for.

3.1.2 Roaming actions vs. associations

The RADIUS accounting data supplied by the University all encompass data about which user is associated with which access point at a given time. This cannot be directly extracted from the data though without some pre-processing. If you want to know which client is active at a given access point at a given time you would have to sift through the records backwards in time starting at that timestamp to look for alive and start records of active clients. Because this is a slow and error prone method, we need to pre-process the data. To be able to pre-process this data into something we could easily work with for our purpose, there are mainly two different approaches. The first approach is to collect only the roaming actions from one access point to another, resulting in one record in a table per roaming action. This approach however has some clear problems:

- We lose all data about the length of a session, as the first action we record is the roaming from one to another access point and the last action is the transition to the last access point. The client could easily spend a lot of time on this last and/or first access point, giving a false impression of session length.

- It becomes hard to associate a roaming transaction with a session. This is mainly because the records themselves represent the roaming actions, which can occur at any discrete point in time.

Since we would like to be able to associate roaming actions to sessions, this way of representing the data is too low level for the scope of this assignment.

The second option is to pre-process the data we have into one record per association with an access point. Now we can store start- and end times of an association with an access point resolving the problem about session length, since we can associate access point associations with a session. Furthermore, we could quite easily extract the roaming actions from this pre-processed data. Because if you take the data from one client, between every record there is a potential roaming action if the start time of the latter record is the same as the end time of the first record. Because the second approach certainly has more potential than the first, we continued with the second approach.

3.1.3 Non-existing mac-addresses

After reviewing the dataset, we found a problem in the dataset: mac addresses of access points that did not exist in the access point dataset were found in the accounting dataset. Further research of this problem led us to the guest network of the University of Twente. When a guest wants access to the wireless network, he can login through the guest-network, after which he has to enter credentials of an existing account (for instance the account of a colleague) at the guest network landing page. Only after entering these credentials, Internet access is granted to this client until logging out. In this case, when account details are entered and validated, another accounting session is started from the server on which was just logged in. This explains a few of the unknown mac addresses for they are clearly no wireless access points and for that reason not present in the access point dataset. Therefore, we decided to only pre-process data from access points known to us by means of the access point dataset.

3.1.4 Pre-processing

The process of preparation of the RADIUS accounting data into a table containing a record for each association between an access point and a client was fairly straightforward. After loading the raw accounting data into a MySQL database and converting the verbose English timestamps to Unix timestamps, we ordered the data by this timestamp. Now, by iterating over this data, we can add the associations with an access point to our new table. The data that is important for our resulting table is the mac address of the client, mac address of the access point, start time of the association and end time of the association. To pre-process the data we iterated over the records in the accounting data. While iterating over the records in the data, we keep track of the associations currently going on in the dataset by storing these sessions in a map. This map holds objects representing the current associations, containing client mac address, access point mac address, start time and end time. For every combination of access point and client there is only one entry in this map, because you can't associate twice with an access point at any given point in time. When an association ends it is saved to the database and removed from the map. The associations, which are still present in the map when all rows have been processed, are also saved to the database. The process is described in the following algorithm:

- For each record in the accounting data execute the following steps.
    - Get the timestamp from the record (with the delay subtracted from it)

    - Does an entry exist in our current associations map for the client and access point contained in this record?
        - Yes: Is the accounting status type “Stop”?
            - Yes: Update the end time of this association in the current associations map and add this association to the new table. Then delete the association from the current sessions map.
            - No: Update the end time of this session in the current associations map.
        - No: add this association to the current associations map and set the start and end time of this association.
- End of iteration over the accounting data.
- For each record in the current associations map execute the following steps.
    - Add this association to the new table.
- End of iteration over the current associations map.

The last iteration step in the algorithm is needed to include all potentially not finished associations, which were still going on at the end of our dataset. Analogue to this, we don’t check for “Start” records as well in the data since a session could begin with an “Alive” record if the session begun before the first timestamp in the dataset. In the latter case, we considered the first “Alive” record as a “Start” record. Another way of interpretation would have been to consider the start to be at the beginning of the dataset or to disregard this whole association. We opted for this solution however, because the exact times of begin and end of the sessions don’t matter in an assignment about roaming. If we had needed to analyse throughput or user behaviour with regards to the time of activity, this would have had more value.

3.2 Defining sessions

When we explained roaming in the introduction of this paper, we wrote about a user which hops from access point to access point because the signal strength of the former access point declined until under a certain threshold. Every time a user switches access points, in the way the network is setup at the University of Twente, the new access point will negotiate a new accounting session with the RADIUS server and the old accounting session will be stopped by the former access point. This means that our data consists of records about associations with access points for each user. To be able to extract roaming from this data however, *we need to group these associations into something we call a session, which we could use to identify roaming*. Enter the session, which can be described as all wireless activity occurring from turning on your wireless apparatus to turning it off. Unfortunately, this description is not accurate yet because it is very well possible that someone wanders briefly into a spot without wireless reception or puts their apparatus to sleep. Furthermore, our data does not indicate if the client has just turned on or turned off its apparatus. *So we defined the concept of session being a group of associations to access points, which follow each other without long interruptions.*

This definition however, is a vague definition. The term “long interruptions” is something, which needs to be researched and fine-tuned to an appropriate value so that we do retain most of the roaming but don’t unnecessarily see access point switches, which are not related to roaming as a roaming action. In other words we are seeking for the maximum length of the gap between associations with access points from the same user for which we will call the transition a roaming action. This threshold will be referred to further as the *session-timeout*. It should be noted we are

not interested in reflecting the session from laptop on until laptop off. This would be impossible because of the problems mentioned above with interruptions of connectivity. In this assignment, we are interested in the sessions in which roaming is identifiable. That is, we should be able to say with probability bordering on certainty that it was a roaming action and not a new session that we identified. So there should not be too much roaming included which is actually not roaming at all.

While there are studies that reason with what the client perceives as an uninterrupted session and therefore use session-timeouts as high as 30 seconds (Kotz & Essien, 2005), we will try to establish a reasonable session-timeout from the characteristics of- and the data in our dataset. Furthermore a longer session-timeout is often used because the source of the data is SNMP, which – as explained before – causes gaps in the measurement data.

A logical value of the session-timeout would be zero. This implies that we will only consider a transition from one to the other access point roaming if the first association ends at the same time the second association starts. The timestamp data in our dataset however, are UNIX timestamp. Since the UNIX timestamp measures the number of seconds since epoch, we are dealing with a granularity of one second. This implies that it is perfectly possible that although two associations were originally just milliseconds apart, our dataset could imply a one second delay between them. With a session-timeout value of zero, these associations would be grouped in two different sessions although the client perceived it as one roaming action and thus one session. Because of this, we can say at this point that the session-timeout of zero is not the right one to choose.

The RADIUS data actually includes a delay value (the delay in seconds before the status message was sent by the access point) and the timestamp of arrival at the RADIUS server. So, because we know the time of arrival at the RADIUS server which is measured from one clock and we know the delay in seconds before the message got sent out from the access point, we know that the timestamps of the messages are fairly correct. This also means that the gaps between two consecutive associations of longer than one second will probably have nothing to do with the granularity of the timestamp and will be disconnects and reconnects with another access point. To check this, we constructed a graph of gaps between consecutive associations with two access points for the whole dataset. If our assumption is right, there will be a peak at gaps of one second, or at gaps of zero seconds. This assumption is verified by Figure 3.1.

To compile a graph of the gaps occurring on the wireless network of the University of Twente, we setup an experiment which calculated all the gaps between two associations with an access points of every user in the dataset. To execute this experiment, we sorted the database on client, start time in ascending order. The sorting on client makes sure we can easily process all records from a single client in batch. This also ensures we don't accidentally measure the gap between an association of client A and an association of client B. To temporarily store the number of occurrences of a certain gap value (length in seconds) we constructed a map which maps from gap value to the number of occurrences in the dataset. The complete algorithm for analysis of the gap value is described below.

- Initialize the variables lastClient and lastEndTime.
- Initialize the map of gaps to occurrences named gapMap.
- Sort the records of associations by Client, start time ascending
- For each record:
    - If the client in the record is the same as lastClient

        - Calculate the gap: starttime (current record) minus lastEndtime
        - Increment the value at the gapMap on the index gap
    - Save the endtime of the current record to lastEndtime
    - Save the client of the current record to lastClient
- End of iteration
- Save the keys and values of the gapMap to a textfile for representation in GNUPlot

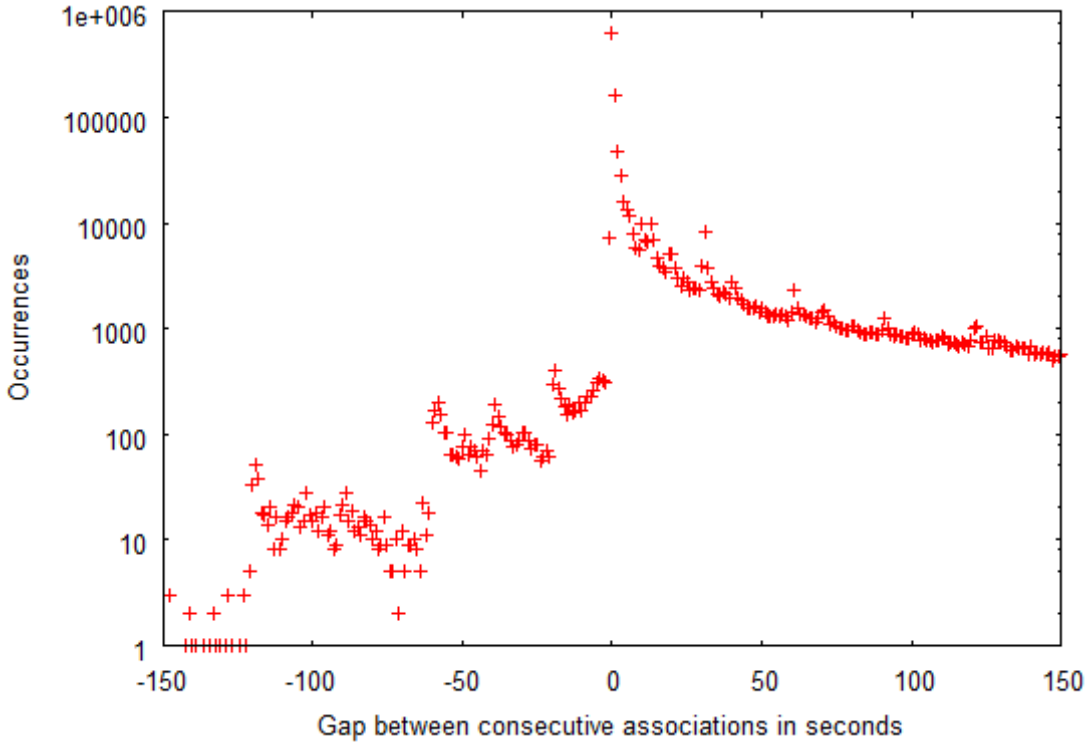

Figure 3.1 Gaps between two consecutive associations with access points (seconds)

In the graph (Figure 3.1) we can also observe an unexpected result: negative gaps. These negative gaps range from -1 to -24605 seconds since the last session. This is very odd since a negative gap would mean that a session started before the last one ended. After some further research we found out that there are roughly three cases in which these negative gaps occur.

- A long association with access point AP1 starts on exactly the same time as another, very short association with another access point (AP2). (Figure 3.2a)
- An association with an access point (AP2) starts within 120 seconds from termination of another association on another access point (AP1). (Figure 3.2b)
- A whole structure of associations (AP2-AP4) happens during an association with another access point (AP1). (Figure 3.2c)

A large part of the negative gaps is caused by the situation depicted in Figure 3.2a. This situation can be explained by the same reason we constructed earlier for the session-timeout of one second. If there is a very short association, for which there is milliseconds delay, it can appear as though it started at the same time as a longer session while the shorter association did in fact start earlier. Therefore, this is a valid structure of associations. Actually if we change the order of these two

associations, we observe only a gap with the length of the shorter association. This is also likely the way the associations occurred in reality.

In Figure 3.2b, we observe another cause of negative gaps. In this situation, the association with one access point (AP1) goes on for maximum 120 seconds while an association with another access point (AP2) has already been established. This situation can be explained because it is possible that a timeout occurs at the RADIUS server when an access point does not send an alive or termination status message. The RADIUS server will now terminate the accounting session after a certain configured timeout. Reasons for not sending an alive or termination message could be power failure, loss of wired network connectivity etc. at the access point.

The last situation (Figure 3.2c) is until now unexplained. This situation however, is very rare and if found, the underlying structure of other associations with access points is most of the time longer than the first association. Therefore it looks very similar to a normal structure of associations and can be regarded as one.

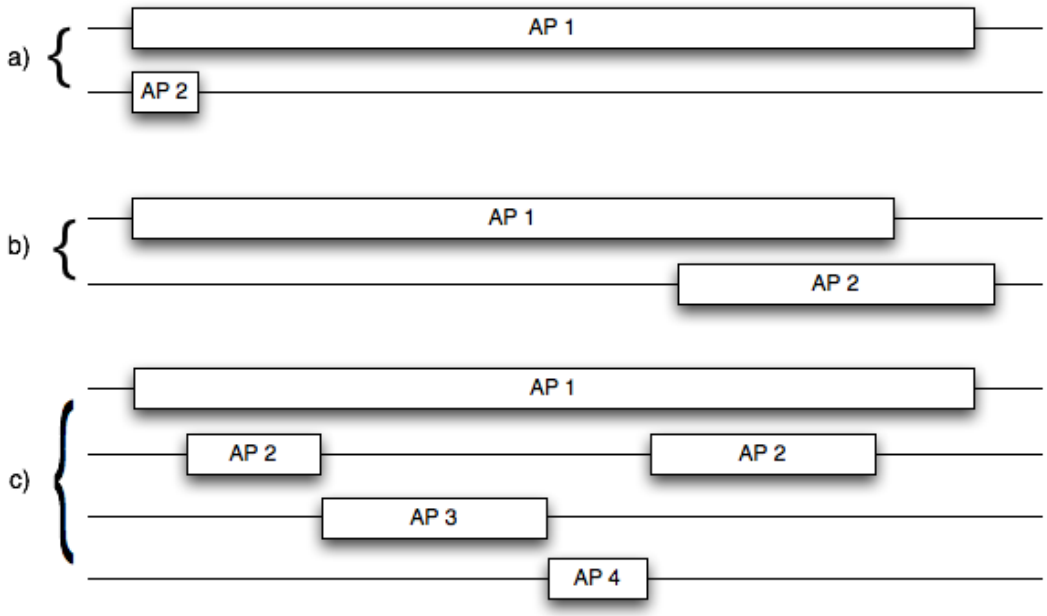

Figure 3.2 Associations with accesspoints (AP) occurring nested causing negative gaps

When we look at the implications of the situations with negative gaps in our dataset, we can clearly see that there is no negative implication for analysis about how the roaming took place. We can still clearly detect each roaming action from one access point to another. Therefore we decided not to evict these associations from our dataset.

Concluding, after analysis of the graph with gaps we finally chose a one second session-timeout as the session-timeout in which a transition from one to the other access point is still considered of the same session. Although a session-timeout of two seconds is also a fitting choice for a session-timeout, the number of transitions in which a gap of two seconds appears, is a lot less than the cases where a one second gap appears. Now, for the negative gaps we chose to ignore these cases and just regard the associations as a session in the order in which they appeared in the dataset. Therefore we chose one second as the session-timeout value.

3.3 Analysis

3.3.1 General figures

After compiling the RADIUS data into a table with associations, our associations table contains the data for 1,758,513 associations. This translates to 953,484 sessions with our algorithm and the timeout we defined for defining sessions. 167,615 of these sessions actually included roaming which brings the percentage of sessions including roaming to 17.6%. This is not a very low number, and indicates that there is actually quite some roaming going on. Furthermore, just 21.8% of the users never roamed in any of their sessions.

The average length of a session is 42:38 minutes, which is (possibly coincidentally) near the length of half an average lecture before the break begins. If we take the average length of all sessions which include roaming we find a longer length. This can be explained by the fact that roaming actually prolongs the time your session lives further when you're (almost) out of reach of the previous access point.

The average number of sessions per client in our dataset is just above 95. With only 70 days of data in our dataset this means that the clients on the network cause more than one session a day on average.

Number of days in our dataset	70
Number of associations with access points	1,758,513
Number of sessions	953,484
Without roaming	*785,869 (82.4%)*
With roaming	*167,615 (17.6%)*
Average length of session	42:38
Without roaming	*34:15*
With roaming	*81:53*
Number of unique clients	10,031
Without any roaming at all	*2,191 (21.8%)*
With only roaming sessions	*198 (2.0%)*
Average number of sessions per client	95.0537
Average number of unique access points per session	1.2655
Average number of associations per session	1.8430

Another interesting fact of roaming is the development of roaming when we look at the number of roaming actions and the total number of sessions that show this number of roaming actions. (Figure 3.3) As we can expect, the number of sessions decreases when the number of roaming actions increases. What is interesting though is that it decreases very fast. The number of sessions per number of roaming actions is already under 50 with 49 roaming actions. An irregularity which is not displayed in the graph is that there are two sessions with more than 2,000 roaming actions. These sessions are probably associated with a lab notebook with wireless access or an info display with a thin client attached to it. However, there is no way to tell that for sure.

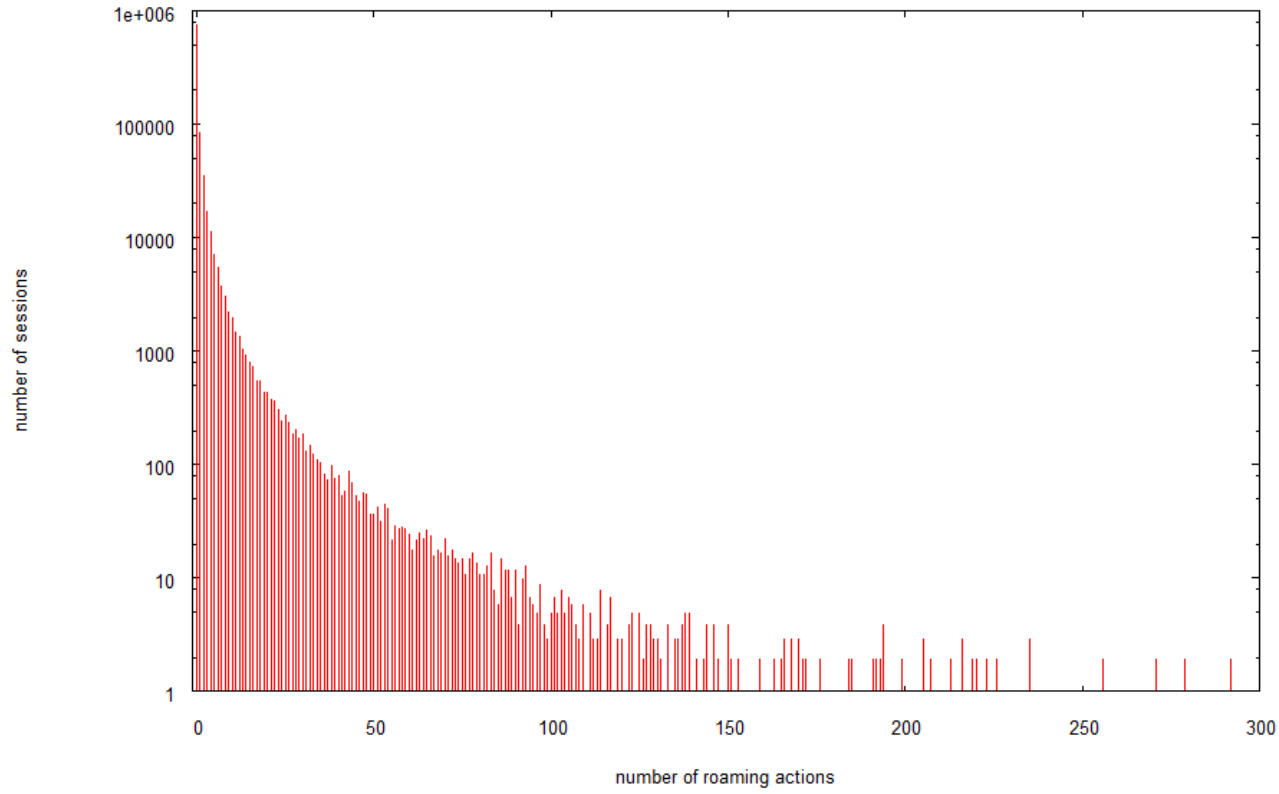

Figure 3.3 The number of sessions containing x roaming actions

3.3.2 Unnecessary roaming

Another interesting aspect we identified in the assignment description is the amount of "unnecessary roaming" on the network. A central point in this aspect is the definition of "unnecessary roaming". Initially we defined unnecessary roaming as the roaming a client causes when he keeps on roaming back and forth between two access points. However, we could identify unnecessary roaming as well when three access points are continually associated with. And what about a client that walks the same round six times a day? (For instance a security officer) Where would we need to stop considering roaming "unnecessary"? Because we could identify a multitude of legit cases which could potentially be identified as unnecessary roaming, we decided to use another strategy.

When we thought about how to represent unnecessary roaming without having to specify all cases that would be unnecessary, we thought about using a fraction to identify this. This fraction would indicate the fraction of associations with access points which are not unique for the session. This means we would count every association with one access point after the first association as unnecessary. If we describe this in a formula, we get this:

x=associations
y=number of unique access points
unnecessary=1 – (y/x)

When we take the weighted average of these ratios for all associations in our dataset, we find that the amount of unnecessary associations is 31.3%. To shed some insight in this percentage, we graphed the percentage of unnecessary roaming aggregated over the number of unique access points in a session (Figure 3.4). When aggregated by number of access points, we would expect the

amount of unnecessary roaming to be non-existent if there is only one access point involved. However, as visible in Figure 3.4, this is not the case. After some research we found that it is related to the maximum timeout (one second) we used between associations to define sessions. Because it is possible that there are two associations within a second on the same access point, which are considered to belong to the same session, we can explain it perfectly. What we can also see from this graph is that the majority of clients don't access more than 3-5 access points. Furthermore, there is a cut between 25 and 34 access points. There are no users that actually roamed to these numbers of unique access points during the period of our dataset.

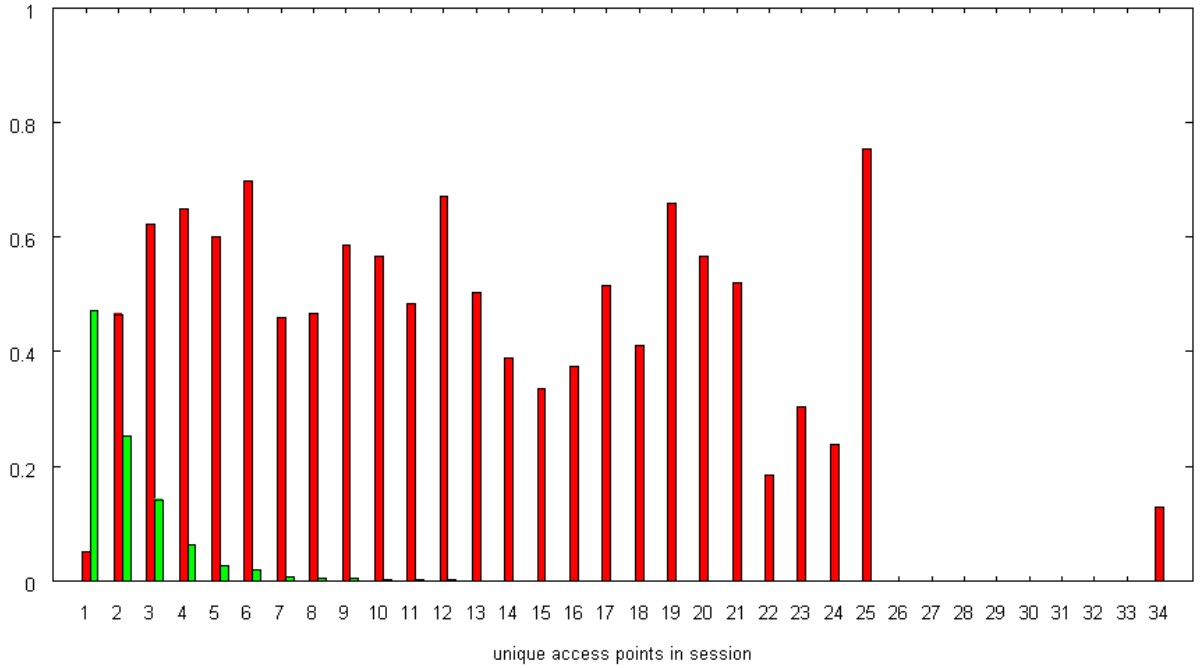

Figure 3.4 The amount of unnecessary roaming (percentage, red) and the percentage of the total number of associations (green) aggregated by number of unique access points in the session

We also aggregated the percentage of unnecessary roaming over the session length in minutes (Figure 3.5). As can be expected the amount of unnecessary roaming increases as clients are longer connected to the network. However, approximately after the 50-minute mark, the percentage of associations drops so low that few associations are considered for the unnecessary roaming. Therefore, the extremes begin to diverge further and further. After the 200-minute mark we cut off the graph because the line only diverges further.

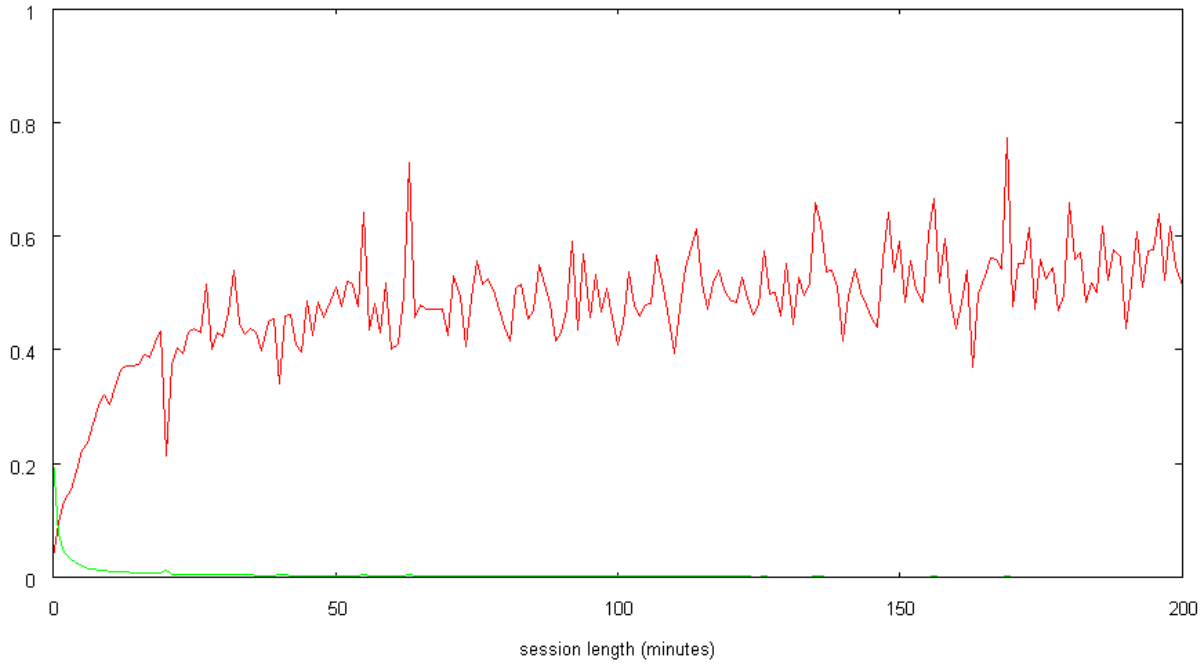

Figure 3.5 The amount of unnecessary roaming (percentage, red), and the percentage of the total number of associations (green) aggregated by session length in minutes, cut off at 200 minutes

4 Roaming visualization

4.1 First approach: 2D visualization

The first approach we considered when we first started thinking about visualization of the roaming the roaming data, was the possibility of representing this data as a 2D image. The access points could be visualized by nodes; the roaming by arrows between nodes since roaming is directional from one access point to the other. Now, the thickness of the arrow could represent the fraction of roaming. This idea is heavily inspired by the Flow Map layout. (Phan, Xiao, Yeh, Hanrahan, & Winograd, 2005)

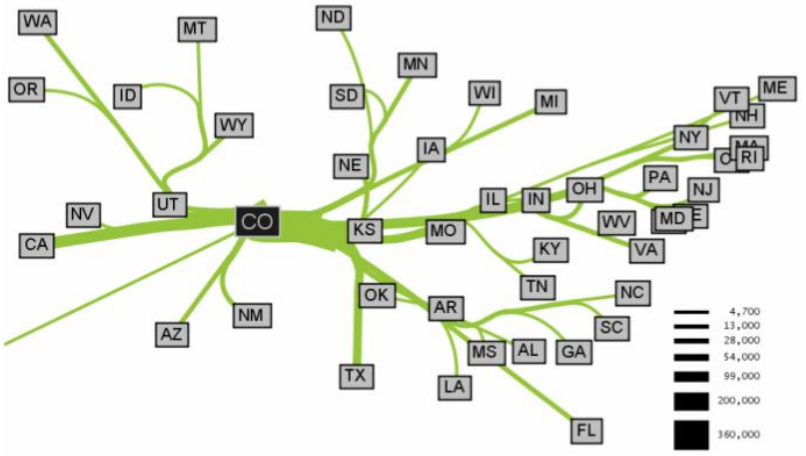

Figure 4.1 Outgoing migration from Colorado (1995-2000) visualized in FlowMap layout (Phan, Xiao, Yeh, Hanrahan, & Winograd, 2005)

At first glance, our dataset looks quite suitable for representation in the flow map layout. (Figure 4.1) After constructing an impression of what this could possibly look like (Figure 4.2) we quickly identified a few problems, which prevented us from exploring this option further.

- *It would be very hard if not impossible to maintain a relation to the location of the access points on the campus.*
  Flow maps tend to work really good for geolocated points between which there are travellers or roaming. Unfortunately, this works rather well when these locations are of the two-dimensional type, and do not stack on top of each other as is the case in buildings with multiple-floor buildings on the campus. It would be impossible to analyse from the resulting visualization at which buildings or physical location most of the roaming occurs.
- *The scale of our data in terms of the number of access points would prevent a clear visualization.*
  There are more than 800 access points which would need to be visualized which either would mean a very large picture which could not be interpreted as a whole or a smaller picture in which it is very hard to distinguish the different access points because of the web of lines in between them.
- *The multiplicity of paths that can be taken from each access point is very large.*
  From each access point it is possible to reach a lot of other access points, which may not be easy to visualize in such a picture. Take for example the access points on high buildings, which have free line of sight to a whole bunch of other access points to which someone can roam while on the ground in between them.
- *There probably is not one unique solution to how to represent the nodes.*
  Closely related to the big number of access points and the fact that it is possible to roam to a big number of other access points from each access point, there will not be one unique solution of how to order the nodes in the resulting visualization.

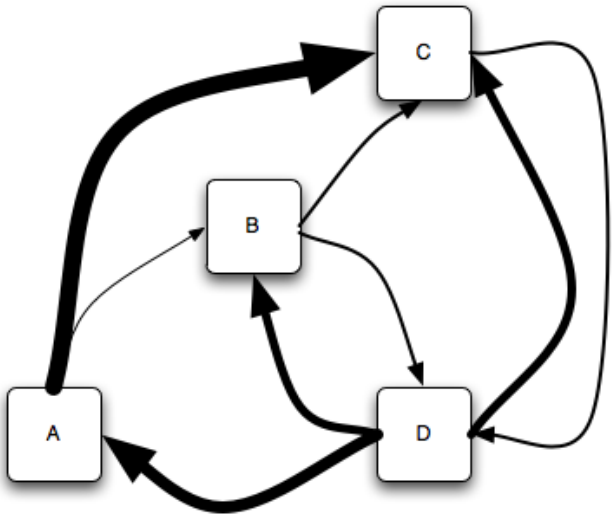

Figure 4.2 Impression of using the FlowMap layout for roaming

4.2 Second approach: 3D

The second approach we considered was the representation of the roaming in a 3D image instead of 2D. This could also be a network of nodes ordered by the amount of roaming going on between each

set of two access points, but the same problems could be identified as with the first approach. Furthermore, the problem of access points that are located by the algorithm behind each other would not be visible. However, this problem would disappear if it is possible to rotate the image either live or by means of using a movie with a rotating 3D picture. Also, the connection of the locations of the access points on the campus versus the location in the 3D image would still not exist. Therefore we decided not to use a 3D image with nodes on arbitrary locations but to use geographical locations instead.

Because there is data about the geographical location of the access points in our dataset, we soon thought of using this for our 3D images. Since this location includes x, y and z coordinates, we could construct an image of the campus by just plotting the locations of the access points. (Figure 4.3) This plot has been made with gnuplot, a cross-platform graphing utility, which is used by many in the academic world. The plot itself is one of the first versions we made, but it is already clearly visible that we can use this to get the relation between the access point nodes and their location on the campus. If you are familiar with the campus of the University of Twente, you will probably see in a glance that the building on the top right position is the “Horst” building. This makes this form of representation very suitable for our problem.

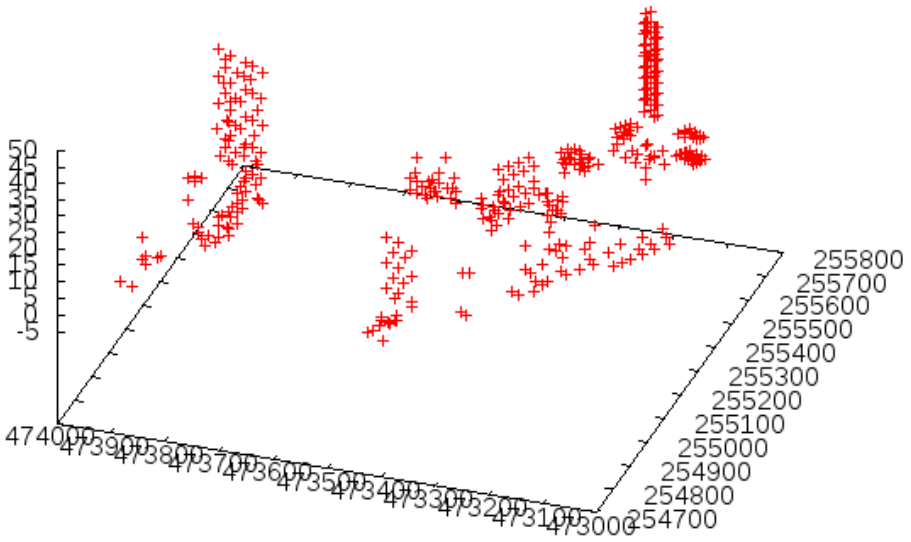

Figure 4.3 A plot of the geolocated access points on the campus. We can clearly see the Horst at the top right.

Now that we have a framework on which we can represent the roaming on the campus, we can begin to look at how to incorporate the roaming actions into the above image. In an earlier stage, we looked at representing the roaming by means of lines of different widths between the access points, to represent the fraction of the roaming going on between each pair of access points. However, when we first tried to represent all roaming present in our dataset on the 3D image, it was evident that it would not be feasible to vary the width of the lines. All roaming actions were represented by a single-width line between two access points. However, since there is roaming from each access point to multiple other access points, this picture made it clear that this representation gets messy very soon. (Figure 4.4) Furthermore, if there are multiple roaming actions between two access points, this is now represented by one single-width blue line. It is impossible to analyse how much roaming is going on between two access points by looking at the visualization. We would need to find another

way to represent the multiple roaming actions between two access points and to decrease the number of roaming actions shown at the same time.

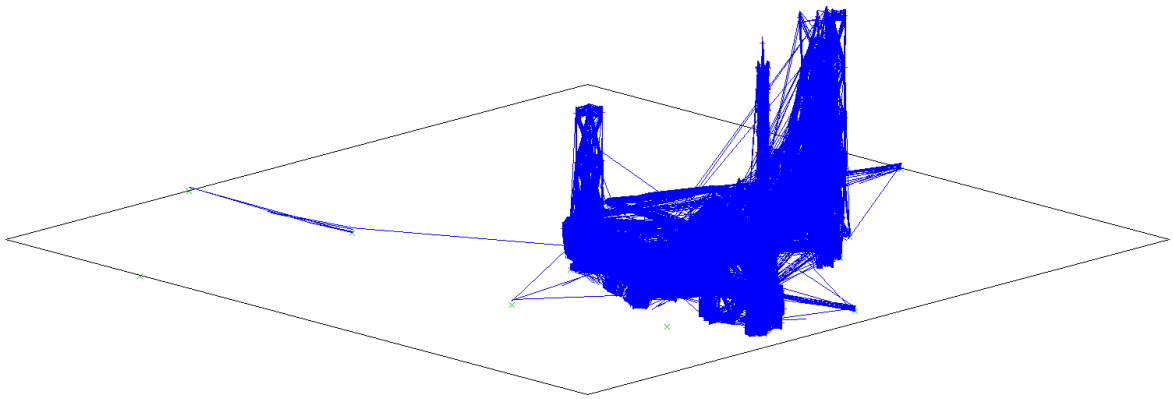

Figure 4.4 A plot with geolocated access points and all roaming in the dataset displayed

4.2.1 Multiple roaming actions between access points

As mentioned above, when we have multiple roaming actions between two access points, we essentially just see one small blue line because all the roaming between these access points starts and stops at the exact same place. (Figure 4.5a) From this indication it is impossible to distinguish between one roaming action and multiple roaming actions between access points.

Initially we solved the problem discussed before this by taking a random point around the access point within two meters on all axes. (Figure 4.5b) This however, did not reflect the direction in which the roaming was taking place and also looks quite chaotic.

A further iteration was the addition of an algorithm to determine where the roaming was coming from and positioning the line ends at that side of the access point. (Figure 4.5c) This is done by comparing the values for each axis with the location of the former access point. If it is a lower number on the axis, the line end should also be placed within the two metres on the lower side of the access point. When this is done for each axis, the line arrives at a more natural side of the access point.

What solved our initial problem of visualizing the amount of roaming between multiple access points provides another problem: when a client roams from access point A to access point B and on to access point C, the incoming line at B and the outgoing line from B get decoupled from each other. Because these roaming actions belong to one session, it would be better if it's an uninterrupted line. To solve this we decided to make the seed of the random number generator we used for defining the offset from the access point dependent on one or more factors contained in the database. To make this work however, we needed to define a few constraints. The randomize function should not return the same location at an access point for one client every time, however if we request the coordinate for the same roaming action twice it should return the same location. Also within a session, there should be random difference between the coordinates to prevent having the same problem as in Figure 4.5a. Therefore, we decided to construct the seed of the random number generator from the clients' mac address, the session identifier and the timestamp. This way, when a client roams back and forth between two access points, we can still distinguish this from just one roaming action. Now

because this function reliably returns the same location for a roaming action at an access point at a given time, we now also have consecutive lines instead of broken lines for all roaming in one session.

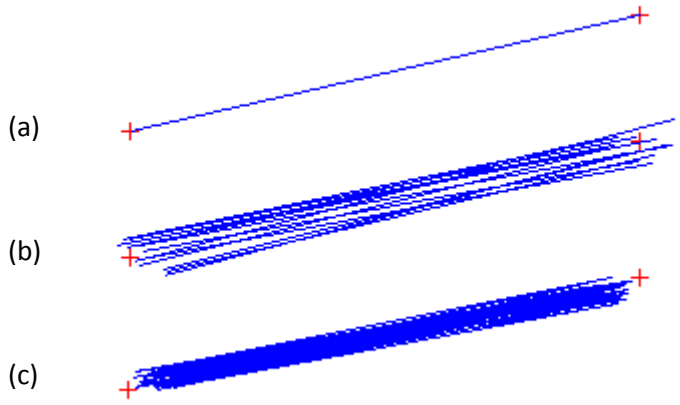

Figure 4.5 Three iterations of the way of visualizing multiple (20 in this case) roaming actions
(a) Roaming actions go to the exact coordinate of the access point
(b) Roaming actions go to a random point within two meters on each axis
(c) Roaming actions go to a random point within two meters on each axis reflecting the direction

The problem we had with the lines hiding the access point (Figure 3.1b), returned because we reintroduced consecutive lines with our algorithm. However, because the lines are initially placed at the side where the roaming action came from, the problem only arises when the user roams to another access point at the opposite side. Since the problem doesn't arise when we roam back in the direction (within about 180 degrees), the problem isn't as big as before. Therefore we decided to leave it this way.

4.2.2 Moving window and animation

As we noted in section 4.2, we have to reduce the number of roaming actions displayed in one single image because it is not feasible to display all roaming of the whole dataset in one image. After experimenting with shorter intervals (weeks, days, hours) to display the roaming data from, we decided that an interval of a few hours or shorter gave enough detail to be able to see what is going on. However, because this is such a short interval in our large dataset, we would be looking at a huge amount of images. So to be able to interpret the visualization, we decided to make a film of the 3D plot. Furthermore, to be able to see some flow of roaming in the film, we defined a method of displaying the data: the *moving window visualization*. In this method, we define a *window* of a certain number of minutes of which we display the data. This method is loosely based on the sliding window protocol which is also used in TCP. Furthermore, we define the *move* which is smaller than the window and contains the number of minutes we slide the window further in the dataset with every new image.

To be able to plot the images from which we could later make a film, we needed to pre-process the data into files with the desired format for usage with gnuplot. Per move, we need one file with the coordinates from and to which coordinates the lines must travel for all roaming actions within that window. To do that, we first load the whole dataset into our program and iterate through the records to find the borders of each window. The indexes of the starts and ends of each window are saved for later usage. Now for each window's start and end we just saved, we iterate again through

that part of the data and save all (randomized) coordinates to a file named to identify which window it contains.

Because the 3D we use in our visualization contains only points and lines, it can be quite hard to see that it is 3D instead of a 2D image. Panning, zooming and rotating the image made it easier to identify different buildings and get a sense of the fact that it actually is 3D. For our visualization we therefore chose to rotate the plot. Trial and error with a smaller part of the dataset teaches us a few things. For the 3D view, an angle of 35-40 degrees is best for viewing our 3D plot. Furthermore, to get a smooth image, we decided to use 24 frames a second, which is also used in the film industry. This was quite convenient since 24 degrees of rotation per second is a nice rotational speed. Also, we decided to move the window one step further four times a second as to give the possibility to the viewer to see movement but not at warp speed and also not too slow to keep the attention of the viewer.

To construct the actual images of which the film is made, another text file with gnuplot commands is created by another Java program. This program constructs this file with the appropriate commands to rotate by one degree between each frame and to change the data file every six frames. Furthermore the plot is refined by setting the title, removing the z-axis and adding labels above the buildings. When this file is executed, gnuplot generates full-HD (1920x1080 pixels) PNG images numbered sequential starting by one. This takes a few hours for the whole dataset dependent on the chosen values for window and move. If we now feed the first image into virtualDub, we can render the sequence into an AVI video file containing a rotating 3D plot of the campus with lines representing the roaming.

For the window and the move we tried different values. With a too long window with a short move however, it looks like a roaming action lasts very long. With a move of 15 and a window of 120 minutes for instance, every roaming action stays on screen for 8 moves or two seconds. With a shorter window there are less roaming actions visible and we should be aware to not bring that down too much. An optimum for the amount of roaming on the campus of the University of Twente was at a window of 30 with a move of 15 minutes. One roaming action is now on screen for just half a second and it looks much more dynamic than the former example.

4.2.3 Geolocating the access points

The access point data contains every access point in the wireless network of the University of Twente. The data contains 1049 rows representing 819 access points, of which 11 access points are in the city centre of Enschede. The data contains one row for each interface of the access points, since each interface has a different mac address. This concerns the access points that support both 802.11b and 802.11g, which are mainly located in the educational buildings. Besides the mac address of the interface, the table consists of a description, location, floor, geographical x – y – z coordinates, name of the access point, name of the interface, channel, power and status (up or down).

Of the fields contained in this data, the geographical coordinates are very interesting for the visualization of the data. These coordinates are expressed in Rijksdriehoeksstelsel (RD), which is the common coordinate system for expressing geographical coordinates in the Netherlands. Unfortunately, *the access point data does not contain geographical coordinates for every access point in the dataset*. There are 527 out of 819 rows in the dataset, which do not have geographical coordinates. Most of these access points without coordinates are access points in buildings, which

are not used for education. So this includes the student housing, children day-care, facility services etc.

A solution to this problem would be to measure the geographical coordinates for all the 527 access points that lack that information. This would be a very time consuming option when done by us, or a very costly option when outsourced. Therefore, we quickly discarded this solution. Another solution to the problem would be to not show access points without coordinates in the visualization. However, this would mean that the visualization would ignore the roaming between 65% of the available access points. Since we would like to provide a representative visualization of roaming on the whole campus, we came up with a compromise solution.

In order to decrease the number for which we had to measure the geographical location, we grouped the access points without geographical coordinates by location. We were able to do this because the naming of the access points in the data is very systematic. The names all begin with the abbreviation of their location (like “vr” for the “vrijhof” and “tennis” for the tennis courts), followed by a clue of their location (floor number, wing number, outdoor or a combination of these) and some other indications, which are either room numbers or arbitrary numbers. The number of locations we had to measure the geographical locations for was now reduced to 35. By interpreting the abbreviation of the location, we could pinpoint the location of these groups on a web mapping service. Then, we converted the coordinates from WGS84 (World Geodetic System 1984, a global coordinate system widely used on the internet) to RD in which the other access points are geolocated. To be able to distinguish between handpicked geographical coordinates and official coordinates, we added a flag in the table to indicate this. To make sure there is a clear distinction in the plots between access points with “official” coordinates and groups of geolocated access points, we decided to make these stand out by assigning another colour to these groups. The associations with “official” coordinates in all our plots and movies are coloured red and the groups of access points with a handpicked location are coloured green.

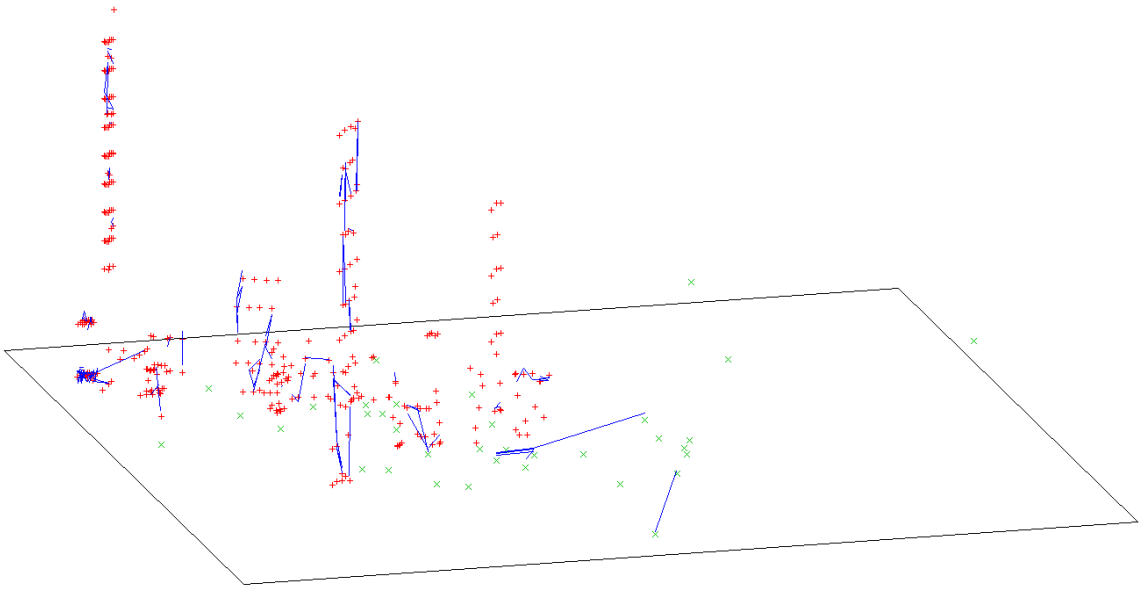

Figure 4.6 Geolocated access points (red), groups of handpicked coordinates (green) and roaming (blue)

Using this compromise solution does have a drawback. By grouping multiple access points into one group, the roaming within this group cannot be visualized. One of these groups of access points for example contains a row of student houses on one street of the campus. Since every house contains at least one access point, the potential amount of roaming in this group is quite big. However, we can still see the roaming between these groups. Although roaming between these groups of access points is less common than roaming between access points within these groups, this compromise solution actually shows some roaming at these access points as opposed to no roaming at all when we left out all these access points, so this solution is still favourable above the other.

5 Conclusions and recommendations

In the assignment description we posed the question how much roaming is actually going on in the wireless network of the University of Twente. Although the actual number of sessions in which roaming occurs seems to tell a story of little roaming occurring on the network (17,6%), the number of clients who didn't roam in the whole dataset (21,8%) tells a completely different story. With almost 80% of the clients having roamed at least once in the dataset, we cannot say that roaming occurs only with a select few clients.

While we thought mobile devices would increase the amount of roaming on the network, it could very well be that it actually doesn't. Because smartphones and other ultraportable devices don't have heavy batteries included, they have to use energy sparingly. Since internet connections actually take a lot of energy, most smartphones seem to break up the connections to the network in little actions. For instance retrieving mail and terminating the connection every five minutes. Because these actions are short, the chance of roaming in such a session is minimal.

The question how much "unnecessary" roaming occurred on the wireless network, has been answered in this assignment by using a factor of unnecessary roaming in a session. With 31,3% the percentage of "unnecessary" associations with access points seems quite high. However, because of the method we used this is a pessimistic estimate since not all associations we deemed unnecessary are necessarily unnecessary. Therefore we recommend further research on this subject. It might be a good idea in future research to exhaustively describe all forms of "unnecessary" roaming and filtering the corresponding records from the dataset.

In the beginning of this assignment we were also looking for an answer to the question if roaming increased over time due to the ever-increasing penetration of smartphones and other mobile devices. However, due to a lack of time we couldn't answer this question in this assignment. Since the penetration of ultraportable mobile devices grows bigger each day, it is an interesting question and one we can certainly recommend to be further researched in a later assignment.

A large part of this assignment was focused on the visualization of the roaming data. The goal was to visualize the roaming data in a concise and insightful way that is easy to understand without too much detailed knowledge of roaming. We had three particular aspects which we would like to see in the visualization: when, where and how much. All three of these aspects have been incorporated in our video-visualization. The aspect "when" is answered by the title which constantly updates shows the date and time we're looking at in the video. The "where" aspect is answered by the title (University of Twente campus) and by the geographic location of the access points as well as the vertical labels atop the buildings. Finally, the aspect of how much roaming is going on is slightly

harder to see but thanks to the algorithm we invented for placing the ends of the lines that represent the roaming, we can approximately distinguish the quantity of roaming.

References

Diepenhuis, M. (2003, January 23). Draadloos internet op campus komt eraan. *UT Nieuws*.

Kotz, D., & Essien, K. (2005). Analysis of a Campus-Wide Wireless Network. *Wireless Networks 11*, 115-133.

Phan, D., Xiao, L., Yeh, R., Hanrahan, P., & Winograd, T. (2005). *Flow Map Layout.* Stanford: Stanford University.

Schwab, D., & Bunt, R. (2004). Characterising the Use of a Campus Wireless Network. *IEEE INFOCOM 2004.*

Zola, E., Barcelo-Arroyo, F., & López-Ramírez, M. (2009). User behaviour in a WLAN campus: a real case study. *Third Ercim Workshop on Emobility* (pp. 67-77). Enschede: University of Twente.