The Network Beacon Announcement scanning method for 802.11 networks

A bachelor thesis by

Dirk Kuijsten

Supervised by

Prof.dr. R.J. Meijer

Confidential

-Not for publication or distribution-

Department of Computer Science, Faculty of Science, University of Amsterdam

Universiteit van Amsterdam
The Network Beacon Announcement scanning method for 802.11 networks

Dirk Kuijsten
dirk.kuijsten@student.uva.nl

23-08-2004

Approved: ...........................................

On: ..............................................
1 Abstract

A new method for scanning other access points is needed for 802.11 networks to support realtime applications like Voice Over IP. The current scanning method is the main cause of high latency transitions of clients between access points. I have developed a new scanning method that should improve these transition latencies significantly. The Network Beacon Announcement (NBA) method has multiple advantages over other methods in terms of speed, quality of service and power-efficiency. The method is modeled after functional and network requirements and is specially designed for infrastructure networks and can be used for network management, network optimization and fast roaming. It maximizes the time available for data transfers, thus increasing throughput and decreasing latency. This is achieved by minimizing the time spend on other channels while scanning the availability of other access points.
## Index

1. Abstract........................................................................................................................................2
2. Introduction..................................................................................................................................4
   2.1 802.11 wireless networks......................................................................................................4
   2.2 Fast roaming in wireless networks........................................................................................5
   2.3 Related work ..........................................................................................................................5
3. Improvement of the scanning phase .............................................................................................6
   3.1 Requirements .........................................................................................................................6
   3.2 Simple enhancements in the scanning method .........................................................................7
   3.3 Channel access time ...............................................................................................................7
   3.4 Current method: non-directed probe request .........................................................................7
   3.5 Overview of proposed improvements for the scanning method .............................................8
      3.5.1 Sending beacon frames more frequently .........................................................................8
      3.5.2 Sending special short beacon frames frequently .............................................................8
      3.5.3 Directed probe request [7] ...............................................................................................9
      3.5.4 Directed probe request with fast probe response [6] .......................................................9
      3.5.5 Directed probe request with acknowledgement ............................................................10
      3.5.6 Synchronized beacon frames with known timing ..........................................................10
      3.5.7 Extra beacon frame requested by the client .................................................................10
      3.5.8 Scanning in promiscuous mode .......................................................................................10
   3.6 Summary of the proposed scanning methods .........................................................................11
4. A new scanning method................................................................................................................12
   4.1 Architectural design ................................................................................................................12
   4.2 Network Beacon Announcement ..........................................................................................13
   4.3 Theoretical performance analysis ........................................................................................14
   4.4 NBA summary .......................................................................................................................15
5. NBA reference implementation.....................................................................................................17
   5.1 Introduction .............................................................................................................................17
   5.2 The adapted driver ................................................................................................................18
   5.3 Time synchronization ............................................................................................................20
   5.4 Experimental setup ..............................................................................................................20
   5.5 Results ...................................................................................................................................21
6. Conclusion and future work.........................................................................................................23
   6.1 Conclusion .............................................................................................................................23
   6.2 Future work ...........................................................................................................................23
7. References....................................................................................................................................24
8. Appendixes ..................................................................................................................................25
   8.1 Sequence diagram of NBA proposal .....................................................................................25
   8.2 Sequence diagram of implementation ...................................................................................26
2 Introduction

2.1 802.11 wireless networks

The Institute of Electrical and Electronic Engineers (IEEE) adopted the first wireless Local Area Network (LAN) standard in 1997. The IEEE 802.11 standard is also known as the Wireless Fidelity standard (Wi-Fi), used to certify 802.11 compliant hardware. It is the most popular standard for wireless networks, because the standard supports different types of usage and is suitable for home and enterprise networks. The 802.11 standard extends the IEEE 802 LAN standard with a media access control (MAC) layer and multiple physical (PHY) layers, but is much more complex than layers for wired networks. Wireless networks need more features such as media access sharing, authentication and encryption. And because the standard uses unlicensed spectrum it also employs methods to avoid interference of other devices (phones, microwaves) or interference between multiple 802.11 networks sharing spectrum bands. Thus the standard introduces many new names and abbreviations that identify all the features and properties.

802.11 LAN networks can be created by setting up an ad-hoc wireless peer to peer network or by forming a wireless infrastructure network. An infrastructure network consists of clients and access points. Any device that uses the wireless network is defined to be a station (STA), but often a client station using an access point (AP) is named a station. Access points provide a Basic Service Set (BSS), which consists of all network services provided by the wireless network. Example services are a distribution service: the distribution of data between nodes in the network, an integration service: the integration of a wireless network in other networks, like Ethernet networks, and an authentication service: providing control over which clients can use the wireless network. Multiple access points in the same wireless network together form the Extended Service Set (ESS).

Each access point covers a small area, the basic service area, and the ESS forms the extended service area. A Client station can move from one basic service area to another with the use of the ESS reassociation service. This movement is formally called a BSS transition, provided by the reassociation service, but it is referred to as roaming or a wireless handoff.

![Figure 1: A wireless handoff](image)
2.2 Fast roaming in wireless networks
The 802.11 standard was not designed for large-scale, dynamic networks, but for small networks with static clients. Clients were supposed to be portable, not mobile. Portable clients would work with a notebook computer on one place and move to another without the need to remain connected to the network. Mobile clients are working on the move, requiring a continuous connection to the network. Mobile clients have begun appearing the last three years with the introduction of Wi-Fi enabled PDA’s and Voice Over IP (VOIP) telephones. The 802.11 standard therefore does not offer features that support fast roaming within and between wireless networks. Realtime applications, like telephony and multimedia streaming, used by mobile clients require a reliable network with continuous throughput and low latencies. The problem with these applications in current 802.11 networks is when clients roam between coverage areas of access points. The handoff period between two access points is too high for realtime applications.

The search for other access points, the probe phase, is the dominant factor in the handoff process as shown by Mishra et al. [2]. It measured the probe phase to account for 90% of the total handoff latency. The overall average handoff latency is measured at 186 milliseconds, but some configurations showed an average of 390 milliseconds. This means the probe phase in the measurements was about 167 and 351 milliseconds respectively. In the probe phase the client station searches all wireless channels for access points. The client actively asks responses from any access points on the channel (an active scan) or listens for beacon frames from access points (a passive scan). The passive scan is more power-efficient because the wireless transmitter can be turned off on the client hardware. Both methods require a client to spend a long time on the channel. The active scan searches for all access points on the channel, even from other networks outside the ESS. The passive scan duration depends on the interval of beacon frame transmissions, which is normally at 100ms. Because the scanning phase is the slowest part in the handoff process I decided to develop a new 802.11 scanning method.

2.3 Related work
There is much research being done currently on improving the 802.11 standard [3]. Not only does the standard lack advanced roaming support, but also support for security, quality of service (QOS), inter network handoffs and network management. All these topics are covered by individual task groups (TG) working to extend the standard to implement new features. An example is the recent finalization of the 802.11i standard by the IEEE, which greatly improves security. It introduces the Advanced Encryption Standard (AES) for the encryption of data, an upgrade from the insecure Wired Equivalent Privacy (WEP) standard used today in most networks. Other task groups have previously worked on higher speeds for the standard: the 802.11b, 802.11a and more recent 802.11g standard.

One task group is working on 802.11r (TGr) [4], formerly the Fast Roaming Study Group. Its objective is to make the BSS-transition faster. The study group became a formal task group only recently in June 2004. Currently there is only one preliminary proposal that focuses on the scanning stage of the BSS-transition. The proposal consists of informing clients of neighboring access points together with a minimal adjustment to the current non-directed probe request/probe response mechanism (see 4.1.1). The adjustment is only to decrease the size of the probe response, resulting in a faster transmission time.

This dynamical subject made working on this thesis very hard because each month it became more outdated because of new developments in this field of research. While the 802.11r standardization process continues, increasingly more reference work will become available.
3 Improvement of the scanning phase

3.1 Requirements

What is expected from a new scanning method, what should it do and how fast does it work? To answer these questions let's first look at the network itself. The focus is on wireless networks in infrastructure mode. This means that there are clients and access points in the network. I assume that there are multiple access points in the network, because of the low range of Wi-Fi networks, which form a shared wireless network together (the extended service set or ESS). What are the requirements for the different parts of this network?

Requirements for individual clients:
- The highest availability of the network, in combination with client mobility
- The highest data throughput
- The lowest data latencies
- The lowest data losses
- Possibly some extra information about the client’s location or availability of other (backup) networks (for example other Wi-Fi networks or UMTS/GPRS networks).

Requirements for the complete network:
- The highest overall connection of clients to the network
- The highest overall throughput of all clients
- The lowest overall data latencies for clients
- The lowest overall data losses for clients
- Coverage information of the network.
- Network availability information of the network.
- Information about other (interfering) networks, malicious networks or users, etc.

The scanning method should thus have the following characteristics: it will have to be fast, accurate and share and use information available in the network. If we provide more information in the network to clients and access points, the method can be more advanced. Information that is useful for a scanning method can be the number of available access points, their channels, their addresses, their relative location to other access points, access point load, access point range and client locations. If for example the client station knows for which access point to scan for, if it has lost its current wireless connection because the client moved out of range, the scan can be done much faster. Thus one thing the method should be doing is using all current network information available.

While a client is scanning its network interface isn’t available for sending or receiving data. This means it can’t fulfill the requirement of the client’s network at this time. Only minimizing the time spend scanning can be insufficient: data losses could occur if the access point has data for the client. If the access point does not know that the client is not available it will transmit and discard the data. The 802.11 standard uses retransmission of unacknowledged frames resulting in extra bandwidth loss. Thus the access point should be informed that the client is unavailable, otherwise this will only lower the overall network throughput available to other clients of the access point.

The method should be fault tolerant and should respond quickly in case of network failure. The method should handle signal loss of the AP effectively by reconnecting to the network as fast as possible.
3.2 Simple enhancements in the scanning method
There are multiple obvious improvements over the current scanning method that can be used for a new method. The first improvement the client on which channels to scan for access points within the same network. Most deployments only use tree or four channels of the available thirteen channels, because the channels overlap. Only scanning four out of thirteen channels is the first improvement. Secondly there can be other access points on these channels that do not belong to the client’s network. If the client knows the exact access points to scan for on each channel this can decrease the time spend on each channel. The third improvement would be scanning only the neighbors of the access point the client is connected to. If access points would be informed by network management functions of their neighbors they can inform the clients of these neighbors. The final most sophisticated approach would be the use of client location information, by using GPS or triangulation.
Overall there can be done a lot to minimize the complete search for different access points, but to incorporate all optimizations in a new method is not enough.

3.3 Channel access time
Timings that are very important for the speed of a scanning method are channel access times. This means how long clients need to wait before being able to send data. To allow multiple clients and access points share the same medium (the clients all share the same spectrum band), the 802.11 standard uses Carrier Sense Multiple Access (CSMA) to control access to the medium. Because collisions can occur more often because more clients can share the same medium the standard uses Collision Avoidance (CA) instead of Collision Detection (CD) used in 802.3 Ethernet. The implementation of these methods is divided in several parts. The first part is the Short Interframe Space (SIFS). This is used to determine if the medium is busy and after this time only high-priority frames can be transmitted. These frames are the Request To Send (RTS), Clear To Send (CTS) and acknowledgement (ACK) frames. After the SIFS time comes the DCF Interframe Space (DIFS). This timing is the minimal time used by the Distributed Coordination Functions (DCF), an access mechanism used by almost all 802.11 networks today. The DCF specifies a minimal time, the DIFS, plus a random interval time. This interval, chosen by each node in the network at random, avoids that two clients access the medium at the same time. If two or more clients chose the same random time, they should notice this and will try again at a later, larger random interval. This is the random backoff referred to below. The probe response will be made in unknown time because of the DIFS backoff mechanism. Another, optional coordination function was developed for the 802.11 standard: the Point Coordination Function (PCF). But it was not implemented by most manufacturers.

3.4 Current method: non-directed probe request
The current method works by sending a probe request for information on other access points. A probe request by the client is a broadcast message to any access point on the same channel. The client waits a minimal time (or MinChannelTime) for any activity on the channel. If there was no activity, the client assumes the channel to be empty. If there was any activity, the client waits for replies of access points for a maximal time (or MaxChannelTime). If the MaxChannelTime is set to a low value, there is no guarantee that the access points the client is interested in (within the same ESSID) will be able to respond in time. Other clients of the access point, which are in communication with the access points, or other access points of other networks (ESSIDs) could respond to the probe request. This other unwanted activity makes it impossible for the required access point to respond within the maximal time. Therefore the MaxChannelTime is set on a high value around 100 milliseconds. Velayos and Karlsson [5] showed that by tweaking the MaxChannelTime the scantime can be reduced to 70 milliseconds.
Another problem is the probe request send by the STA (active scan). This is less power-efficient than waiting for beacon frames (passive scan).
3.5 Overview of proposed improvements for the scanning method

There are several methods to enhance the scanning for access points. Besides the standard settings and tweaks to speedup the scan, there are several new methods proposed to the IEEE. The next part is an overview of the current and proposed methods for scanning, together with their advantages and disadvantages:

3.5.1 Sending beacon frames more frequently

The standard setting for 802.11b networks sends beacon frames every 100 time units at 2Mbps. A time unit (TU) is 1.024 microseconds. If the beacon frames are send every 10TU, the client could wait for the beacon in stead of starting an active scan. Because the channel will be used more for management frames the overhead will increase, thus the channel bandwidth will decrease.

+ no active scan by the client: less power-efficient.
+ a better determined maximum wait time: near 10 ms.
+ no need for extra implementation of the method in clients.
- higher overhead, thus less bandwidth available to clients.

3.5.2 Sending special short beacon frames frequently

The current beacon frame consists of much information that doesn’t change frequently, but is transmitted every time. If you would omit this information, the frames can be shortened and the overhead on the channel can be lower. The gained bandwidth can be used to send more (shorter) beacon frames, for example every 10TU.

+ no active scan by the client: less power-efficient.
+ a better determined maximum wait time: near 10 ms.
+ no need for extra implementation of the method in clients.
- Only a slightly higher overhead.
3.5.3 Directed probe request [7]

A directed probe request by the client is a unicast message to the access point on the channel of that access point, instead of a broadcast to all access points on that channel. The client therefore only has to wait for one response. The maximum time is therefore related to the time a probe response can be done in crowded networks. The response will come in unknown time because of the DIFS backoff mechanism.

![Figure 3: Schematic overview of directed probe request](image)

- faster than broadcast probe request.
- very fast with low network loads.
- undeterminable probe response time.
- client needs new drivers to send unicast probe request frames.

3.5.4 Directed probe request with fast probe response [6]

The access point answers directly (after SIFS timing) to the client instead of a random backoff as in 3.5.3. The client does not acknowledge the frame, because the client knows the access point received the probe request. The access point will see the absence of another probe request as an acknowledgement. The total time can be determined by:

\[
\text{time} = \text{DIFS} + \text{random backoff} + \text{probe request frame transmission time} + \text{SIFS} + \text{probe response frame transmission time}.
\]

![Figure 4: Directed probe request with fast response](image)

- fastest possible answer by the access point: after SIFS time.
- time determined answer by AP if it is on the channel and in range.
- maximum time determined by channel access time for probe request (DIFS + random backoff). This can be described with a probability distribution dependant on the load of the access point (or medium).
- less power-efficient: active transmission by the client.
- more scans: higher overhead on channel.
- not conforming to current 802.11 standards: no acknowledgement from the client. This needs a firmware update for the access point as well as for the client.
- new management frame for probe request and response. The method should be able to distinguish clients using the new and old method. This means the method is backwards compatible with existing hardware.
3.5.5 Directed probe request with acknowledgement

This is the same as in 3.5.4, but instead of replying with a probe response, the access point will reply with an acknowledgement frame. This is in accordance with the current 802.11 standard, which requires acknowledgements of all unicast messages. The client should in turn be able to receive and process the acknowledgement.

- SIFS timing response.
- No modification of the 802.11 standard: already uses acknowledgements.
- New management frame for unicast request.
- Client hardware needs a firmware update to process the acknowledgement as a probe response frame.

3.5.6 Synchronized beacon frames with known timing

All access points synchronize their beacon frame communication. This can be done with the Inter Access-Point Protocol (IAPP). This is a supplement to the 802.11 standard (IEEE 802.11f) that enables standardized communication between access points. The access point could communicate the timings of other access points to the clients together with a neighborhood map for example. The clients switch to the channel of another access point on a predetermined time, to wait a short period for the beacon frame. The client knows in advance the next beacon transmit time of the other access point.

- Almost determined maximum time: client knows when next beacon frame is send. However, beacon frames also use the DIFS and backoff mechanism.
- Not a overhead off extra scans or beacon frames.
- Passive scan by client: power-efficient.
- Not useful if client needs to switch to a other immediately. The maximum time will then be longer but known: it is the transmission time of the next beacon frame.

3.5.7 Extra beacon frame requested by the client

The client will ask an extra beacon frame from the other access point via the current access point (directly over the network to the other access point, or to the current access point that handles the rest via IAPP). The other access point will respond with the timing of the beacon frame transmission. The client switches to the channel just before the specified time to listen to the beacon frame.

- Short time on the other channel. The request is done on the current channel, on which the client can still transmit other data.
- If the current access point sends the information in broadcast, multiple inactive clients can scan concurrently.
- The current access point is informed about the client status: frames destined for the client can be buffered like in power save mode.
- Extra management requirements of access points.

3.5.8 Scanning in promiscuous mode

The client can use any scanning method, but also listens to other frames from the access point to other clients. These frames also allow the signal strength measurement that is necessary for the scan. The client would have to be in a promiscuous mode, enabling the reception of frames not destined for the client.

A better defined upper time limit is now possible with the scan, because the access point will definitely send frames to clients, either a beacon frame or probe response or to other clients that send/receive data from the access point, forcing it to send data or acknowledgement frames. The placement of other access point on the same channel should not be in range of this access point, otherwise they could delay transmissions from the access point.

- Defined upper time limit: this would make the used scanning method faster.
3.6 Summary of the proposed scanning methods

Table 1 shows the overview of all the scanning methods. Included in the table is my proposed method described in chapter 5. For each method the different properties are displayed. The time needed is defined in the type and time of channel access used (SIFS/DIFS) and the number of frames transmitted. The possible speed follows from this time needed. For beacon frames this is the average beacon frame transmission time (in time units). The problem with beacon frames is that they are transmitted at a standard interval. Clients cannot wait on this interval if they need to scan directly. This is the cause of the maximal possible speed.

<table>
<thead>
<tr>
<th>methods vs. properties</th>
<th>client status known</th>
<th>determined time</th>
<th>active or passive</th>
<th>possible speed (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td># SIFS # DIFS frames</td>
<td></td>
<td>min.</td>
</tr>
<tr>
<td>Current</td>
<td>No</td>
<td>1 2 3</td>
<td>No</td>
<td>active</td>
</tr>
<tr>
<td>High beacon rate</td>
<td>No</td>
<td>0 1 1</td>
<td>No</td>
<td>passive</td>
</tr>
<tr>
<td>High beacon rate and small beacons</td>
<td>No</td>
<td>0 1 1</td>
<td>No</td>
<td>passive</td>
</tr>
<tr>
<td>Directed probe request</td>
<td>No</td>
<td>0 2 2</td>
<td>No</td>
<td>active</td>
</tr>
<tr>
<td>Directed probe request; fast response</td>
<td>No</td>
<td>1 1 2</td>
<td>No</td>
<td>active</td>
</tr>
<tr>
<td>Directed probe request; fast ack</td>
<td>No</td>
<td>1 1 2</td>
<td>No</td>
<td>active</td>
</tr>
<tr>
<td>Synchronized beacon frames</td>
<td>No</td>
<td>0 1 1</td>
<td>No</td>
<td>active</td>
</tr>
<tr>
<td>NBA scan</td>
<td>Yes</td>
<td>1 0 1</td>
<td>Yes</td>
<td>passive</td>
</tr>
<tr>
<td>Promiscuous mode</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 1: Overview of the different methods with their properties
4 A new scanning method

4.1 Architectural design

For the new scanning method I implemented a few distinctive features to meet the requirements for clients and the wireless network as described earlier:

a. The current access point should be informed that the client is unavailable
b. The client should report scan reports back to the network
c. Minimal time spend on other channels while the client is scanning

To begin with the last feature we need to look at the scan measurement process. The measurement reported by scans is the signal strength of access points. The whole purpose of the scan is to get this signal strength. The client needs only one packet from the desired access point to report the signal strength. It would also be possible to let the access point perform the scan. In this case the client would need to send one packet to the access point. If we want to minimize the time on the channel to the absolute minimum, what is the time needed to scan for one access point?

The minimal time needed for the scan measurement is the time in which a single packet can be received by the client from an access point. Thus, first we need a single packet from the access point. There are three options to get the access point to send such a packet:

1. Let the access point send packets at specified times: beacon frames.
2. Ask the other access point to send it at and specified time via the current (old) access point’s connection.
3. Request it on the channel of the other access point.

Requesting the packet on the other access point’s channel requires extra time on the channel, thus violating the objective. This is also true if we need to wait for the next beacon frame. Table 1 shows the time on the channel expressed in the number of frames and channel access timings (DIFS and SIFS). The 802.11 standard specifies acknowledgement frames from the receiver for each unicast frame send. Those acknowledgement frames are sent after the SIFS time. All other frames that need to be sent follow the CSMA/CA design used in the standard. This means waiting for the DIFS time plus a random backoff time. During this time other access points or clients can begin transmission before the backoff time is over. This will lead to retries with longer backoff times. The table contains the number of frames and timings that result from the different methods proposed as described earlier.

It is clear in the table that a single broadcast message from the access point to the client has the shortest possible duration. This would require the access point and client to know when the frame transmission will start. This can be communicated on the channel of the current access point when the current access point clears the client to begin with the scan.

The first feature, that the access point should be informed of the unavailability of the client (a), can be combined with a timing request. A timing request is a request from the current access point to the other access point to agree on a time to send the broadcast frame to the client. The timing reply from the other access point will notify the current access point that the client will be unavailable at this time. Thus with the time information available on the current access point, it can inform the client of the time and set the client at unavailable internally at that specific time.

Now the client is informed of the time the frame is send, it can minimize the time spend on the other access point’s channel, because the time is known exactly. Thus feature c, a minimal time spend on other channels while the client is scanning, is accomplished. The last feature, that clients should inform the scan results can easily be added to a reassociation request. This request is already a frame used in the standard if a client station decides to move from one access point to another.

This is now a new solution that follows the network requirements and should be faster than competing methods.
4.2 Network Beacon Announcement

From the solution that follows the requirements specified, I propose an overall new method to initiate and perform client network scans. The method extends the solution by changing the initiation of scans to allow networks to efficiently combine multiple scanning stations. Not only the clients should be able to initiate a scan, but preferably the access point should initiate scans. Access point can start a scan following a client scan request or initiate a scan automatically. This is why I have named the new method “Network Beacon Announcement” (NBA): clients are informed by their current access point with the announcement of special, extra beacon frame transmissions (NBA frames) of other access points within the network.

The current access point starts a scan by selecting its current neighbors. These neighbors are any other access points that could be in range of the client, or clients, requiring a scan. The distinction between a single client and multiple clients is important. A scan can be performed by multiple clients at the same time because no individual client interaction with other access points is required during the scans. Thus combining multiple scans would use the overall bandwidth more efficiently. The difference between a single client and multiple clients can be important if a more advanced neighbor selection process is used. Several studies have shown that client localization is possible [8],[9]. Thus if the location of the client is known neighboring access points can be selected in a different order or some could be discarded.

This is because the method tries to minimize the time the client(s) need to scan for access points with good reception. If the list of access points to scan for is ordered successfully, the closest access points to the client(s) should be scanned first. After a successful scan clients should return to their own access point or move to a scanned access point if signal strength is better than that of the current access point.

After the selection of a list of neighbors, the current access point informs all other access points on the list to send a beacon frame on a specified time. The current access point selects these times on the order of the list. The fist access point on the list will send its beacon frame first, followed by the other access point on the list. The times specified by the current access point should be spaced enough to allow clients to switch between the channels. The times also need to be spaced extra to compensate for inaccuracies of the actual time between access points. See below for more information on time synchronization.

When all other access points are informed the current access point can issue a frame to the client or clients informing them to start a scan. The frame contains the list with the neighbor access points together with their channels, time of beacon transmission and timeout number. Clients that are active can ignore the scan initialization if they have a good signal reception of the current access point. Clients with low signal reception and inactive clients can start the scan. The method could be extended to force clients with good signal reception to start a short scan. This would force the clients to scan a few other access points and report the results. More information would be available for network management.

Clients now have all information needed to start the scan. They switch to the specified channels on the specified times to wait for frames from the other access point points. The other access points should send a NBA beacon frame. This is a special frame, different from normal frames. It can be an empty frame, with only a frame header containing the source address. Thus this frame is very small, comparable with the size of clear to send and acknowledgement frames. Like these frames the NBA frame should be transmitted after the SIFS time. This ensures that the frame is transmitted before other stations can transmit their frame. It is also the fastest method and does not require the DIFS backoff mechanism. This gives the method a predetermined maximal time of transmission.
Note that the clients should listen to any frame from the access points, because any received frame can indicate signal strength. This is because another client can be transmitting data to the other access point at the NBA specified time. Immediately after the data frame should come an acknowledgement frame to that client from the access point. This frame can also be received by the scanning client and also indicates signal strength. The clients should listen for a maximal time specified by the timeout. This timeout is based on the accuracy of the actual clock time between access points and is determined by the current access point. Clients with low signal reception on the current access point should stop the scan if another access point is found with better signal reception. How much better this reception should be must be determined dynamically by the client. In networks with a high density of access points it could be sufficient to scan for a few other access points. In networks with a low density of access points, where the chances that multiple access points cover an area are slim, it could be better to select the first access point within range.

If a client has finished its scan it can return to the old access point or select a new access point. If it returns to the old access point it should report back any successful scanned access points. The old access point stores the information of the scan and sets the clients internal state from unavailable to active. If a client selects a new access point it should reassociate to the new access point. Together with the reassociation information it sends information of other access points scanned successfully. The new access point adds the client to its list of clients and informs the old access point to transfer the client. The old access point can update its internal list to remove the client and can forward extra information to the new access point. This information could be authentication information like encryption keys, client statistics or buffered data destined for the client.

See figure 5 for a schematic overview and Appendix 8.1 for a sequence diagram of the NBA scanning method.

4.3 Theoretical performance analysis

What is the total maximal time needed for a scan measurement [10]? This can be exactly determined theoretically because the NBA scanning method provides a predetermined maximal time by using the SIFS time mechanism.

I explained that another client can be in the middle of a transmission when the NBA beacon frame has to be transmitted by the access point. This data frame can be the largest possible frame size that the 802.11 standard supports: 2304 bytes. This transmission cannot have been started earlier as the SIFS time frame because otherwise the other client would detect a collision and would wait. After the data frame is transmitted the access point should send an acknowledgement frame after the SIFS time frame following the standard. This frame can be received by the scanning station, thus this is the maximal time a client should theoretically wait for the NBA scan, should all access points have exactly the same clock time.

Normally faster access point channel access is possible after the SIFS time frame. It can send a NBA frame of for example 20 bytes. Table 2 shows the overview of the timings. These timings are for the 802.11b standard. The lowest supported speed is 1Mbps thus this is the largest possible transmission speed of the frames. When using the NBA scanning method in networks with a high density of access points, the slower speeds supported can be disabled, allowing a maximal total time of 1706 µs at 11Mbps, the normal speed of 802.11b networks.
<table>
<thead>
<tr>
<th></th>
<th>bytes</th>
<th>time (us)*</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFS</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Maximal 802.11 data frame size</td>
<td>2304</td>
<td>18432</td>
</tr>
<tr>
<td>SIFS</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>ACK frame</td>
<td>14</td>
<td>112</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>18564</strong></td>
</tr>
<tr>
<td>SIFS</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>NBA frame</td>
<td>20</td>
<td>160</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>170</strong></td>
</tr>
</tbody>
</table>

Table 2: Maximal time for NBA scan  
*data transmitted at 1 Mbps

Although the table shows a maximal value of 18.5 milliseconds, this normally should not be the average time. Most wireless networks deployed do not have an overall high network load, thus there is little chance that clients are transmitting large data frames at a slow speed. Individual access points with high load are more typically closer to each other because network administrators try to distribute the load over more access points. Thus most clients can transmit frames at higher speeds because the nearest access point range is smaller. Clients also always try to send at the highest possible speed available, most of the time at 11 or 5.5Mbps.

4.4 NBA summary

Advantages:
+ The current access point knows the unavailability of a client: frames can be buffered for the client, resulting in better client and overall network performance.
+ Exactly known upper limit of the needed time. This is important when providing quality of service (QOS).
+ Multiple clients receive a NBA announcement: inactive clients can join a scan. This is useful for scanning in between normal network connectivity, making more management information about the network available.
+ Power-efficient because it is a passive scanning technique.
+ Very fast, resulting in low latencies of client network connections.

Disadvantages:
- Not useful if the connection with the current access point is lost, before the scan can be started. This can be avoided to automatically start a scan if signal strength falls below a certain point. But another scanning technique is still necessary in some cases, if for example the current access point is turned off, and when the client needs to connect to a network for the first time.
Figure 5: Schematic overview of directed probe request
5 NBA reference implementation

5.1 Introduction

I have made a reference implementation on a existing hardware platform. I choose to develop a real implementation above a simulation of the NBA method because:

1. Simulations could not be very representative for the performance in real world conditions.
2. Not much software is available for the simulation of 802.11 networks. This software would have to be adapted for the NBA scanning method and would probably not be less difficult as the adaptation of existing drivers for the NBA reference implementation.
3. A real implementation could show that the NBA method complied with the current 802.11 standard and could be used with currently available 802.11 hardware.
4. The working implementation would of course be more impressive and should lead to an easier adoption of the NBA scanning method as an IEEE standard.

To test the NBA scanning method with 802.11 hardware I needed to implement low-level functions which are not available in the current 802.11 standard. To implement the NBA scanning method I needed to interact directly with the hardware. The problem is that most low-level functions are implemented in firmware on the hardware. All 802.11 hardware has a special controller chip, often a low-power ARM model, which runs the software located in the firmware. This software controls low-level functions like medium access, timing control, message validation and network management. The first three operations especially need a microsecond resolution, thus requiring a dedicated controller. The driver running in the operating system interacts with the controller. The driver only controls the network configuration and is responsible for data transport between the hardware’s memory and the main memory. This data consists of 802.11 data frames that are exchanged between the client and access point. The driver converts these to standard 802 data frames which are presented to the operating system’s network stack.

Because of the dedicated controller most manufacturers design their hardware functionality in firmware. They implement the available modes like access point mode, client station mode and sometimes a monitor mode. Each mode requires more or less functionality in the firmware. An access point firmware needs to manage multiple connections to client stations instead of one on a client station firmware. The monitor mode is a simple mode that can be used for network monitoring. In this mode the hardware forwards all received frames to the driver, but works in a silent mode: nothing can be send out to the network and it cannot join the network.

Fortunately Intersil [11], one of the largest 802.11 chip designers and a co-developer of the 802.11 standard, developed an extra mode, the Host AP mode. This mode implements only medium access and time critical communications like acknowledgement and beacon frames. Intersil developed this mode to allow access point manufacturers to develop more advanced access points with more functionality, like access point management functions. In this mode management frames are not handled by the firmware, but are forwarded to the driver. This mode made it possible for me to implement the NBA scanning method on Intersil’s Prism hardware. It allowed me to model the reference implementation as close as possible to the proposed NBA scanning method.
Another option would be to use network cards with access point firmware with a simple driver. But this would mean that I could not use the special frame types that the NBA method uses. This would require me to implement the method within normal data frames that would need to be filtered on both ends. This would complicate the development and if the method should be adopted as a standard in the future, my own implementation would be incompatible with the standard. The driver that supports this option is the linux-wlan driver from AbsoluteValue Systems [12], commissioned by Intersil for their Prism chipset product development. Developed in 1999, this driver is the oldest open source driver for Linux, but only supports the infrastructure mode with access point firmware, which is not publicly available, and not the Host AP mode.

A final option could be to approach chipset manufacturers for development tools for their firmware. This was not a viable option because I probably would not have the required expertise for these tools. Also this would mean I needed to write my own special drivers for my own firmware. This would all be very time consuming.

I used the combination of Intersil Prism hardware running in Host AP mode together with the open source Host AP mode drivers developed by Jouni Malinen [13]. This driver is widely used by companies, researchers and expert users and has an active mailing list. It is the reference implementation of Wireless Extensions and Wireless Tools, respectively the wireless API and toolset for Linux. Another advantage is a minimal implementation of the Inter Access-Point Protocol (IAPP, IEEE 802.11f) also used by my NBA method.

5.2 The adapted driver

I have implemented a preliminary implementation of the NBA scanning method in the Host AP driver. My primary focus was on the scanning results measured on the client side. The implementation is only a proof of theory and needs further improvements and real world testing for use in deployed wireless networks.

This implementation is modeled for a single usage profile: only one client on an access point using a realtime application such as VOIP that generates a continuous data stream. The client’s access point notices a decrease in signal strength as the client moves through the network. From previous signal strength measurements the access point has calculated that the client is moving in the coverage area of a neighboring access point. Thus the access point instructs the client to start a scan for the other access point. Other usage profiles will need different settings in the NBA scanning method. These different profiles can be implemented following the reference implementation.

The reference implementation features limited functionality because it was only needed for a performance evaluation of the method. This means that the software does not yet make automatic roaming decisions and the timings in the driver could be more advanced. With this I mean that I currently use simple udelay function calls within the kernel. These calls should not be used for these purposes as they block all other software and the kernel from running. The function should not be used for multiple millisecond pauses, but I implemented it after trying everything with schedule_timeout calls. This function, however, does only support a 10 millisecond resolution and does not guarantee a direct return after the timeout. Changing the code to the udelay function improved the accuracy of the measurements, demonstrated in a factor 10 improvement of the standard deviation. Possibly the code can be rewritten to take advantage of Prism hardware’s event functionality; it can be set to give interrupts at specified intervals. The driver can use this interval to be sure a schedule_timeout function returns within a defined short time accuracy. This is very complex and not yet tested if it could work accurately.

Also there is no support for data buffering on the access point if the client is scanning on other channels, while it is still connected to its old access point. Data and authentication forwarding if the client decides to move to the other access point is also not supported. These options can be implemented later on as they have shown to have a positive impact on the client’s network performance in terms of packet loss and TCP throughput [14].
The method is started on the client’s current access point with the use of a Linux wireless utility, named iwlist. It needs three arguments: interface, client MAC address and the other access point’s time offset. The first argument is the wireless network interface the client is associated to. Iwlist forwards this information using a Linux ioctl system call. This allows sending private commands to a driver, which the device driver module should implement [15]. I extended the ioctl command list of the Linux Wireless Extensions, located in the ./include/linux/wireless.h header file, which is in the kernel source tree. From within the Host AP driver I used the standard method to communicate to the Host AP daemon via the second, virtual interface. The Host AP daemon sets up a virtual wireless interface (wlan#ap) for each master device (wifi#), to which the driver forwards all management frames. It may appear strange to start the scan from the command line via the kernel driver back to a user space daemon, but the command line method should not be used later on to start the scan. The decision to start should be implemented in the driver, where all the information about clients’ signal strengths are, or from the Host AP daemon itself if more advanced network management is needed. The other advantage is that the ioctl method now uses my earlier code that received a scan request from the associated client and started the scan from within the driver. This code can still be used, allowing another part of the method be implemented faster. This part from the method is that a client should be able to initiate a scan by itself if it, for example would detect a drop in signal strength from its access point. Using the iwlist and ioctl method allowed me to initiate the scans manually to measure the scan performance. The arguments it uses are also already available to the access point (interface and client MAC address), except for the other access point’s offset. In the manual mode I used the two different commands ntpdate and ntpq to query this offset. The first command, ntpdate, gives the offset after it measures it realtime by starting a time synchronization sequence with the other access point. The “ntpq -p” command queries the local NTP daemon for the last known offset of the other peer. Thus ntpdate should give a more accurate measurement of the offset.

Now the Host AP daemon starts the NBA scanning method. If first needs a neighbors list of other access points. The current implementation reads this from a configuration file together with the other settings. It can be extended to add access points to the neighbors list if other access points announce a client transition from the local access point. This announcement is already implemented by the Host AP daemon using the IAPP method. For now the neighbors list contains only one access point and is send to the client station. This is different from my proposed method. I propose that all other access points in the list are informed first and then the client(s). But my problem with this is that it would add another timing inaccuracy. There was a time inaccuracy on the client, because in the NBA scanning method the client gets the list of neighbors together with their beacon transmission times. These times need to be compared against a clock. I propose for the method to use the time of reception on the client of the scan initiation frame from the current access point. This is because the hardware gives each received frame a timestamp at reception. This timestamp has microsecond accuracy. The problem is that when the access point needs to transmit the times, it does not now the exact time of frame transmission, because the frame is send by the firmware inside the hardware. If the proposed NBA method is implemented the times within this frame need to be updated by the firmware at the actual time of transmission. This is possible but this would need a firmware update for the hardware. I solved the problem by sending a unicast frame with the information to the client. The client’s hardware will send an acknowledgement frame back immediately thus giving the access point a accurate time at reception of the acknowledgement frame. Now both the client and access point have a very accurate time measurement. The client adds a standard time defined in the driver to this time measurement. The client uses this time to know the exact time the other access point should transmit its beacon frame. The current access point also adds the standard time to the time measurement. But now the current access point adds the time offset of the other access point to this time. This time is send to the other access point and is the reference time for the other access point to send the beacon frame.
This complex method allowed me to implement the NBA scanning method very accurately on the current Linux driver and hardware that is available. It is as accurate as the formal NBA scanning method can become if it would be implemented on access point firmware. This is because the only large inaccuracy in both the method proposal and the real implementation is in the inaccuracy of the time offset between access points.

5.3 Time synchronization

This part gives a short overview of how good the NBA scanning method could perform. Not only does the speed of the method depend on the number of frames to be transmitted it also depends on good time synchronizations between access points. This is important because one access point has to tell another access point when to transmit the beacon frame. This time should be the same on both access points, because if it is not the client will listen at the wrong time for the beacon frame from the other access point.

All clocks tick at different speeds, especially the clocks running on cheap hardware like normal access points. From these inaccurate clocks the software needs to determine an accurate time. In my implementation of the NBA scanning method I have used the standard Network Time Protocol [16]. This provides a accuracy on local area networks and high-speed wide area networks of 1 millisecond as described by Mills [17]. I confirmed this accuracy in my measurements if the NTP daemon could ran for more than a day. The use of NTP as time synchronization method could be improved if needed by running NTP servers on the routers and switches. Cisco for example has devices with this functionality [18]. Other methods that provide an even better accuracy have been proposed and implemented for use with real time operating systems and applications [19]. In wireless networks that are deployed over very large distances a wireless backbone network can be used to provide a common time source to all access points. This wireless backbone network can also provide the network connection between the access point and a wired network. Backbone networks that could be used are GPRS, UMTS or the new IEEE 802.16 Wi-Max standard. A wireless backbone network can provide accurate time synchronization between access points because all nodes in the network can receive information on the shared medium at the exact same time. This is also true for multiple clients on a 802.11 network connected to an access point.

5.4 Experimental setup

I have used three embedded systems running 300 MHz processors running Linux 2.4.23. Two where configured for access point mode and one in client station mode. The access points where working on channel 1 and 13 to avoid interference. The beacon frame interval on the access point was lowered to 4 seconds to avoid the interference of beacon frames to a minimum. The Host AP daemon software (hostapd) ran at the highest priority in Linux, above the other import daemon running on the systems: the NTP daemon (ntpd). This resulted in the fastest possible execution of the NBA parts running in the Host AP daemon. The NTP daemon was configured to use a list of 5 primary (stratum 1) time servers on the internet and my provider’s secondary (stratum 2) time server. The three systems where configured in NTP as peers to each other, to synchronize their clocks towards each other. The NTP daemon kept the time on the systems within an offset off 10 milliseconds and with a maximal jitter of 2 milliseconds.

No other clients where active on the access points and the access points did not serve other services. The wired network consisted of a standard 8-port 100Mbs switch, which did not have other high network loads.

The measurements are generated by my implementation of the NBA scanning method. The values of scan time are determined in the kernel driver and printed as debug output. This is captured to a text file used for further processing. Multiple single measurements where validated using a laptop with a wireless Prism network card running in Monitor mode. In combination with the use of Ethereal [20], which could capture all the complete 802.11 frames on the channel, the transmission of NBA frames was verified. Also no other interfering frames where detected.
5.5 Results

Table 3 shows a summary of the measurements. I performed 100 scans with multiple, random seconds between each scan. After a scan the client station returned back to its access point. The client would wait 30 milliseconds before switching to the other access points’ channel. After these 30 ms, which would normally allow the client to keep transmitting data to its current access point, the client switched the frequency and started waiting for the NBA frame of the other access point. The time waited until this frame is showed in the table. The table shows the results of two of these measurements, each using one of the two methods for time synchronization. Column two shows the results if the one outlier at 10878 microseconds in the measurement is left out. If we compare the two last columns we see that the ntpdate method is only slightly more accurate. Figure 6 on the next page shows a plot of the frequency of the measurements according to the time. It shows that the outlier clearly is a singular point. The interesting part of the figure is the time range from 2000 to 3000 microseconds. The majority of measurements is situated in this time range.

<table>
<thead>
<tr>
<th>time synchronization method</th>
<th>ntpc</th>
<th>ntpq 1 filtered</th>
<th>ntpdate</th>
</tr>
</thead>
<tbody>
<tr>
<td>standard deviation</td>
<td>882.3</td>
<td>342.0</td>
<td>317.3</td>
</tr>
<tr>
<td>mean</td>
<td>2697</td>
<td>2616</td>
<td>2639</td>
</tr>
<tr>
<td>minimum</td>
<td>2206</td>
<td>2206</td>
<td>2215</td>
</tr>
<tr>
<td>maximum</td>
<td>10878</td>
<td>5067</td>
<td>4514</td>
</tr>
<tr>
<td>range min-max</td>
<td>8672</td>
<td>2861</td>
<td>2299</td>
</tr>
</tbody>
</table>

Table 3: Measurements of the NBA reference implementation (in µs)

Another measurement I did was to instruct the client to switch to the other access point after it received the NBA beacon frame from the other access point. On the other access point it received another NBA initialization frame, starting a new scan, resulting that the client would switch back after the scan to the original access point. Note that I changed the times of transmission of the NBA beacon frames. The client did not wait 30 ms before switching to the other access points’ frequency but switched immediately after reception of the NBA initialization frame. The waiting time is now larger because the client was instructed this way. The total time is also smaller (from 30+2 to 13ms) because I defined various different times for the NBA beacon transmission in each measurement. This was done to show that the results where real measurements of NBA frames and did not coincide with other frames such as normal beacon frames. The measurement used the ntpdate method for time synchronization. This measurement is plotted in figure 7 and shows comparable results to figure 6. Again we see the majority of measurements is situated in a 1 ms time range. The total range of all measurements is in a 4 ms range, thus twice as large as the previous measurement with ntpdate, as seen in table 3.
Figure 6: Spread of measurements of ntpdate and ntpq.

Figure 7: Spread of measurements with client switching between the access points enabled.
6 Conclusion and future work

6.1 Conclusion

The measurement results of my NBA scanning method implementation show that fast and accurate scans are possible with the use of existing hardware. The time range to let a client scan for one access point on a specific channel can be set to a comfortable size of 5 milliseconds. In this time a client should be successful in 99% of the scans, if the access point is range of the client. If a client needs to switch to another access point because its current access point is almost out of range, it could be possible it needs to scan multiple neighboring access points before it finds an access point with good signal strength. In normal situations the number of neighbors of a typical access point can be between 4 and 8. Thus if a client needs to scan all 8 neighbors the total time the client is scanning for an access point is 40 milliseconds. This is an order of magnitude faster than the current scanning method. How fast the scan can be started is also important. A mobile client can quickly lose connection of the current access point. Thus the access point needs to be able to send the NBA initialization frame to the client as fast as possible. One option is to use the NTP time synchronization system to keep all access points in synchronization. This way the access point does not have to determine the time offsets just before NBA initialization, which could last more than 10 ms for each other access point. The results show that using NTP is almost as accurate. The method should perform better than other faster methods. This is because access points are automatically informed of the unavailability of client stations, while these are scanning. This prevents access points from sending data to these clients, improving overall network bandwidth, while being able to buffer the data improving client network performance. Last but not least this method is a passive method, requiring no active transmission of probe request frames. This is less power-efficient, which is favorable for mobile clients always running on limited batteries. Thus overall the Network Beacon Announcement scanning method performs very fast in scan and has extra advantages over other fast methods.

6.2 Future work

I have shown that the NBA scanning method works on currently available hardware. But the real implementation of the method should be in new firmware versions in old and new hardware. The timings in the method could be implemented more accurate in firmware and it allows the official mechanism of first informing other access points and clients afterwards. Also the SIFS time frame transmission speed of a NBA beacon frame can only be implemented in hardware. Therefore I will submit the Network Beacon Announcement method to the IEEE 802.11r task group, which is recently formed. The group is responsible for the implementation of a fast client handoff method. This method will need a fast scanning phase that the NBA scanning method can provide. The task group can decide to do further extensive testing off the NBA method using new 802.11 firmware.
References


IEEE 802 Wireless World, a information portal sponsored by the 802.11 and 802.15 Working Groups: http://www.802wirelessworld.com/index.jsp

[4] IEEE 802.11 Working Group:
http://grouper.ieee.org/groups/802/11/Reports/tgr_update.htm

[5] H. Velayos; G Karlsson; “Techniques to Reduce IEEE 802.11b MAC Layer Handover Time”; Department of Microelectronics and Information Technology; KTH Technical Report; Stockholm; April 2003; ISSN 1651-7717

[6] M. Ryong Jeong, F. Watanabe, T. Kawahara; “Fast Active Scan for Measurement and Handoff”; IEEE 802.11-03/XXXr0 Submission; DoCoMo laboratories USA; 11 May 2003


8 Appendixes

8.1 Sequence diagram of NBA proposal

![Sequence diagram of NBA proposal](image)

Figure 8: Sequence diagram of the NBA scanning method
8.2 Sequence diagram of implementation

Figure 9: Sequence diagram of the NBA implementation.