Technical Report
urn:nbn:de:gbv:830–tubdok–11642

# Report on Protocols for Resource Constrained Infrastructure Networks

Helge Klimek, Björn Greßmann and Volker Turau
Institute of Telematics
Hamburg University of Technology
Hamburg, Germany

{helge.klimek,gressmann,turau}@tu-harburg.de

July 2012

Institute of Telematics
Hamburg University of Technology
TUHH

# Table of Contents

# TABLE OF CONTENTS

# Chapter 1

# Abstract

This report introduces three protocols for communication in infrastructure networks consisting of a central server, stationary basestations and mobile devices. Mobile devices use IEEE 802.15.4 to communicate with the basestations. Among other things, the proposed protocols differ in their location management and in the routing of downlink and uplink packets. Through extensive simulations the protocols are compared with the focus on latency and packet loss metrics. The strength of each protocol depends on the traffic pattern caused by an application. The report concludes with a recommendation for the usage of each protocol.

# Chapter 2

# Introduction

An infrastructure network consists of multiple stationary *basestations* which communicate wirelessly with *mobile devices* within their communication area, called *cell* [EHD05] (see Fig. 2.1). Wireless communication is only allowed between a mobile device and a basestation, which means that mobile devices do not directly communicate with other mobile devices and basestations do not communicate with other basestations. Communication between a mobile device and a basestation is single-hop. While the mobile devices have limited energy, e.g. are battery-powered, the basestations use a fixed power supply. Additionally, basestations are connected with a central instance, called *server*, via a wired line backend network. As a result, there exist two communication directions in the network: *uplink* from a mobile device to the server and *downlink* from the server to a mobile device. Basestations simply forward data between mobile devices and the server.

In infrastructure networks, it is easy for a mobile device to send a packet to the server, given that the area is completely covered with basestation cells. On the contrary, for sending downlink packets, basestations close to the current position of the target device have to be selected to forward a packet. In a naïve approach, all basestations are selected to relay a downlink packet. However, as mobile devices and basestations use a single, shared medium for communication, this has the drawback of occupying a channel in all cells. Measures to reduce the number of cells in which downlink packets are transmitted require knowledge about the locations of the mobile devices, in order to select basestations on a more fine grained scale for communication. Gathering this location information usually incurs some sort of message exchange. Once this location information is available, it needs to be transported to the entity making the decision to which basestations a message has to be sent. This extra effort creates costs in terms of resource use such as memory, processing power or channel utilization.

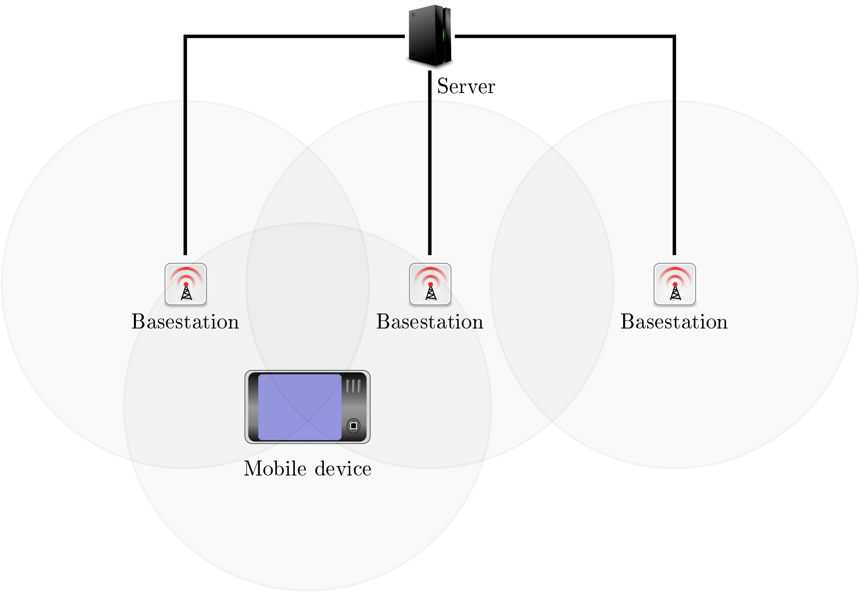

**Figure 2.1:** Infrastructure network: mobile devices communicate wirelessly with a basestation. Basestations are connected via a wired line to a server.

Prominent examples of infrastructure (or cellular) networks are IEEE 802.11 WLAN and GSM. While mobile devices in these networks are relatively powerful in terms of processing power, memory and communication bandwidth, they have higher power consumption and are higher in price than the IEEE 802.15.4 technology used in this paper. Using this energy-efficient, low-bandwidth communication technology, the characteristics and limitations of an infrastructure network with low-budget mobile devices are analyzed. In the following, three protocols with different approaches for selecting basestations to relay data packets to the mobile devices are considered:

- coarse grained Flooding
- medium grained MobileBeaconing
- fine grained BaseBeaconing

By encapsulating optional features of these protocols into functional blocks, their effect can be evaluated. Each of the protocols can be tuned in various ways, leading to a large number of combinations. In order to reduce the number of combinations, the features for the protocols are evaluated in simulations with the focus on packet loss and latency. The simulations are set up and carried out without having a particular application in mind, using a testbed consisting of 18 basestations, arranged into a grid with little overlap of the radio cells. Mobile devices move randomly through these cells. After evaluating the different features, the three protocols are compared against each other in order to analyze their behavior, regarding packet losses and latencies.

The rest of the paper is structured as follows: In Section 2 the network structure is introduced and the protocols are presented. Section 3 presents related work and infrastructure network technologies. Section 4 evaluates the impact of protocol-specific features and sets the basis for a comparison of the protocols in Section 5. The last Section concludes with a summary and an outlook on future research activities.

# Chapter 3

# Network Setup

In the communication endpoints – mobile devices and server – packets are generated and consumed in an application layer. The application layer hands packets to the network layer which sits on top of the IEEE 802.15.4 MAC and PHY layers (see Fig. 3.1). The network layer implements measures for the reliable packet transport and for gathering location information about the mobile devices.

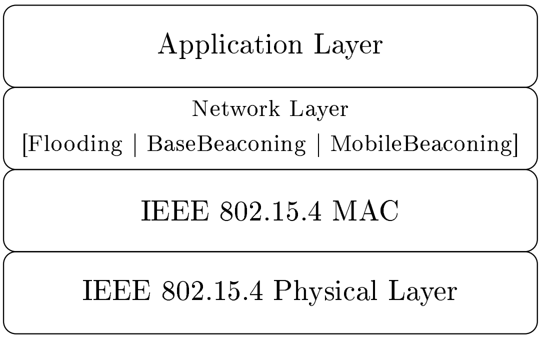

**Figure 3.1:** Protocol stack

To increase the reliability, a retransmission mechanism for uplink and downlink packets is used in all protocols. When a packet has been sent and the sender does not receive an ACK within a specified timeout period, the sender retransmits the packet. *Downlink ACKs* are sent by a mobile device to acknowledge the reception of a downlink packet, *uplink ACKs* are sent either by the server or a basestation to acknowledge the reception of an uplink packet. Receivers have to be able to decide whether packets are new and have to be forwarded to the application layer or whether they have already been received and can be discarded. Sequence numbers in the packets are used to identify duplicates. Checking the order of the sequence numbers alone is not enough, since packets can be delivered out of order, e.g. due to retransmitting single lost packets. Ring buffers are used to store recently received sequence numbers. If a packet arrives, it is checked whether its sequence number is contained in the buffer. If so, the packet has been received before. Otherwise, the packet is regarded as new and its sequence number is inserted into the buffer. In the following, the three protocols are explained in detail.

## 3.1 Flooding

Flooding is a simple protocol requiring minimal logic and buffer space on the basestations, which basically relay packets. Thus, basestations do not send beacons nor do they keep track of which mobile devices reside in their cells. The requirements on software and hardware of the basestations are lower than for the other two protocols.

### 3.1.1 Downlink Packets

Downlink packets are sent to all basestations. The basestations randomly delay packets with a maximum backoff $T_{BBo}$ before sending them over the channel to prevent collisions with the same packet sent by neighboring basestations. Once a mobile device receives a packet, it uses its ring buffer to determine whether it is new, retransmitted or old. If the packet is new or retransmitted, an acknowledgment is sent. There are two alternative policies to send a downlink ACK:

- DLACK = `reliable`: The ACK is sent unicast to the basestation the mobile device received the downlink packet from. It waits for an acknowledgment (called *ACKACK*) for the reception of the downlink ACK from the basestation and will resend its downlink ACK $N_{DlACK}$ times, if it does not receive an ACKACK.
- DLACK = `unreliable`: The ACK is broadcasted, so all receiving basestations forward it to the server. Acknowledgments for the downlink packets are not repeated.

The intention for providing a reliable acknowledgment mechanism is to reduce downlink retransmissions caused by lost downlink ACKs. Downlink retransmissions are sent to all basestations, which in turn will send the packet over the channel using the same shared medium, blocking it for other transmissions. Using retransmissions for downlink ACKs is localized, additionally, downlink ACKs are much shorter than downlink packets so the blockage of the channel is shorter than for a downlink packet. Implementing a reliable acknowledgment requires little buffer space at the mobile device for keeping track of which downlink ACKs have been sent.

### 3.1.2 Uplink Packets

Uplink packets are broadcasted to the base stations, which upon receipt of an uplink packet forward it immediately to the server. After forwarding the uplink packet, the basestations send an uplink ACK to the mobile device themselves. The server is not involved in sending the ACKs. These ACKs are randomly delayed, to avoid collisions with ACKs for the same packet from neighboring basestations.

## 3.2 MobileBeaconing

In MobileBeaconing transmissions are utilized by the basestations to keep track of mobile devices within their transmission ranges. Additionally, mobile devices periodically send beacons, so the basestations receive packets from a mobile device within a predefined interval. For each cell, a history of the last reception timestamps of present mobile devices is managed. If a basestation does not receive a packet (application packet or beacon) within a time span $T_b$ from a mobile device, the according entry is deleted from the history. Downlink packets are sent into every cell having an entry for the destination device in its history. The value of $T_b$ has an impact on the number of basestations that forward a downlink packet. For larger $T_b$, a downlink packet will be transmitted into a higher number of cells. As a result, the communication overhead is increased. On the other hand, $T_b$ has to be long enough, so when sending a downlink packet, the destination device is contained in the history of at least one of the basestations that forward the packet, therefore $T_b$ should be larger than the beacon interval. Two different approaches to manage the history are considered. In the DISTRIBUTED APPROACH, each basestation manages its own history, in the CENTRALIZED APPROACH the server manages a global history.

- DISTRIBUTED APPROACH: Basestations use all incoming packets from mobile devices to manage their history of devices. Beacons are not forwarded to the server. Downlink packets are sent to all basestations. These forward the packets if the identifier of the destination device is contained the history.

- CENTRALIZED APPROACH: A beacon received by a basestation is forwarded to the server. The server maintains a history for every basestation using all packets received from the mobile devices. Downlink packets are sent only to those basestations which have the destination device in their history.

### 3.2.1 Downlink Packet

The distributed approach and centralized approach require a different treatment of downlink packets. Thus, the two cases are discussed separately. In both cases, downlink retransmissions are initiated by the server. The uplink side of MobileBeaconing behaves exactly as the uplink side in the Flooding protocol.

**Distributed Approach:** Downlink packets are sent to all basestations. Upon reception of a packet a basestation checks whether the device is contained in its history. In this case, the packet is sent after a randomized delay proportional to the age of the last packet reception. This is a precaution to avoid collisions and favors basestations with more current timestamps. The delay $t_{delay}$ is calculated as following:

$$t_{delay} = rnd \cdot T_{BBo} \cdot \frac{\Delta t}{T_b}, \tag{3.1}$$

where $\Delta t$ denotes the time since the last packet reception, $rnd$ is a random number in the range $[0 \ldots 1)$, and $T_{BBo}$ is a constant denoting the maximum delay.

If a mobile device receives a new or retransmitted downlink packet, a downlink ACK is broadcasted. Basestations receiving this ACK check whether they have delayed downlink packets in the buffer and cancel their scheduled transmission. Finally, the downlink ACK is forwarded to the server.

**Centralized Approach:** When sending a downlink packet, the server selects all basestations with the destination device in their history. Packets to the different basestations are sent with a randomized delay proportional to the age of their respective timestamps. This delay follows Equation (3.1).

Once basestations receive a downlink packet from the server, they immediately forward it unicast to the mobile device. If the device receives the packet, it checks whether it is new and hands it over to the application layer. If it is new or retransmitted, a downlink ACK is broadcasted. The basestation simply forwards any ACK to the server. When the server receives a downlink ACK, it cancels all delayed scheduled transmissions of the downlink packet for other basestations.

The main advantage of the centralized over the distributed approach is that the server is able to cancel a higher number of downlink transmissions. In the distributed approach, a delayed transmission is only cancelled if the basestation receives a downlink ACK. In the centralized approach all delayed transmissions are cancelled if at least one basestation has received a downlink ACK. In the distributed approach the basestations relieve the server of the book keeping for the different basestations. Additionally, beacons do not have to be forwarded to the server.

## 3.3 BaseBeaconing

In BaseBeaconing, basestations periodically broadcast beacon packets to enable mobile devices to recognize basestations in their vicinity, so they can register to a single basestation. The basestation a device is registered to is called *home station*. The server stores the home stations of the mobile devices and sends downlink packets only to the home station of a mobile device. When a device does not receive a packet from its home station within a maximum time span $T_{regTO}$ – e.g. because it has left the cell – it registers to a different basestation once it receives a beacon.

When an unregistered mobile device receives a beacon, it initializes a registration to the basestation by sending a registration request. These requests are forwarded by basestations to the server, which updates the registration information. Next, the server sends a de-registration notification to the old home station which performs management operations, such as canceling all retransmissions for the mobile device. Additionally, the server sends a registration response via the new basestation to the mobile device. When it receives the response from the requested basestation, that basestation becomes its new home station. If the device does not receive a registration response within a defined time period $T_{regResp}$, it stays unregistered and waits for the next beacon of a basestation to send another registration request.

If FASTBSSWITCH is enabled, the mobile device stores a list of $N_{bs}$ basestations from where it most recently received a beacon. If a device sets its registration state to unregistered and FASTBSSWITCH = `enabled`, the device instantly selects the most recent basestation from its list and sends a registration request in order to reduce the time it is unregistered. Additionally, if the device does not receive a registration response from a requested basestation within the timeout, it immediately selects a different basestation to send a registration request.

### 3.3.1 Downlink Packet

In BaseBeaconing, downlink packets are not sent to mobile devices instantly, but the beacons of the basestation contain a list of device IDs for which downlink packets are pending at the server. After receiving a beacon a mobile device actively pulls the data. This mechanism allows duty cycling at the mobile devices, as they can turn off their transceiver between consecutive beacons of their home station.

When a downlink packet is sent to a mobile device by the server, the server buffers the packet and sends a downlink notification containing the ID of the mobile device to the device's home station. Next, the home station adds the ID of the device to an ID list. Whenever a basestation broadcasts a beacon, this beacon contains a traffic indication map (TIM) with the IDs of the devices for which at least one pending downlink packet is buffered at the server.

Every time a device receives a beacon from its home station, it checks whether its ID is contained in the TIM. In this case, the mobile device sends a pull packet to the home station which removes the ID of the device from its ID list and forwards the pull packet to the server. After the server has received a pull packet from a mobile device, it sends all downlink packets pending in its buffer.

Downlink packets are forwarded unicast by the basestations to the destination mobile device. There is a two-level retransmission scheme in BaseBeaconing: basestations as well as the server perform retransmissions. After $N_{BRt}$ retransmissions from the basestation, the next retransmission at the server is triggered, the server sends at most $N_{SRt}$ retransmissions. Whenever the mobile device receives a downlink packet, it sends an ACK to its home station. When a basestation receives such an ACK, it cancels the retransmission mechanism for the acknowledged packet and forwards the ACK to the server, which then also cancels the retransmission mechanism.

### 3.3.2 Uplink Packet

In BaseBeaconing, there are two alternative policies for sending uplink packets. Uplink packets are either sent unicast to the home station (ULDEST = `home station`) or broadcasted (ULDEST = `broadcast`). The second option increases the reception probability, but also increases the number of uplink ACKs sent by basestations.

When the mobile device is not registered to a basestation, broadcast is used. In case of retransmissions, the uplink packet is either sent to the home station or broadcasted. If the mobile device is unregistered and ULDEST = `broadcast`, the uplink packet is broadcasted, otherwise the packet is not sent. In both cases, the retransmission counter is incremented and after $N_{MRt}$ unsuccessful retransmission attempts, the delivery of the packet fails. The basestation performs a duplicate check upon reception of an uplink packet. If the packet is not a duplicate, it is forwarded to the server. In any case, an uplink ACK is sent to the mobile device.

# Chapter 4

# Related Work

Infrastructure networks usually are assembled from more complex and expensive hardware, like the Global System for Mobile Communication [Sau11, Kö5] (GSM) or IEEE 802.11 WLAN [KR08]. In GSM nationwide hierarchical network consisting of mobile switching centers and transceiver basestations manage the location and route traffic to mobile devices. Even though mobile devices and basestations exchange information constantly, the position of the mobile user is known only to a level of multiple basestations, called location area (LA). Once a call has to be forwarded to a mobile device, the mobile switching center has to page all the cells of a LA, to determine to which basestation the call has to be forwarded to. In the literature, various approaches for mobility management have been proposed for cellular networks [Tab97, Pla94, SK99].

Also WLAN can be used to create infrastructure cellular networks. This is done, e.g. if the coverage area of one access point is too small. The extended service set (ESS) allows to connect multiple access points to a distribution network, which forwards data to the base station a mobile device is registered to. The mobile devices themselves take care of registering to the access point with the strongest signal strength if they switch cells. The optional IEEE 802.11f *Recommended Practice for Multi-Vendor Access Point Interoperability* standard describes how user information is distributed among the access points when a cell switch happens.

IEEE 802.15.4 WPAN [Soc06] is a technology mainly used in the field of Wireless Sensor Networks (WSN), in which nodes equipped with various sensors measure environmental phenomena, as e.g. in habitat or glacier monitoring, and transmit data wirelessly to a data sink. Consequently, if nodes are difficult to access and battery changes come with a lot of effort, the energy-efficiency of IEEE 802.15.4 is utilized to reduce power consumption to achieve long lifetimes. To allow devices to turn to sleep mode and save energy, traffic indicator maps are used in order to notify mobile devices about waiting packets on a coordinator node. Multi-sink infrastructure networks and mobility management in IEEE 802.15.4 networks are rarely discussed in the literature. [EHD04, EhDH04, EHD05] present WiseMAC, a MAC protocol specially designed for the downlink direction in low traffic situations. Handoff is discussed in [FZA$^+$12, CLBR10]. Mechanisms similar to location update and paging are used in conjunction with object tracking [LTL06]. The location of an object can be acquired via queries, which correspond to paging requests in GSM. When an object is tracked by a different sensor than before, an update of the location of the object is performed. While infrastructure networks and cellular network use one-hop communication, mobility management in wireless mesh networks is more challenging as routing and addressing on IP level have to be handled [XW08, Mas11]. In [ZNW$^+$11] the authors survey mobility management solutions for highly dynamic vehicular networks.

## Chapter 5

# Evaluation

In the following different combinations of features of the protocols are analyzed in simulations. The goal is to identify features that reduce packet losses and latencies. In this paper, the number of packet losses denotes the number of created packets handed to the network layer for which no corresponding ACK is received after the maximum number of retransmissions. Besides failed transmissions, buffer overflows are a possible source for packet losses. The latency of a packet denotes the time between handing it to the network layer and the reception of its corresponding ACK. It depends on the number of packets retransmissions until the the sender receives the corresponding ACK. For BaseBeaconing, additional time is necessary in order to inform the mobile device of pending packets at the server, and for the device to request these packets.

The nominal bandwidth of a 2.4 GHz IEEE 802.15.4 channel is 250 kBit/s and the radius of a cell is 47 m given the selected transmission power of 1.78 mW. All devices and basestations use the same radio channel for communication in order to limit the effect on other technologies using the same frequency band and also to reduce the complexity of the system. As this paper focuses on wireless communication, the bandwidth of the wired link between the basestations and the server is assumed to be unlimited. Downlink and uplink packets are created at the application layer and handed over to the network layer with an inter-packet creation time selected randomly from the interval $[0, p_{dl})$ at the server and $[0, p_{ul})$ at the mobile devices. The destination device of a downlink packet is selected randomly.

The simulations use a simplified Random-Waypoint-Model [JM96]. 100 mobile devices move with a constant human walking speed of 1.34 m/s and the pause time after reaching a target position is set to zero. The target positions are created using random number generators with uniform distribution. The simulation time is 500 s. Every simulation is performed four

| Packet Type | Size [bytes] |
|---|---|
| Uplink data | 79 |
| Downlink data | 79 |
| Base Beacon / TIM | 54 |
| ACK | 6 |
| ACKACK | 6 |
| Registration Request | 6 |
| Registration Response | 6 |
| Mobile Beacon | 4 |

**Table 5.1:** Packet sizes

times with different seeds for the random number generator for controling the mobility. Simulation results are averaged over the four simulations runs. The simulations are built on top of the OMNeT++ framework. For the wireless channel, the 802.15.4 PHY and MAC layer from MiXiM are used. A simple pathloss model with $\alpha = 3.5$ and the carrier frequency set to 2.412 MHz is used. The packet sizes are summarized in Table 5.1. The simulation area is 240 m × 310 m. It is placed in a test layout consisting of 18 basestations, arranged in a grid of 3 × 4 basestations (with 6 basestations filling the holes) resulting in total coverage of the area. This layout has been chosen in order to have enough space for mobile devices to move around and cross different cells.

## 5.1 Flooding

The option for the reliable downlink ACK policy is evaluated. The question is, whether this policy significantly reduces the number of downlink packet retransmissions caused by lost downlink ACKs and thereby reduces the communication overhead. Downlink packets are broadcasted in all cells, blocking the channel in the whole coverage area and thereby reducing the total throughput of the network. To answer this question, the server creates downlink packets with varying $p_{dl}$ ranging from 0.0125 s to 1.0 s. No uplink traffic is created in the network.

The results show that with DLACK = `reliable`, the number of retransmissions is lower. For $p_{dl} = 0.025$ s it is reduced from 13.1 % to 8.96 % of the number of downlink packets and for $p_{dl} = 0.2$ s it is reduced from 8.07 % to 7.84 %. The simulations also show that the lower overhead from retransmissions is not outweighed by the costs of additional ACKACKs and retransmissions of downlink ACKs. The number of sent downlink ACKs in both scenarios for DLACK are of the same order for traffic intervals larger than 0.0125 s. In case of high downlink traffic ($p_{dl} = 0.0125$ s), the reliable ACK policy worsens the situation, as many more ACKs and ACKACKs jam the channel. It is important to note, that with the reliable ACK policy, a number of additional ACKACK packets in the order of the downlink ACKs are exchanged.

The results further show, that the better the transmission works, the less beneficial the reliable downlink ACK policy is. For $p_{dl} = 0.025$ s and with DLACK = `unreliable`, an additional 71.57 % of the data volume is added due to broadcasted retransmissions compared to DLACK = `reliable`, for $p_{dl} = 0.2$ s the overhead is still 1.42 % of the data volume with DLACK = `reliable`.

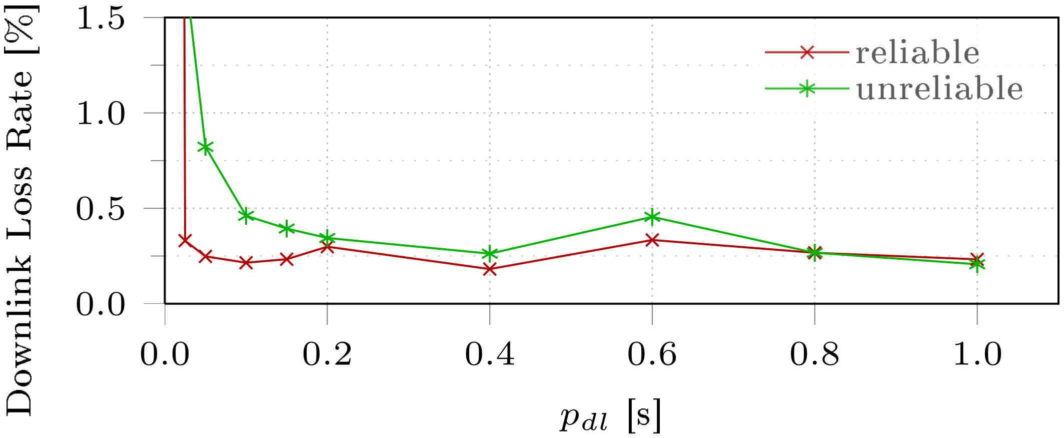

**Figure 5.1:** Flooding: Downlink loss rate using reliable or unreliable ACK policy.

The downlink ACK policy has an impact on downlink loss rates (see Fig. 5.1), which are lower for $p_{dl} \leq 0.08$ s using the reliable ack policy. The rates differ by roughly 0.5 % - 0.1 % and are below 0.5 % for $p_{dl} \geq 0.1$ s. In higher traffic situations, the losses increase drastically to 15 % - 17 % at $p_{dl} = 0.0125$ s. No effect on the downlink latency could be observed and average latencies are about 2.6 s in high traffic scenarios ($p_{dl} = 0.0125$ s). In low traffic scenarios ($p_{dl} = 1$ s) the downlink latency is about 0.2 s.

In summary, using the reliable downlink ACK policy has only limited impact on the downlink losses. In high traffic situations using a downlink interval $p_{dl} = 0.0125$ s without any uplink traffic results in a downlink packet loss rate of up to 15 % to 17 % regardless of whether the reliable ACK policy is used. Its benefit is that it reduces the overhead caused by downlink retransmissions, which are sent in all cells. The reliable downlink ACK policy becomes increasingly beneficial with larger numbers of devices and larger cell numbers in the network.

## 5.2 MobileBeaconing

This subsection analyzes, whether the centralized approach decreases the number of downlink packets transmitted to the wireless channel compared to the distributed approach. For this set of simulations, $p_{dl}$ has been set to 0.5 s to provide a traffic situation that does not overload the network. No uplink traffic is generated. The mobile device beacon interval has been set to 3.0 s providing a suitable value for the given user speed and cell size. The parameter which is varied is the reception time threshold $T_b$ of uplink packets which is used to decide whether a downlink packet is sent into a cell or not.

Using the simulations, the conjecture that using the centralized approach decreases the number of sent packets could be backed. Figure 5.2 shows the number of downlink packets which are sent by basestations to the wireless channel. As expected, the distributed approach leads to a higher number of sent downlink packets. A significant effect on downlink packet latencies or packet losses could not be found.

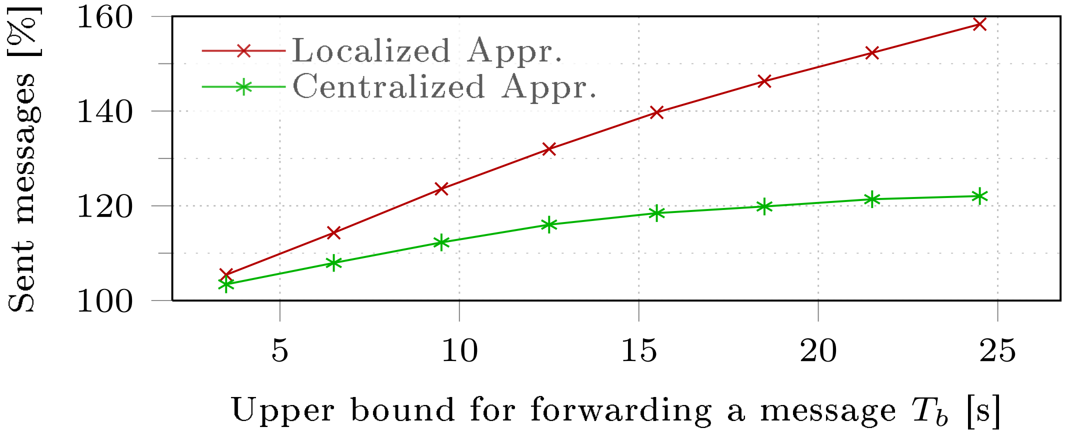

**Figure 5.2:** MobileBeaconing: Downlink packets sent by basestations to the wireless channel in percentage of the number of downlink packets created in the application layer of the server.

Consequently, if the network connection of basestation and server is powerful enough to send data back and forth, the centralized approach is beneficial. It relieves basestations from keeping track of beacons and reduces their complexity.

## 5.3 BaseBeaconing

Referring to Section 3, the impact of two features is evaluated: FASTBSSWITCH and ULDEST. The impact of the sent registration requests and their responses have also been evaluated. Given the speed of the mobile devices and the cell radii, devices registered on average about every 20 s to a new cell. Compared to the simulated data traffic, the effect on the network is negligible.

### 5.3.1 Varying Uplink Traffic

The simulations contained uplink and downlink traffic: the maximum uplink packet generation interval $p_{ul}$ for each device is varied from 0.2 s to 1.0 s, while the server creates downlink packets with a maximum downlink packet interval of 0.1 s with random mobile devices as destination. As in MobileBeaconing, the beacon interval is set to 3.0 s.

In addition to latency and throughput, the impact of the features on the registration processes is analyzed. This includes the total times of devices not being registered and the average duration a device is unregistered. During these times, the server has incorrect information about the location of the device causing downlink packets to be sent to the old home station, leading to channel wastage.

FastBSSwitch = `enabled` reduces the average time it takes a device to register to a new basestation (see Fig. 5.3). For comparison, while in the lowest traffic situation ($p_{ul} = 1$ s) the resulting total time devices were unregistered is about 6.4 % using FastBSSwitch = `enabled`, in the highest traffic situation ($p_{ul} = 0.2$ s) it is about 13.1 %. With increasing traffic ($p_{ul} \leq 0.6$ s), ULDest = `broadcast` increases the average times and also the total times a device is unregistered, with devices not being registered 26.3 % of the time in the worst case at $p_{ul} = 0.2$ s (FastBSSwitch = `disabled` and ULDest = `broadcast`).

Summarizing, looking on the impact on the registration process alone, FastBSSwitch = `enabled` reduces the average time it takes a device to register and the total sum of time a device is unregistered. ULDest = `home station` further improves this metric in higher traffic situations ($p_{ul} \leq 0.6$ s), therefore a combination of these features is best regarding the registration processes.

Using ULDest = `broadcast` reduces the uplink loss rates irrespective of FastBSSwitch (see Fig. 5.4(a)), the standard deviation is below 1 %, for $p_{ul} > 0.6$ s it is below 0.2 %. Using ULDest = `broadcast` increases the number of uplink ACKs due to multiple basestations receiving the packet and responding with an ACK. In situations with high traffic ($p_{ul} = 0.2$ s) it adds 22.9 % of the number of ACKs that are sent in a scenario in which ULDest = `home station`. In a low traffic scenario ($p_{ul} = 1$ s), 12.3 % ACKs are added.

The combination of FastBSSwitch = `enabled` and ULDest = `home station` results in the lowest uplink latencies (see Fig. 5.4(b)). In general, in low traffic situations ($p_{ul} \geq 0.6$ s) all feature combinations behave similar. In mid to high traffic situations ($p_{ul} \leq$

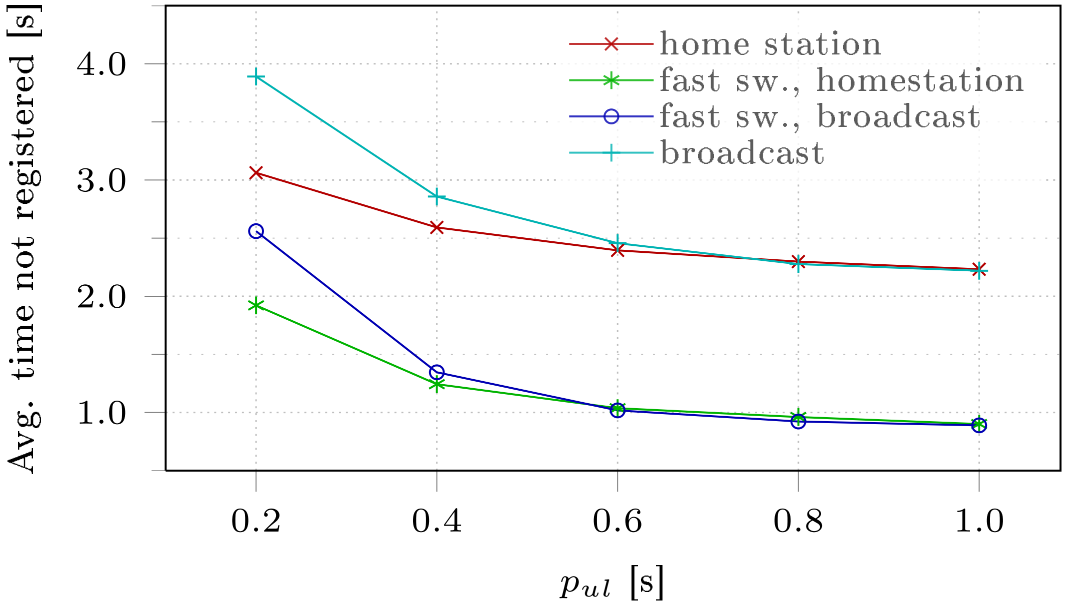

**Figure 5.3:** BaseBeaconing: Average time it takes a device to register to a BS after it has set its state to unregistered.

0.4 s), the two combinations with ULDEST = `broadcast` perform worst (about 1.5 s) regardless of the setting for FASTBSSWITCH.

It was assumed, that the high latency for ULDEST = `broadcast` is due to colliding ACKs in high traffic scenarios. After running simulations with varying ACK backoff values, it showed, that this parameter has the expected effect on the latency of uplink messages, but no significant effect on the uplink losses was found.

Even though the amount of downlink losses is higher in high traffic situations ($p_{ul} = 0.2$ s), if ULDEST = `broadcast`, the total amount of downlink losses remains low, at 5 % to 7 %. With ULDEST = `home station` the losses vary between 2 % to 3 %. In lower traffic situations, all combinations of features perform similar. For $p_{ul} = 0.4$ s, the loss rate is below 0.5 %, in all other situations ($p_{ul} > 0.4$ s) the loss rate stays close to 0 %.

In general, the downlink latency is higher than the uplink latency. Beacons are sent regularly by basestations with a fixed interval. After mobile devices receive a beacon and check their TIM, they request pending downlink messages from the server. Until then, messages are waiting on the server. When devices leave a cell, it takes them $T_{regTO}$ to recognize this, before they register to a new basestation. In between, beacons from the previous basestation cannot be received and checked for pending messages. Additionally, if downlink packets were already requested, the packets are sent in the old cell, causing them to be retransmitted after a timeout.

In all uplink traffic situations (low to high traffic), the performance of the feature sets

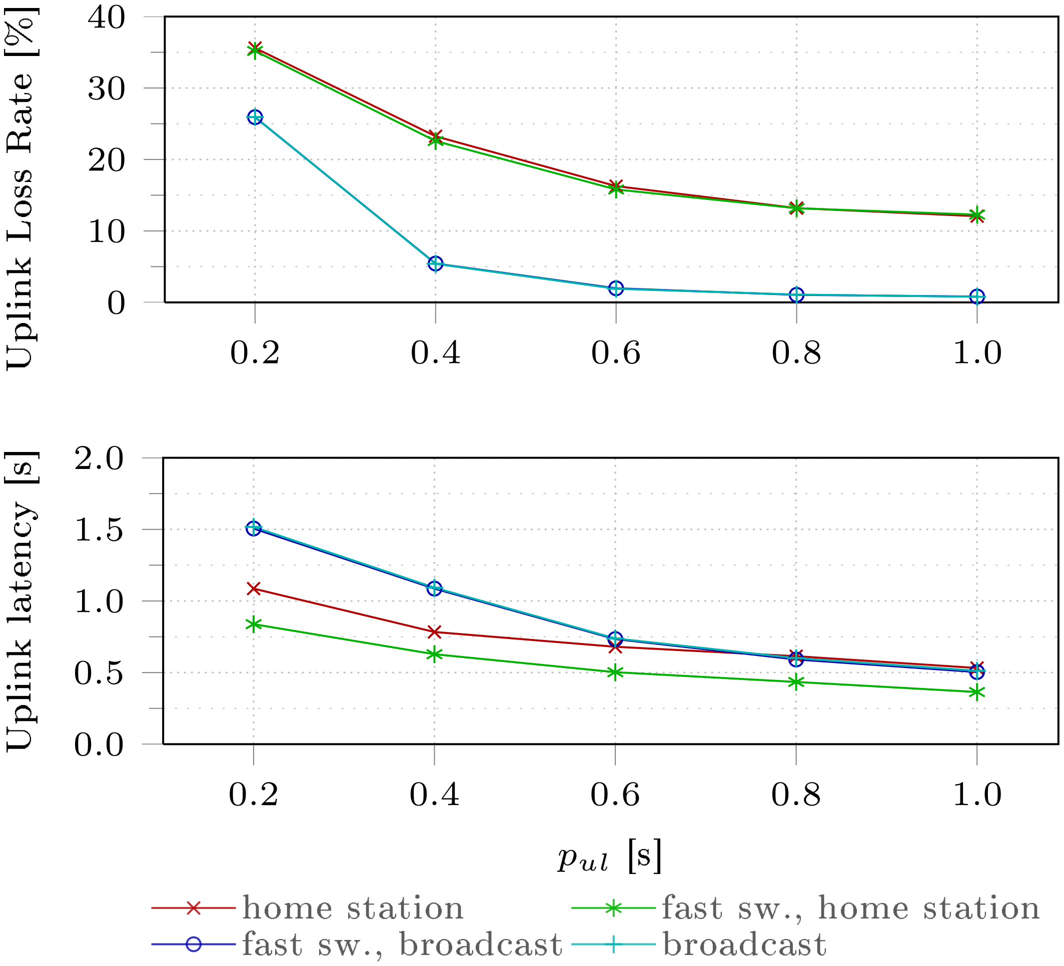

**Figure 5.4:** BaseBeaconing: Uplink loss rates (5.4(a)) and uplink latencies (5.4(b)).

with ULDEST = `home station`, were similar. The latency ranged from about 5 s with $p_{ul} = 1$ s, to 18 s for $p_{ul} = 0.2$ s. While ULDEST = `broadcast` has a similar latency as ULDEST = `home station` in low traffic situations ($p_{ul} \geq 0.6$ s), the latency increases in higher traffic situations with a maximum average of over 30 s.

### 5.3.2 Varying Downlink Traffic

In a second set of simulations, the maximum downlink packet interval $p_{dl}$ has been varied from 0.01 s to 0.6 s. The maximum uplink interval was set to 0.6 s. This experiment verifies the results of the experiments with varying uplink interval and is also used to analyze the impact of varying downlink traffic.

Uplink losses behave as in the previous set of experiments. ULDEST = `broadcast` significantly reduces the uplink loss rate, but increases the number of uplink ACKs due to multiple basestations responding with an ACK. With varying downlink traffic interval, the uplink loss rate (below 2.5 %) and number of sent uplink ACKs remained constant, except for high traffic scenarios ($p_{dl} \leq 0.1$ s) where an increase in the packet loss rate of about 2 % points can be noticed.

The latencies for uplink packets are the lowest when ULDEST = `home station` and FASTBSSWITCH = `enabled` (0.6 s to 0.8 s), matching the results of the previous experiments. If ULDEST = `broadcast` is set, the latency is among the highest (0.7 s to 0.9 s). Here too, the latencies remain relatively constant in mid and low traffic scenarios ($p_{dl} \geq 0.1$ s). A slight increase of about 0.2 s is noticed in high traffic scenarios.

Looking at the registration procedures, the results verify the findings from the experiments with varying uplink traffic, with FASTBSSWITCH = `enabled` reducing the average time and the total time devices are unregistered. For the varied traffic intervals, the values for these metrics stayed constant.

Regarding downlink packet latency, FASTBSSWITCH = `enabled` performs best, regardless of the setting for ULDEST, although the differences are not significant (see Fig. 5.5(b)). The latencies are about 5 s in lower traffic scenarios ($p_{dl} \geq 0.2$ s). Starting from a downlink traffic interval of 0.1 s the latency increases to a maximum of about 25 s at $p_{dl} = 0.01$ s. Average downlink losses are not significantly influenced by the different features (see Fig. 5.5(a)), with the standard deviation being below 1.5 % for $p_{dl} < 0.1$ s. Losses are almost 0 % in low traffic situations ($p_{dl} \geq 0.1$ s), but increase in higher traffic situations up to 39 % to 44 % at $p_{dl} = 0.01$ s. The number of downlink ACKs is almost equal for the different features enabled, implying that the features do not have an effect on the number of downlink ACKs. From the analyzed data, it can be seen that starting from $p_{dl} = 0.1$ s the system starts thrashing with losses and latencies growing rapidly.

In the simulations varying downlink traffic, the tested features have an effect mainly on the uplink direction, while the downlink direction shows similar results for the different sets. The downlink direction maintains a low packet loss rate in low to mid traffic scenarios ($p_{dl} \geq 0.1$ s). As expected, the downlink latencies are high with more than 5 s, due to the beaconing and pull mechanism. If ULDEST = `broadcast`, the uplink packet loss rate with about 2.5 % is the lowest – however, uplink messages sent via broadcast negatively influence the uplink message latencies.

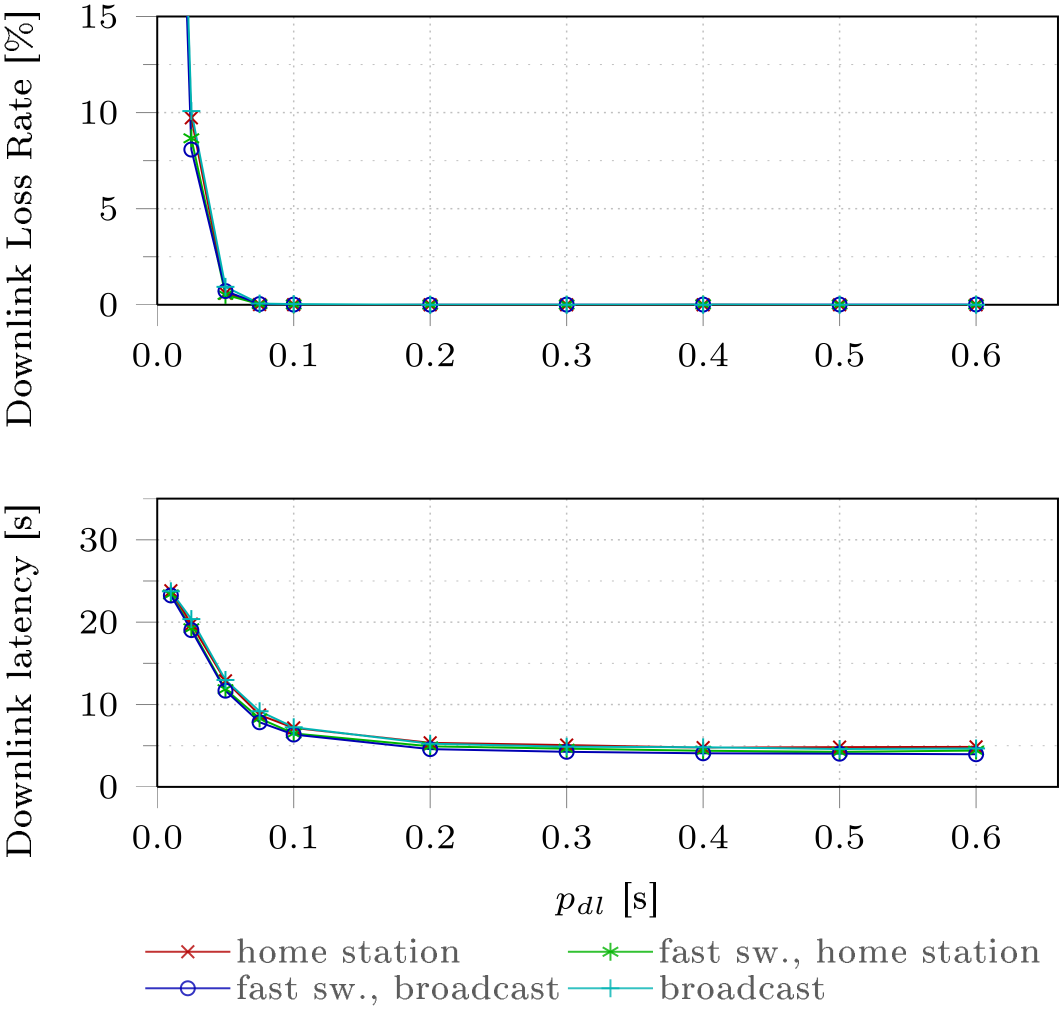

**Figure 5.5:** BaseBeaconing: Downlink loss rates (5.5(a)) and downlink latencies (5.5(b)).

## Chapter 6

# Comparison of the protocols

In Section 5, features have been identified which increase the performance of the protocols. In the next step, the protocols are compared to each other with the identified optimal features enabled. From the design of the protocols, several characteristics can be expected: Using Flooding, latencies in the downlink direction can be expected to be smaller compared to BaseBeaconing, as no artificial delay except a randomized maximum of $T_{BBo}$ at the basestations is added.

For MobileBeaconing, the traffic overhead caused by beacons and therefore the probability of collisions increases with increasing number of mobile devices and decreasing beacon interval. Compared to Flooding, the number of cells a downlink packet is transmitted into is reduced, but the exact number depends on parameters like the speed of the mobile devices or the value of $T_b$. The downlink delays are expected to be in the same order as for Flooding. BaseBeaconing is expected to yield the largest downlink latencies. The downlink loss rates are expected to be lower than in the other two protocols, as a two-level retransmission scheme is used. In contrast to Flooding and MobileBeaconing, in an ideal case a downlink packet is transmitted in only one cell. Another disadvantage of BaseBeaconing is its complexity. While these characteristics result from the protocol design, it is not clear how the different protocols influence downlink and uplink packet loss rates and how the different protocols perform compared to each other.

Some of the expected characteristics are validated by three sets of simulations using the test layout described in Section 3. In the first set, the maximum downlink interval $p_{dl}$ is varied, while the maximum uplink interval for each device is constant. In the second simulation set, the maximum uplink interval $p_{ul}$ is varied and the maximum downlink interval is constant. In these two simulation sets, the values of $p_{ul}$ and $p_{dl}$ are chosen to yield medium traffic situations that do not overload the network. The third set of simulations is performed with varying numbers of devices in low, medium and high traffic scenarios. Each set of simulations is performed with each of the three protocols. For Flooding, DLACK = `reliable` is used, for MobileBeaconing the CENTRALIZED APPROACH and for BaseBeaconing FASTBSSWITCH = `enabled` and ULDEST = `broadcast`.

## 6.1 Varying Downlink Interval

In the first set of simulations, the maximum downlink interval $p_{dl}$ is varied from 0.05 s to 1.0 s and the maximum uplink interval $p_{ul}$ is 0.6 s. The results in Figure 6.1(a) show that the downlink loss rate is relatively constant for each of the protocols. While the downlink packet loss rate is about 6 % for MobileBeaconing ($\sigma \leq 0.75\,\%$), it is about 0 % for BaseBeaconing. This is likely to result from the two-level retransmission scheme providing a more reliable delivery of downlink packets. The Flooding downlink loss rate is about 0.7 % ($\sigma \leq 0.25\,\%$). The uplink loss rates show no significant differences, they are between 1.6 % and 2 % for all protocols.

The latency of the downlink packets varies significantly for the protocols (see Fig. 6.1(b)). Flooding and MobileBeaconing yield average downlink latencies of about 0.6 s, but downlink packets in BaseBeaconing have an average delay of about 4 s at $p_{dl} = 1.0\,\text{s}$ up to about 12 s at $p_{dl} = 0.05\,\text{s}$. This is the result of mobile devices waiting for a beacon and pulling pending downlink data after checking the TIM, while in Flooding and MobileBeaconing downlink data is sent after short delays of maximum $T_{BBo}$s for avoiding downlink collisions at the mobile device. The latencies of uplink packets stay at the same level for all three protocols, at about 0.8 s.

## 6.2 Varying Uplink Interval

In a second set of simulations, the maximum uplink interval $p_{ul}$ is varied from 0.1 s to 1.0 s and the maximum downlink interval $p_{dl}$ is 0.05 s. The downlink loss rate increases with growing values of $p_{ul}$ for all three protocols (see Fig. 6.2(b)). While BaseBeaconing has the lowest loss rates in low traffic situations ($p_{ul} \geq 0.6\,\text{s}$), in higher traffic situations the loss rates increase nearly to the level of MobileBeaconing at $p_{ul} = 0.1\,\text{s}$. MobileBeaconing has the highest loss rates, while Flooding comes with the lowest downlink loss rates. For all protocols, the standard deviation is below 1.1 %.

For low traffic ($p_{ul} \geq 0.6$), the uplink loss rates are about 2 % for each of the protocols (see Fig. 6.2(a)). They begin to increase to rates of about 53 % for MobileBeaconing and

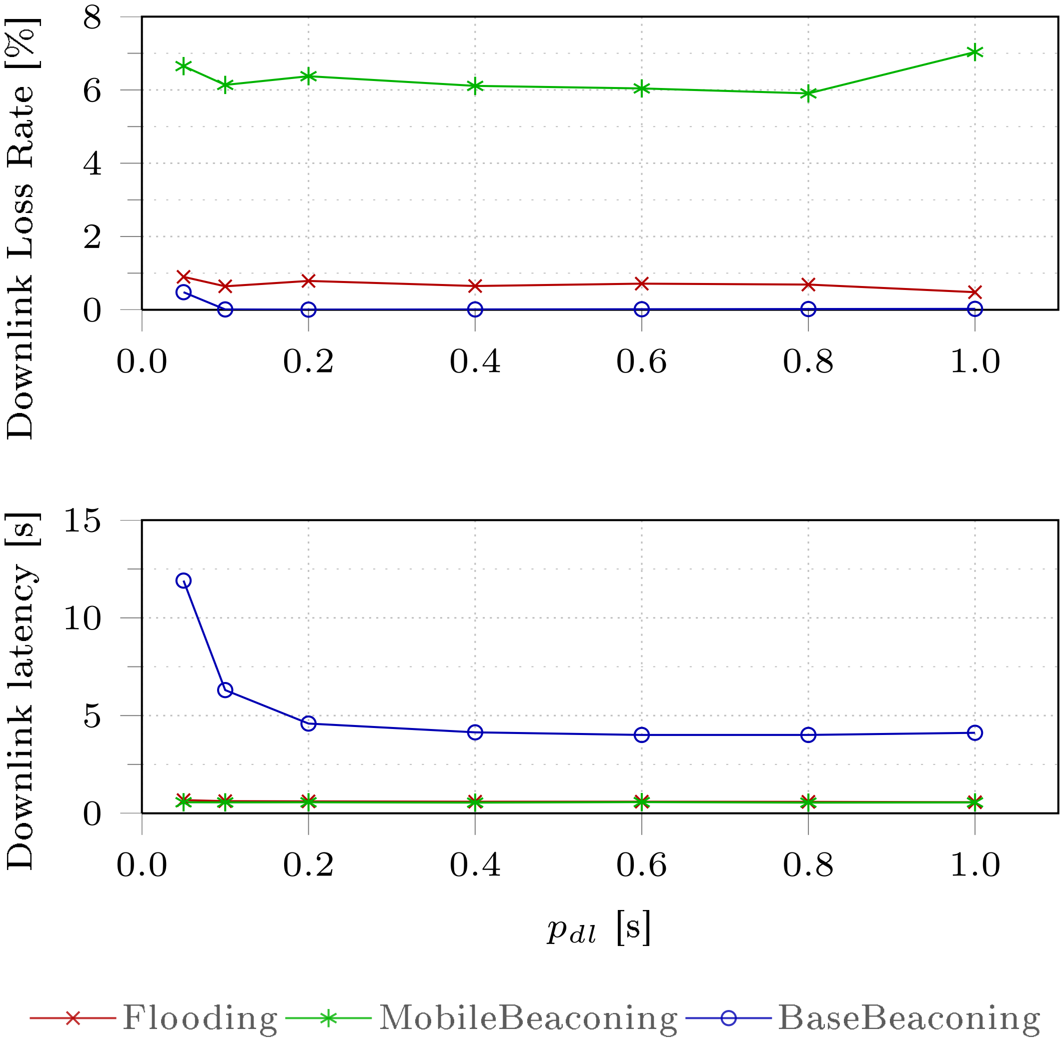

**Figure 6.1:** Comparison of downlink loss rates (6.1(a)) and downlink latencies (6.1(b)).

BaseBeaconing at $p_{ul} = 0.1$ s. For Flooding, the uplink loss rates are higher, and increase up to rougly 65 % at $p_{ul} = 0.1$ s. For all protocols, the standard deviation was below 1.5 %. Looking at the downlink latencies (see Fig. 6.2(c)), Flooding and MobileBeaconing perform best with latencies of roughly 0.5 s to 2.2 s, the values for BaseBeaconing are higher, ranging from about 8 s to about 39 s. Uplink latencies of MobileBeaconing and BaseBeaconing are close, ranging from 0.5 s to 1.5 s ($p_{ul} = 1.0$ s, . . . , 0.1 s). For Flooding, the latencies are slightly higher, ranging from 0.5 s to 1.75 s.

| Scenario | $p_{dl}$ | $p_{ul}$ |
|---|---|---|
| Low traffic | 0.125 s | 0.8 s |
| Medium traffic | 0.1 s | 0.6 s |
| High traffic | 0.075 s | 0.4 s |

**Table 6.1:** Traffic scenarios

## 6.3 Varying Number of Mobile Devices

In a third set of simulations the number of mobile devices is varied from 50 to 250 in steps of 25 to compare the performance of the protocols with increasing number of devices. For this comparison, three different traffic scenarios with fixed packet intervals $p_{dl}$ and $p_{ul}$ are used (see Table 6.1), resulting in scenarios in which the total downlink traffic volume in the network is constant and the uplink traffic volume increases with the number of devices.

The results show that BaseBeaconing has the lowest and MobileBeaconing the highest downlink loss rates for all numbers of devices, in all scenarios. The results for the high traffic scenario in Figure 6.3(a) show that the downlink loss rates for BaseBeaconing increase slower than for the other two protocols with increasing device number larger than 150. The downlink loss rates for BaseBeaconing are below 15 %, while Flooding and MobileBeaconing have maximum loss rates of about 30 % and 42 %, respectively, with $\sigma \leq 1.2\,\%$. The uplink loss rates are similar for all protocols in the low and the medium traffic scenarios, while in the high traffic scenario Flooding has higher uplink loss rates with an increasing number of devices (see Fig. 6.3(b)) ($\sigma \leq 1.5\,\%$). The uplink and downlink latencies are below 3 s for Flooding and MobileBeaconing in all scenarios with all numbers of devices. The uplink latencies of BaseBeaconing are in the same order, but the average downlink latencies are larger, with an increase of up to 48 s in the high traffic scenario and 250 devices.

BaseBeaconing scales best with respect to the total number of mobile devices and downlink losses. Downlink packets are sent to a single basestation, which conserves channel capacity. The added overhead due to registration requests and beacons has no substantial effect, since their number is small compared to the data traffic. For Flooding, the effective number of downlink packets determines how many mobile devices the network can support. In case the server sends downlink packets seldomly, packet retransmissions can be tolerated. If larger numbers of devices or higher downlink traffic occur, retransmissions choke the network. This is expected to happen regardless of the number of basestations, since downlink packets are transmitted into all cells. Using MobileBeaconing, the server sends downlink packets to multiple basestations, additionally all mobile devices regularly send beacons. The total number of mobile devices, and thus the number of beacon emitters, and the density of mobile devices

| | **Flooding** | **MobileBeaconing** | **BaseBeaconing** |
|---|---|---|---|
| **Varying Downlink Interval** | | | |
| DL Loss Rate | ○ | ⊖ | ⊕ |
| UL Loss Rate | ⊕ | ⊕ | ⊕ |
| DL Latencies | ⊕ | ⊕ | ⊖ |
| UL Latencies | ⊕ | ⊕ | ⊕ |
| **Varying Uplink Interval** | | | |
| DL Loss Rate | ⊕ | ⊖ | ○ |
| UL Loss Rate | ⊖ | ⊕ | ⊕ |
| DL Latencies | ⊕ | ⊕ | ⊖ |
| UL Latencies | ○ | ⊕ | ⊕ |
| **Varying Numbers of Mobile Devices in High Traffic** | | | |
| DL Loss Rate | ○ | ⊖ | ⊕ |
| UL Loss Rate | ○ | ⊖ | ⊖ |
| DL Latencies | ⊕ | ⊕ | ⊖ |
| UL Latencies | ⊕ | ⊕ | ⊕ |

**Table 6.2:** Summary of the results in Section 6

is expected to be a crucial factor for this protocol, as the uncoordinated beacons of the mobile devices lead to many collisions. As a result, in the given scenarios MobileBeaconing performs worse than Flooding in the downlink direction.

In summary (see Table 6.2), the results show the expected behavior of large downlink latencies for BaseBeaconing, which increase to a multiple of the beacon interval in high traffic situations. Flooding results in lower latencies. BaseBeaconing yields the lowest downlink loss rates, except for in the simulations with varying uplink interval, Flooding outperforms BaseBeaconing in high uplink traffic situations. Nevertheless, in these situations, Flooding has the highest uplink loss rate. Regarding downlink losses, MobileBeaconing performs worst in all sets of simulations. BaseBeaconing scales best with respect to the number of mobile devices and downlink losses. MobileBeaconing showed the worst performance and Flooding is in between.

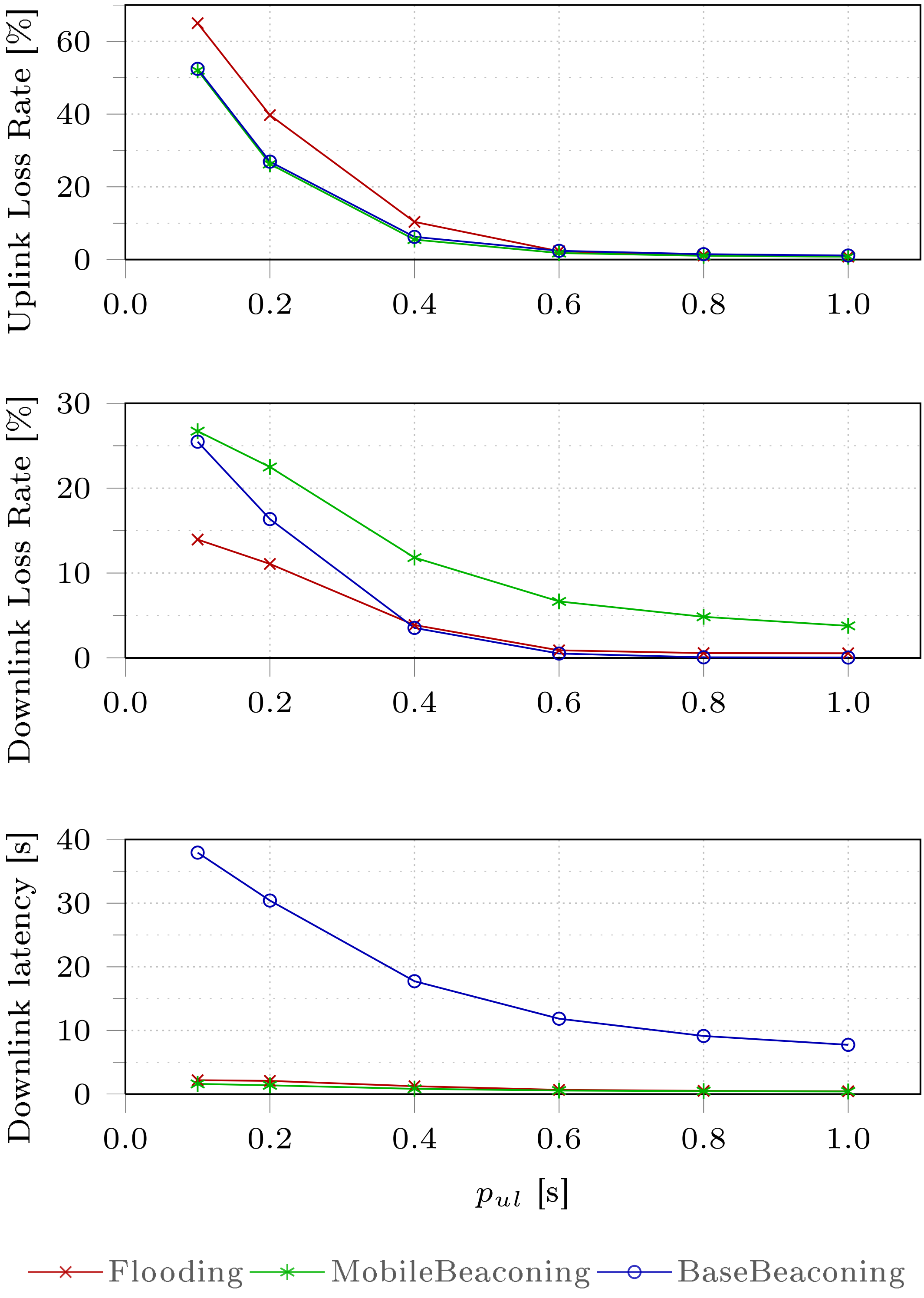

**Figure 6.2:** Comparison of uplink loss rates (6.2(a)), downlink loss rates (6.2(b)) and downlink latencies (6.2(c)).

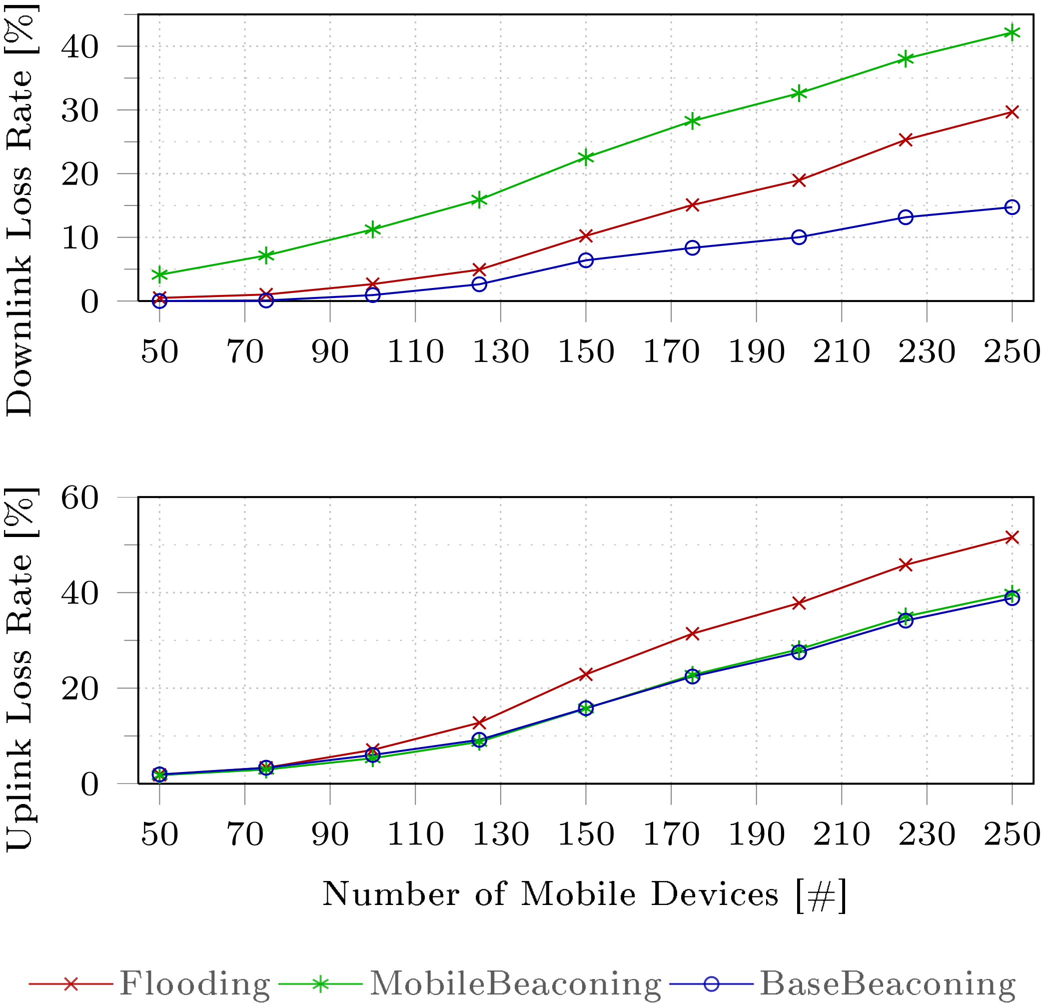

**Figure 6.3:** Downlink loss rate (6.3(a)) and uplink loss rate (6.3(b)) of the protocols in the high traffic scenario.

Chapter 7

# Summary and Outlook

The protocols were designed allowing multiple options and possible configurations. When analyzing these protocols, this leads to a large number of combinations of settings. The simulations described in this paper reduce the combinatorial configuration space by providing a first estimation which features have a positive impact on latency and throughput. For this, different configurations of the same protocol were compared in simulations with a simple basestation layout.

The results show that BaseBeaconing is suitable for applications that expect a higher ratio of downlink traffic than uplink traffic. These applications benefit from the relatively low packet loss rates, but need to take a high delay into account. Flooding is suitable for applications that have low traffic requirements, require low latencies and can deal with downlink packet losses. Although, MobileBeaconing does not perform especially good in any of the simulations, it might find its use in protocols that use the mobile beacons for receiver initiated polling of downlink data.

More fine grained simulations of different traffic volumes, traffic patterns as bursty or regular traffic and varying densities and total numbers of mobile devices have to be performed. Other environmental parameters such as speed of the mobile devices, degree of overlap of the cells and total number of cells have an impact on the performance, too. As these simulations can not be made for general purpose scenarios, a real-world example scenario will be considered. Regarding MobileBeaconing and BaseBeaconing, the effects of changing the beacon intervals, speed of mobile devices and cell radii are also connected. Especially for BaseBeaconing, the beacon intervals are part of the downlink latency.

# Bibliography

[CLBR10] Octav Chipara, Chenyang Lu, Thomas C. Bailey, and Gruia-Catalin Roman. Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 155–168, New York, NY, USA, 2010. ACM.

[EHD04] A. El-Hoiydi and J.-D. Decotignie. Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks. *Computers and Communications, IEEE Symposium on*, 1:244–251, 2004.

[EHD05] Amre El-Hoiydi and Jean-Dominique Decotignie. Low power downlink mac protocols for infrastructure wireless sensor networks. *Mob. Netw. Appl.*, 10:675–690, October 2005.

[EhDH04] A. El-hoiydi, J.-D. Decotignie, and J. Hernandez. Low power mac protocols for infrastructure wireless sensor networks. In *In Proceedings of the Fifth European Wireless Conference*, pages 563–569, 2004.

[FZA+12] Hossein Fotouhi, Marco Zuniga, M$#225;rio Alves, Anis Koubaa, and Pedro Marrón. Smart-hop: a reliable handoff mechanism for mobile wireless sensor networks. In *Proceedings of the 9th European conference on Wireless Sensor Networks*, EWSN'12, pages 131–146, Berlin, Heidelberg, 2012. Springer-Verlag.

[JM96] David B. Johnson and David A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[Kö5] Axel Küpper. *Location-Based Services: Fundamentals and Operation*. Wiley, October 2005.

[KR08] James F. Kurose and Keith W. Ross. *Computer Networking - A Top-Down Approach*. Pearson Educational International, 2008.

[LTL06] Chih-Yu Lin, Yu-Chee Tseng, and Ten H. Lai. Message-efficient in-network location management in a multi-sink wireless sensor network. In *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing -Vol 1 (SUTC'06) - Volume 01*, pages 496–505, Washington, DC, USA, 2006. IEEE Computer Society.

[Mas11] K. Mase. Layer 3 wireless mesh networks: mobility management issues. *Communications Magazine, IEEE*, 49(7):156 –163, july 2011.

[Pla94] D. Plassmann. Location management strategies for mobile cellular networks of 3rd generation. In *Vehicular Technology Conference, 1994 IEEE 44th*, pages 649 –653 vol.1, jun 1994.

[Sau11] Martin Sauter, editor. *Grundkurs mobile Kommunikationssysteme*. Vieweg+Teubner, 2011.

[SK99] Atif A. Siddiqi and Thomas Kunz. The peril of evaluating location management proposals through simulations. In *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, DIALM '99, pages 78–85, New York, NY, USA, 1999. ACM.

[Soc06] IEEE Computer Society. Ieee standard 802.15.4, September 2006.

[Tab97] S. Tabbane. Location management methods for third generation mobile systems. *Communications Magazine, IEEE*, 35(8):72 –78, 83–4, aug 1997.

[XW08] Jiang Xie and Xudong Wang. A survey of mobility management in hybrid wireless mesh networks. *IEEE Network*, 22(6):34–40, 2008.

[ZNW$^+$11] Kun Zhu, Dusit Niyato, Ping Wang, Ekram Hossain, and Dong In Kim. Mobility and handoff management in vehicular networks: a survey. *Wireless Communications and Mobile Computing*, 11(4):459–476, 2011.