Institutionen för datavetenskap
Department of Computer science

Thesis paper

Simulation of Scalable Streaming Protocols using Wireless Multicast/broadcast

by

Christoffer Johansson, Martin Sjödin Jonsson

LIU-IDA/LITH-EX-A--16/001--SE

****-**-**

Linköpings universitet

Linköpings universitet
SE-581 83 Linköping, Sweden

Linköpings universitet
581 83 Linköping

Linköpings universitet
Institutionen för datavetenskap

Thesis paper

Simulation of Scalable Streaming Protocols using Wireless Multicast/broadcast

by

Christoffer Johansson, Martin Sjödin Jonsson

LIU-IDA/LITH-EX-A--16/001--SE

****-**-**

Handledare: Niklas Carlsson
Examinator: Nahid Shahmehri

Abstract

In a society where the popularity of mobile video streaming increases, the need to understand the affect on the power consumption on mobile devices increases. This thesis looks into which broadcasting protocol to use when trying to optimize the energy consumption and startup delay while receiving a video from a transmitting source using scalable streaming protocols. This thesis evaluates four different scalable streaming protocols and how much power mobile clients consume when receiving the video over different number of channels and how different interfaces such as 3G and WiFi affect the power consumption. The protocols are simulated using our own written simulator and a tool called EnergyBox, which simulates both 3G and WiFi connections. The thesis also compares the protocols with each other. Based on these simulations and comparisons we conclude which protocol of these four provides the best energy-delay tradeoff.

Acknowledgments

First and foremost we would like to express our gratitude to our supervisor Niklas Carlsson, for answering all of our questions and providing us with useful advice when conducting this study. Furthermore, we would like to thank Ekhiotz Jon Vergara for answering our questions regarding EnergyBox.

Contents

List of Figures

List of Tables

1 Introduction

It is getting more and more common for video services to find their way to portable devices. However, video streaming is putting a strain on most phones or tablets as these services are high consumers of bandwidth and computing power. One of the big power consumers is the transmission between a base station and the mobile device. There are certainly ways to make the receiver work more efficiently, for example allowing the receiver to rest between receiving bursts of data. Mobile devices is still a growing area where more than half a billion devices were activated for the first time during last year. Furthermore, these devices are responsible for a huge growth in data traffic, with video traffic being responsible for the largest increase in data values. For example, in 2015, video traffic in mobile networks were accounting for 55% of all network traffic done by these kinds of devices [5].

1.1 Broadcast

Broadcasting is a technology that allows one sender to deliver content to several mobile devices and could be used for sending live television through Long-Term Evolution (LTE) based mobile networks [11]. It is most easily understood as a technology that allows you to efficiently send from one sender to several receivers. Compared to traditional broadcast used in Television distribution, these old systems might require up to three times the amount of frequency spectrum to cover the same area as a system broadcasted using the LTE network and more modern protocols. Starting with LTE, mobile networks has become fast enough to handle high quality video transfers. This, combined with the fact that modern protocols allows distributors to allocate frequencies dynamically has created these big advancements in frequency efficiency. If broadcasting is used today in LTE networks it is mostly using the same technique as traditional TV which means that all TV channels are allocating their frequencies all the time even when no-one are watching them. In these cases the advantages are minimised compared to more modern solutions as Evolved Multimedia Broadcast Multicast Services (eMBMS) which will only broadcast to channels that actually got viewers and allow the additional frequencies to be used for other purposes.

1.2 Multicast

Efficient information distribution is optimal during any scenario, making it important to use the correct protocol when it comes to disseminating information. Multicasting is a one-to-many or many-to-many distribution system that only sends data and information to receivers who are listening for that specific information and if there are no-one listening, the data is not transferred.

When multicasting over LTE networks, using eMBMS is an efficient way to do so. eMBMS is particularly efficient when the receiver has a temporary location and is often on the move, e.g., mobile devices [4]. The most distinct difference between multicast and regular broadcasting is that multicasting has intended receivers, while regular broadcasting sends data to anyone who listens to the transmission.

1.3 Motivation

Understanding when to use which protocol for different occasions is important when trying to optimize the power consumption while broadcasting data to clients. Knowing which protocol that is the most energy efficient could be used on a larger scale, such as the mobile 3G network when distributing large amount of data to clients.

1.4 Aim

In a disaster scenario it is important to be able to disseminate large amount of data and information in an efficient way. The purpose of this thesis project is to evaluate the effectiveness of using different broadcast protocols to deliver data to a large number of wireless devices. This report is also looking into making a model for how to efficiently make the radio transmitter rest when it is not actively retrieving data on all available channels. This when streaming video through a multicast solution.

1.5 Research Questions

- How does different broadcasting protocols affect the energy consumption with mobile units that receive data?
  - How does the amount of channels used to distribute the data affect this?
  - Does the startup delay have any influence on the power usage?
- How does the different protocols perform in comparison to each other?
- How does power consumption differ when using 3G versus WiFi?

1.6 Limitations

This thesis only focus on evaluating the receiver and investigate the retrieved data with the receiver, such as used bandwidth and buffer constraints. Another delimitation that has to be taken into account is that we are not able to evaluate each and every broadcasting protocol there is. Thus, we pick four protocols that have proven valuable in other contexts [7].

2 Background

2.1 Periodic Broadcast Protocol

Periodic broadcast protocols [10] broadcasts videos periodically, meaning that a new stream is commenced every S minutes, where the video is transmitted from the host to the receiver. The interval in which the streams are started are called *batching interval*. This technique of data transmission is also called *data centered* because the server channels are dedicated to the data that is being sent, rather then focused on the user.

2.2 Pyramid Broadcasting

Pyramid broadcasting (PB) [10, 7, 15] was introduced in 1995 to reduce the service latency created by earlier periodic broadcast protocols. These earlier protocols broadcasts a video every batching interval, making increases in the server bandwidth the only way to improve service latency.

Figure 2.1 illustrates the operation of Pyramid Broadcast, where each video file is partitioned into n segments, of increasing sizes. The protocol then divides the available bandwidth on the server into n equally sized channels, over which these segments are transmitted. Channel number i is used to broadcast segment i of all the videos, and these segments are sent in the correct order. Since the first segments are a lot smaller then the remaining segments, these earlier segments can be broadcasted more frequently on their respective channel, thereby using the bandwidth more efficiently. On the contrary, pyramid broadcasting puts high demands on the receivers equipment. Clients will have to be able to receive at least two channels at the same time, meaning that the necessary bandwidth with the receiver has to be very high. The client will also be needed to store a large amount of data being transmitted early in the broadcast, more particularly the last two segments of the video and these two segments account for almost 80% of the total file size.

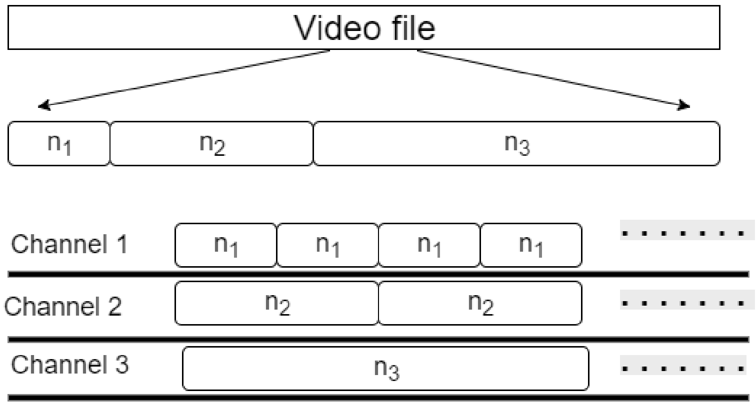


Figure 2.1: Pyramid broadcasting protocol

2.3 Permutation-based Pyramid Broadcasting

Permutation-based pyramid broadcasting was introduced in 1997 [10, 7, 15] to greatly reduce the high equipment requirements with the client that PB created. The method they used to reduce the requirements was to divide each channel into k subchannels and then even out the starts of segments on these subchannels. By doing this they were able to prevent the client from having to receive data from more than one channel at the time, and thereby reducing the amount of bandwidth used by the client. This decrease in used bandwidth also led to a decrease in needed storage on the receiving unit, in fact just a third of the storage required when using PB.

2.4 Optimized Periodic Broadcasting Protocol

In optimized broadcasting protocol (OPB) [8, 14, 2] the video file is divided into N segments. Where the size of each segment is determined by the bandwidth size of each channel. The size of each segment is also optimized such that the next segment s is downloaded just in time for when the playback of segment s-1 is done. This can be observed in Figure 2.2 where it also can be observed that the segments are transmitted one per channel and the segments are repeatedly transmitted at the rate r times the media playback rate. Any client who requests the file immediately starts listening to the first channel and as soon as the segment is downloaded, the client commences playback. Simultaneously, the client listens to the channel that transmits the next segment. This makes the channels usage very efficient thought out the transmission.

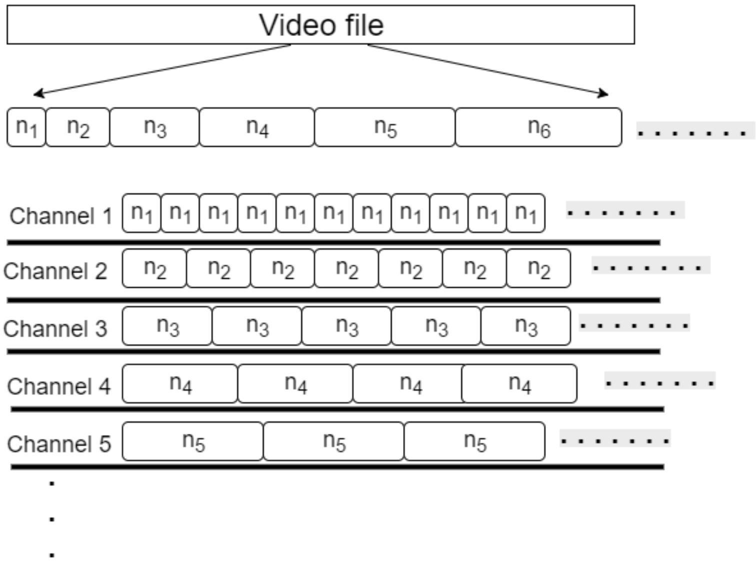


Figure 2.2: Optimized periodic broadcasting protocol

2.5 Fast Broadcasting Protocol

Fast broadcasting (FB) [7, 15] divides each video object into n equally sized segments and transmits these on a predetermined amount of channels. Figure 2.3 illustrates how each video file is broken up into smaller segments. With FB, the data is transferred in parallel by using as many channels as are available with every increasing batch sizes. How the data segments are structured can be observed in the series below,

$$\{1, 2, 4, 8, 16, 32, 64, ...\}.$$

FB follows this set to determine the amount of segments that will be broadcasted on each channel. In other words, FB uses many more segments than for example SB. This leads to a decrease in waiting time for each client. The disadvantage with this protocol is that the client has to receive data from all channels at once, which leads to a much higher transfer rate per channel and the client also has to store up to half of the videos locally. Although there is a mode where clients only receive data from one channel at the time and this mode does not require any local storage, the maximum waiting time for this mode is half the duration of the video. This amount of waiting time is most likely to long for any video popular enough to be broadcasted.

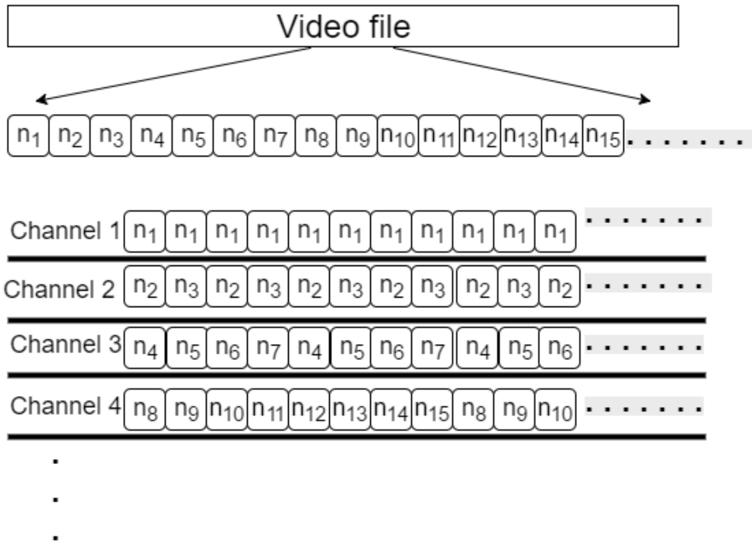


Figure 2.3: Fast broadcasting protocol

2.6 Skyscraper Broadcasting

Skyscraper broadcasting (SB) [7, 16, 15] was introduced in 1997 for the purpose of solving problems found in PB and Permutation-Based pyramid broadcasting. In the first case related to size on disk required for saving each packet and in the other the startup delay, instead of using different sized segments. It is instead packaging everything as equally big chunks of data. This can be observed in illustration 2.4. The segments are then structured using the following series: n segments $S_1, ..., S_n$.

$$\{1,2,2,5,5,12,12,25,25,52,52..\}.$$

Channel 1 will only transfer segment 1 repeatedly, while channel 2 will transfer S_2, S_3 repeatedly, channel 3 will transfer S_4, S_5 and channel 4 will transfer S_6, S_7, S_8, S_9 ,S_{10} and so on. This series grows slower than the ones implemented in the other pyramid based protocols which helps in avoiding getting overly huge chunks at the end. There was also an update to the protocol in 1998 in-order to allow for dynamic allocations of channels and for a better scheme of channel-segment allocation. But as it is optimized for client usage, there is a waste for the broadcasting server at a certain stage as segments are re-sent more often then necessarily, up to tree times at S_{15}. This is a potential waste of bandwidth for the server.

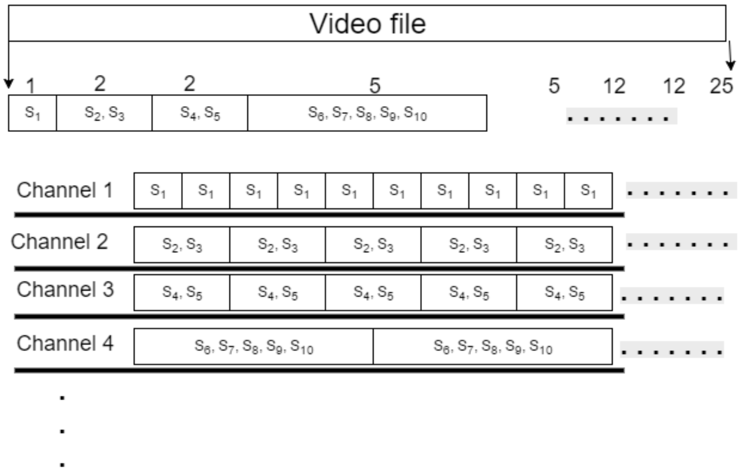


Figure 2.4: Skyscraper broadcasting

2.7 Harmonic Broadcasting

In Harmonic broadcasting [7, 16, 6, 15] each video is divided into N equally sized segments S_1, S_2,..., S_N, just like pyramid broadcasting. But, unlike pyramid broadcasting, each segment S_i is later divided into i subsegments $S_{i,1}$, $S_{i,2}$,..., $S_{i,i}$ of equal sizes. Thereby, segment S_1 has one subsegment $S_{1,1}$, S_2 has two subsegments $S_{2,1}$ and $S_{2,2}$ and so forth. These subsegments are later transmitted on N channels where all the subsegments of segment S_i are broadcasted on channel C_i, sequentially and periodically. The bandwidth on each channel decreases and if the bandwidth required for channel C_1 is the data consumption rate b of the video, the bandwidth required for channel C_i is $\frac{b}{i}$. Therefore, the total bandwidth required for N channels is:

$$\sum_{i=1}^{N} \frac{b}{i}.$$

Later, when the client is ready to consume S_i, the client will receive i-1 subsegments from that specific segment, and the last data slot of this segment can be received during the segments playout time.

2.8 Energy Model

To model the energy usage on the mobile clients, we use the simple energy model in EnergyBox [17], which captures the basic energy states of the receiving interface, including the on/off state. EnergyBox consists of two possible energy modes where it is possible to analyze the power consumption of 3G and WiFi. The energy consumption of these two settings are captured by a finite state machine taking in different parameters to simulate different states for different systems.

3G model

The 3G model of the EnergyBox state machine is divided into three states: Idle or Paging Channel (PCH), Forward Access Channel (FACH) and Dedicated Physical Channel (DCH) [17]. Since the power consumption is similar for both Idle and PCH states compared to FACH and DCH, these are grouped together for simplicity and easier configuration. The system has been simplified in order to retrieve the power levels of each profile, where the simplification consist of making each state fixed to a value corresponding to the average power at that state. A parametrised finite state machine is used to capture the energy consumption of 3G and to later simulate the inactivity timers, the Radio Link Control (RLC) buffers and a low activity mechanism, everything in a packet-driven manner. Figure 2.5(a) shows the states of the 3G model and the state transitions we use in our simulations.

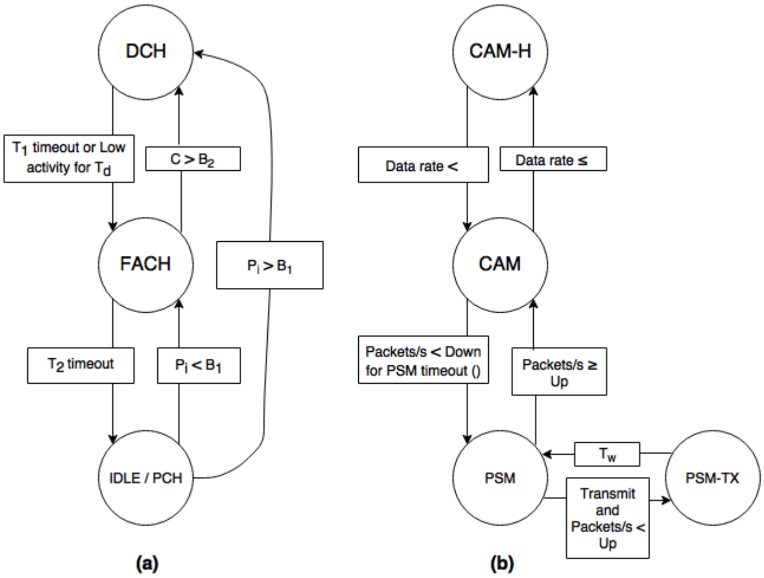


Figure 2.5: EnergyBox State Machine for 3G (a) and WiFi (b)

WiFi model

The WiFi mode in EnergyBox captures and measures the adaptive Power Save Mode (PSM) mechanism based on the amount of packets per second and on the inactivity timer [17]. The state machine shown in Figure 2.5(b) is used to model the data rate behaviour of a station, where the stations switch between two states, PSM and Constant Awake Mode (CAM), using adaptive PSM. No data transmission is performed in the PSM state and the station only wakes up for beacons in this state. This state makes it possible for the station to switch to a low power mode when there is no data transferred. During the transmission interval there are some station models that switch to a high power, and it is in these models where the PSM-TX state represents the transmission of packets in the PSM mode. While the power saving features are disabled the station is awake in CAM.

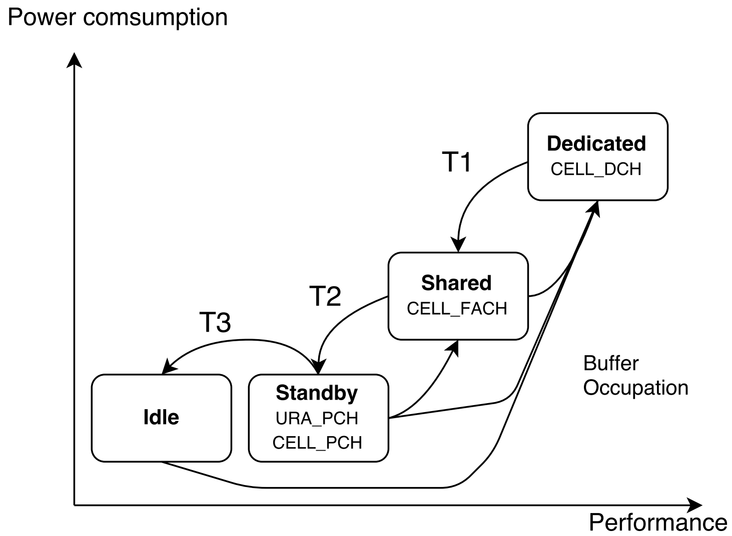


Figure 2.6: RRC State machine

Radio Resource Control (RRC) States

The RRC state machine, shown in Figure 2.6, defines certain specific states that the User Equipment (UE) might be present in [17]. All states have different amount of resources connected with them and these are the resources that a UE uses while being present in the different states. The energy consumption of the UE is determined by this state machine.

In the RRC state there are two basic operational modes: Idle and Connected. The energy consumption is at it lowest in the Idle state where there is no data transmission between the UE and a server or such, put the UE can still check for any available downlink packets. In the Connected state the UE establishes a RRC connection, where the UE connects via the radio physical channel.

In the dedicated state (CELL_DCH) a dedicated physical channel is assigned for the terminal in both uplink and downlink which provides higher data rates in the dedicated state. This terminal has access to dedicated uplink or downlink transport channels, shared transport channels and a combination of them. In the next Shared state, i.e., Forward Access Channel (CELL_FACH), the terminal uses a transport channel in the uplink and listens to the downlink repeatedly. In this state the UE has a lower energy consumption, approximately 50 percent of that in the CELL_DCH, and transmits small data packets at a lower data rate. The last state, the standby state, i.e., Cell Paging Channel (CELL_PCH), the energy consumption is almost as low as in the Idle state but the difference between the Idle and the Standby state is that the UE is allowed to faster switch to higher states. The power consumption in the Standby state is 1-2 percent of the consumption in the Dedicated state.

3 Method

In order to measure how different protocols affects energy consumption in mobile devices, we have constructed a simulator for emulating packet transmissions during a multicast session, while using one of our chosen broadcasting protocols.

Instead of analyzing real video transmission, our simulator broadcasts dummy User Datagram Protocol (UDP) packets, imitating a real UDP carrying video transmission. These transmissions are then recorded with Wireshark. The recorded data is cleaned up so it just contains packets related to the specific transmissions generated by the simulator. These are then analyzed using EnergyBox, to make an estimation of how long the mobile device transmitter has to be in the different transmissions states. This will result in the overall power consumed by the mobile units network components. Finally, the data from each protocol is compared and we analyze how each protocol behave when we change the values of our inputs. These inputs include, the number of channels that the data are broadcasted through, the amount of bandwidth/bit rate on each channel, and the length of the video, where video length and bit rate are kept the same for each measurement. This is done in order to evaluate the behaviours of the different protocols and if there is any significant difference in consumption between them.

3.1 EnergyBox

EnergyBox is a tool developed at Linköpings University in order to measure power drainage in mobile devices when transmitting and retrieving data. The program takes a stored network session in the form of a pcap file and several different configuration files which it combines in order to generate a estimation of the energy consumption that occurred when transmitting the data found in the pcap file. EnergyBox takes real data traces creating realistic and reliable energy estimations.

Given a packet trace and some configuration parameters for EnergyBox, the program outputs the devices states over time. These states are then associated with state-based power levels for a given device and later integrated over time, and this is how the total energy consumption is calculated.

EnergyBox will be used to measure the energy consumption of our simulated protocols, creating graphs of the power usage and to see how much time is spent in each state, discussed earlier in Section 2.8. Figure 3.1 Illustrates generalization process of power usage profiles, and Appendix A presents example profiles from our examples.

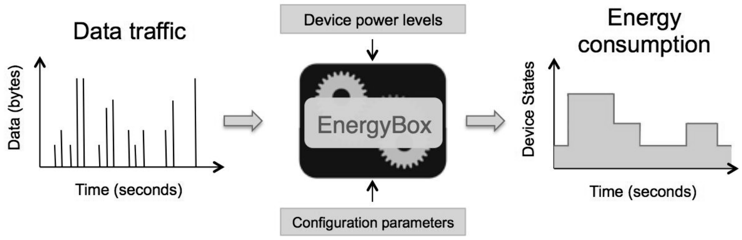


Figure 3.1: EnergyBox [17]

3.2 Simulator

We have built a simulator to emulate the behaviours of the different protocols chosen for this thesis. This software allows us to make customised schedules for the purpose of testing different scenarios, on how fragmentation and channel usage behaves in different scenarios. This while also being able to generate schemes to emulate already established protocols such as Skyscraper, Pyramid, Optimised Broadcasting Protocol and Fast Broadcasting Protocol. The simulator is depending on inputs chosen by the user of the simulator.

- Protocols: SB, PB, OBP, FBP are supported
- Channels: How many channels the simulator will use to transmit the data for the decided protocol.
- Bandwidth: This is converted to a bit rate and used fully.
- Video stream length: The length of the video stream in minutes.
- Playback Rate: How much faster the download rate is compared to playback rate.
- Sleep time: Time the simulator sleep between packet transmissions. If the sleep time is set to 0 then the simulator pre-calculates a δ time to make every packet evenly distributed over the transmission window. Otherwise, δ is as the delay between package transmissions.
- Packet size: Size of each UDP packet in bytes that is to be transmitted.

SB can generate channel requirements depending on assigned bit rate and video length, we have chosen to implement this support into the simulator as-well. When having a predefined amount of channels the simulator schedules fragments to the right channel and in the right order as decided by the chosen protocol. The output is when the channel is receiving data, when it is asleep, how many times the receiver was turned on/off and over how many session totally the channel was receiving data as-well as sleeping. In the final step the simulator will emulate the traffic by sending shaped UDP packets on an available wireless output device, simulating the transmission of our packet model. These transmissions are generated using a multi-threaded solution where each thread represents a channel that during a pre-calculated data timeframe transmits packages and during pre-calculated sleeping zones stops these packet transmissions. In the transmission phase the amount of packets to transmit are calculated by using the values of how many channels that are to be used, the stream length and the size of each packet. These total amount of packages will then be evenly distributed between the transmission phases and transmitted when the simulator enters a pre-calculated data session.

When transferring data there are two modes that can be used. The first is if sleep time is set to zero, then the transmitter will pre-calculate the δ time between each sent package in order to fill the entire data session as seen in Figure 3.2, the other way is to set a predefined δ time to 1 - 2 ms. This will make the transmitter transmit data as fast as possible as seen in Figure 3.3 this effectively shortens transmit time making the resting time frame longer, thus should result in some energy savings.

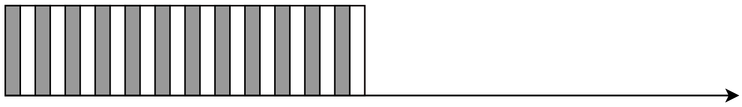


Figure 3.2: Even mode

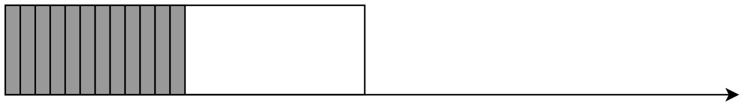


Figure 3.3: Burst mode

3.3 Wireshark

Wireshark is a packet analyzer used for network troubleshooting, network analysis and communications protocol development. It have been used to record data generated by the simulator, as well as a tool for cleaning up the data from other types of packets that might have arrived during the simulated transmission, such as ICMP, ARP packets. This in order to make a trace clean from all other unrelated network traffic, when we make the energy consumption calculation in Energybox.

3.4 Experiment and Design

The experiments are initiated by simulating the different protocols at two pre-determined bit rates, in order to evaluate how much impact a higher bitrate had on the energy consumption. The bit rate also had to be limited as EnergyBox had a hard time handling packet logs surpassing 30Mb. These restrictions lead to the choice of 500 kbit/s and 1000 kbit/s even if both of them are considered to low for a decent video-stream. EnergyBox is fed with configuration files made to emulate a mobile device "Nexus one" and a cellular 3G network "Telia" and the configuration for emulating an "Samsung G. S2" as receiver for our WiFi transmissions. We will use two computers for generating the traffic, one that will be acting as a transmitter and one that will act as an receiver. The receiving end will be using Wireshark and record all transmitted data. While the other is running our simulation software doing data transmissions. The test is replicated at each channel setup. The following raw data is generated.

- Data transmission are done simulating the following protocols PB, SB, FB, OPB using our simulation software.
- Transmission uses two bit rates 500kbit/s and 1000kbit/s using even mode.
- Transmission for each protocol and bit rate setup are transmitted at 2, 3, 4, 6 and 8 channels.
- Startup delay for each channel configuration is measured.
- Transmission for all protocols at 1000 kbit/s bursty mode.

The raw data generation will be followed by simulations that will be executed in EnergyBox, for each of the raw data files generated in the previous phase. These simulations will give insight into how the protocols energy consumption is influenced by different channel configurations as well as how the different protocols distributes transmission sessions.

- Energy consumption is measured at both 3G and WiFi.
- Energy consumption is measured with all above mentioned channel configuration variations.
- Energy consumption at each startup delay is measured.
- Energy consumption differences between even and burst mode is measured.

These simulations outputs files at similar size depending on the bit rate. Recordings done on the 500 kbit/s simulation will result in packet capture of around 15Mb and a 1000 kbit/s transmission will result in a file at around 30Mb, this for a two minute stream simulation. The following settings have been used under these tests. packet size has been set to 1.5Kb, transmit delay has been set to zero for even test and 1ms for all burst mode tests except for FB where huge packet losses occurred when transferring packets more aggressively than what the network adapter could handle, so the parameter had to be set to 2ms. Download rate was set to two times the playback rate. This means that each transfer block represents twice the time during playback.

4 Results

In this section we will take a look at how different protocols behave compared to each other from a set of different variables. Such as energy consumption when using 3G or WiFi, how is evenly distributed packages affecting energy consumption when compared to when receiving them packed together in burst mode. Also how are more channels beneficial for both energy consumption and startup delay and how much energy is required before playback is achieved.

4.1 Pyramid Broadcasting

Energy consumption over channels

Tables 4.1 and 4.2 show the energy consumption for 3G and WiFi experiments, respectively. The results include scenarios when the data rates are 500kbit/s and 1000kbit/s, and cases with both for evenly distributed packages and when packages are sent in burst mode. Results are also summarized in Figures 4.1(a), 4.2(b) and 4.3. We note that the energy consumption at the client is much lower when using WiFi than when using 3G. Both 3G plots show that the energy usage is steadily increased when we increased the amount of channels used, when transmitting the data. While the energy usage is consistent when distributing the video using WiFi. E.g., there is only a difference of approximately 2 *Joule* between the 2 channel simulation and the 8 channel simulation. We can see that when broadcasting over 8 channels the amount of energy consumed is almost doubled when using 3G compared to using WiFi. When comparing consumption using evenly distributed packages and the ones received in burst mode, the the power consumption is much reduced when using bursty transmission compared to evenly distributed ones.

Startup delay over channels

Table 4.3 and Figures 4.1(a) and 4.2(a) show the startup delay for the transmissions, meaning how much time it takes for the client before it is able to start watching the video. There is a major decrease in startup delay when increasing the channels used from 2 to 4, and afterwards it decreases in an even rate. The startup delay is very similar for both 500 kbit/s and 1000 kbit/s, where the largest divergence is at the 2 channel distribution.

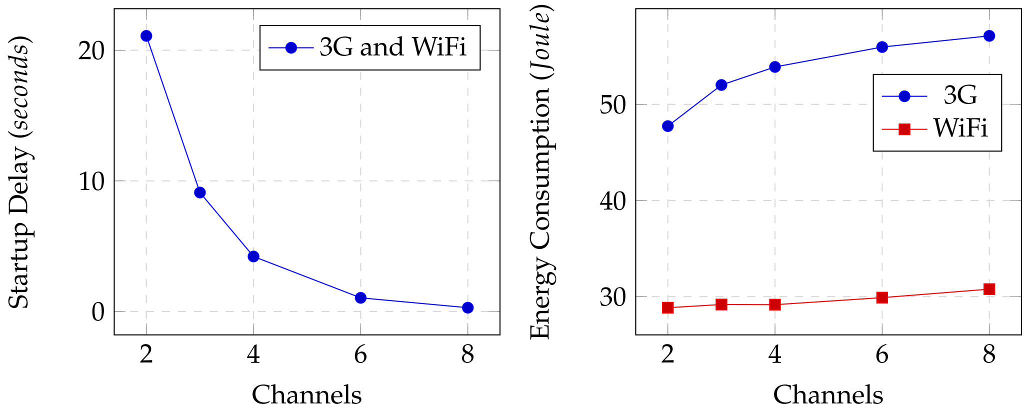


Figure 4.1: Pyramid broadcasting 500 kbit/s, (a) Startup Delay, (b) Energy Consumption.

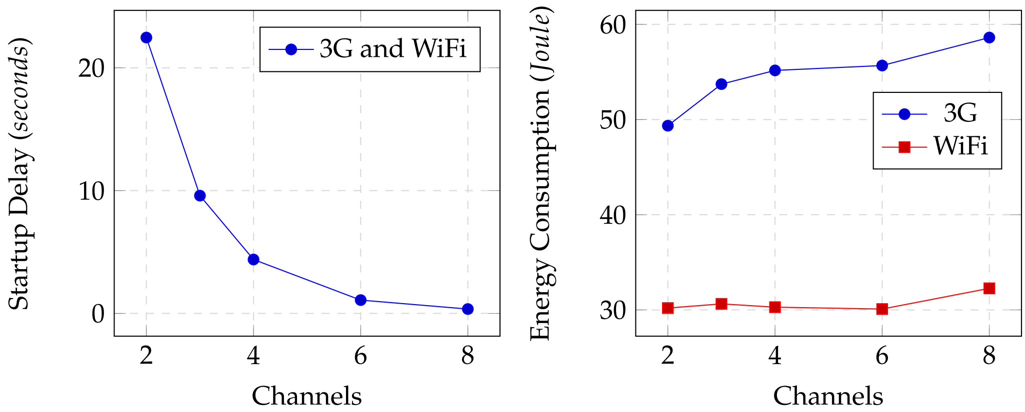


Figure 4.2: Pyramid broadcasting 1000 kbit/s, (a) Startup Delay, (b) Energy Consumption.

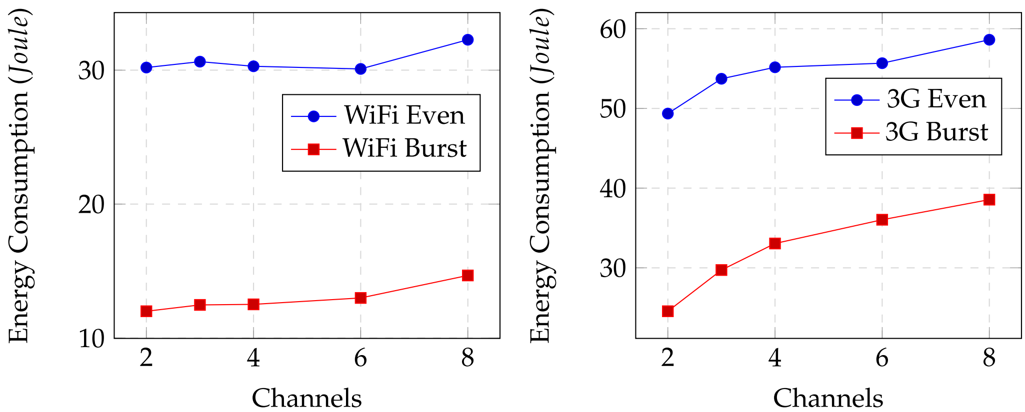
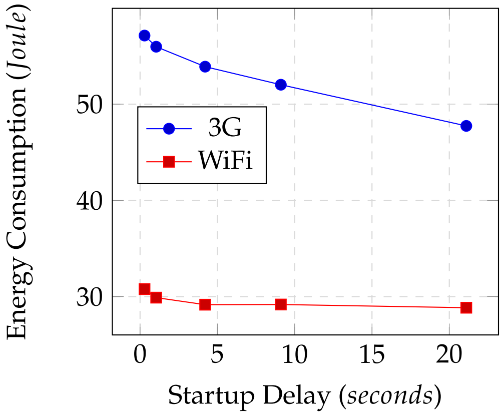


Figure 4.3: Pyramid broadcasting 1000 kbit/s. Even compared to Burst mode, (a) Energy Consumption WiFi, (b) Energy Consumption 3G.

Energy consumption over startup delay

Figure 4.4 shows how much energy it takes to gain the startup delay that each transmission has over 2, 3, 4, 6 and 8 channels. As seen in the graphs it takes about 60 *Joule* to get a startup delay of roughly 0.4 when transmitting over 8 channels. When decreasing the amount of channels the energy consumed decreases as well, though the startup delay increases greatly.



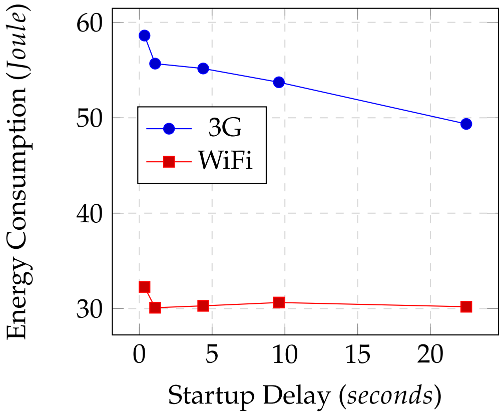


Figure 4.4: Pyramid broadcasting, (a) 500 kbit/s, (b) 1000 kbit/s.

Distribution	Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
Even	500 kbit/s	47.74	52.02	53.90	55.97	57.13
Even	1000 kbit/s	49.35	53.72	55.16	55.68	58.62
Burst	1000 kbit/s	24.55	29.72	33.05	36.03	38.56

Table 4.1: Energy Consumption Pyramid broadcasting 3G

Distribution	Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
Even	500 kbit/s	28.85	29.18	29.16	29.90	30.78
Even	1000 kbit/s	30.19	30.63	30.29	30.09	32.27
Burst	1000 kbit/s	12.00	12.47	12.52	12.99	14.67

Table 4.2: Energy Consumption Pyramid broadcasting WiFi

Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
500 kbit/s	21.101	9.106	4.210	1.043	0.288
1000 kbit/s	22.463	9.589	4.389	1.085	0.356

Table 4.3: Startup Delay Pyramid broadcasting

4.2 Optimized Periodic Broadcasting

Energy consumption over channels

In Tables 4.4 and 4.5 it is shown the energy consumption for 500kbit/s and 1000kbit/s for evenly distributed packages and packages sent in burst mode.This data points are corresponds to the data shown in Figures 4.5, 4.6 and 4.7. The energy consumption with OPB differentiates slightly from the other protocols. When broadcasting using 3G the energy usage reduces when increasing the amount of channels used, for both 500 kbit/s and 1000 kbit/s. Furthermore, the energy consumed when using 8 channels is lower than when broadcasting over 2 channels. The result from when broadcasting using WiFi is pretty similar in both cases and the energy consumption is pretty consistent and close to 30 *Joule* when transmitting on a bit rate of 500 and 1000 kbit/s.

Startup delay over channels

Table 4.6 corresponds to the data found in Figures 4.5(a) and 4.6(a), as they show the startup delay for the transmissions, meaning how much time it takes for the client before it is able to start watching the video. OPB has a startup delay identical to the one of PB, where it peaks at 2 channels and decreases heavily down to around 5 seconds when using 4 channels. It later levels out and the startup delay when broadcasting using 8 channels resulting in a less than half a seconds startup delay.

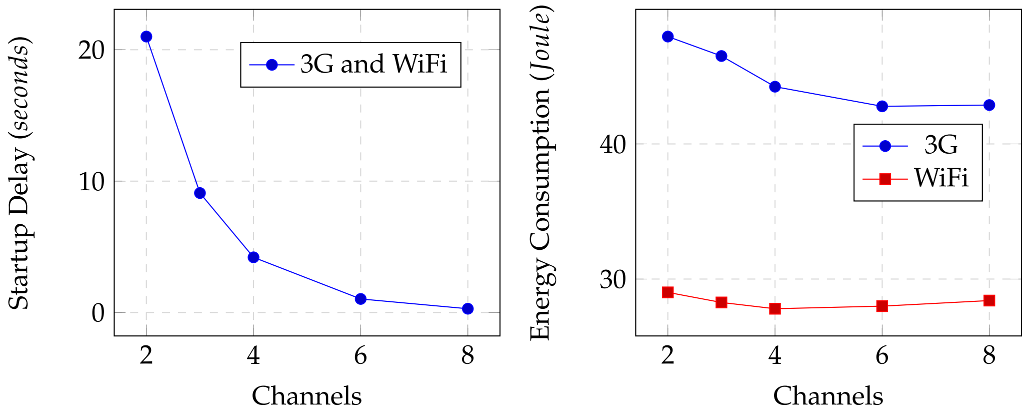


Figure 4.5: Optimized Periodic Broadcasting 500 kbit/s, (a) Startup Delay, (b) Energy Consumption.

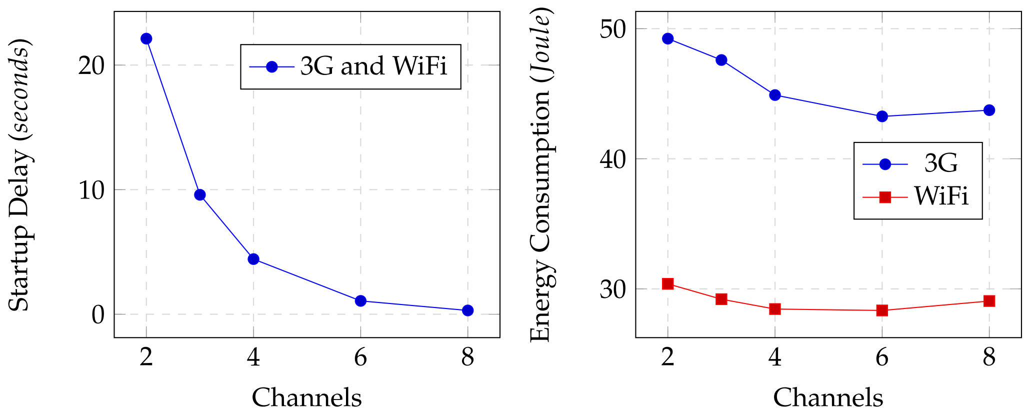


Figure 4.6: OPB broadcasting 1000 kbit/s, (a) Startup Delay, (b) Energy Consumption.

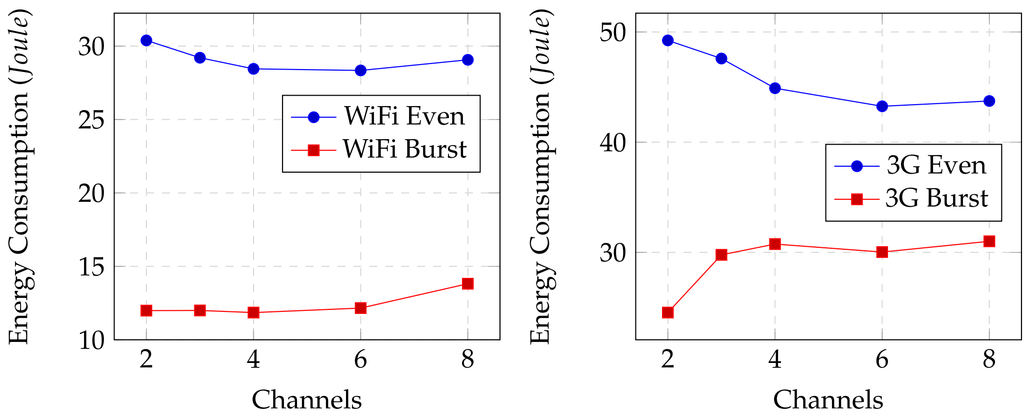


Figure 4.7: Optimized Periodic Broadcasting 1000 kbit/s. Even compared to Burst mode, (a) Energy Consumption WiFi, (b) Energy Consumption 3G.

Energy consumption over startup delay

As shown in Figure 4.8(a) the energy consumption increases along with the startup delay. The consumed energy is at its lowest while transmitting the video using 8 channels, whereas the startup delay is slightly over zero seconds. The energy consumed reaches its peak at 2 channels where the startup delay is around 25 *seconds*.

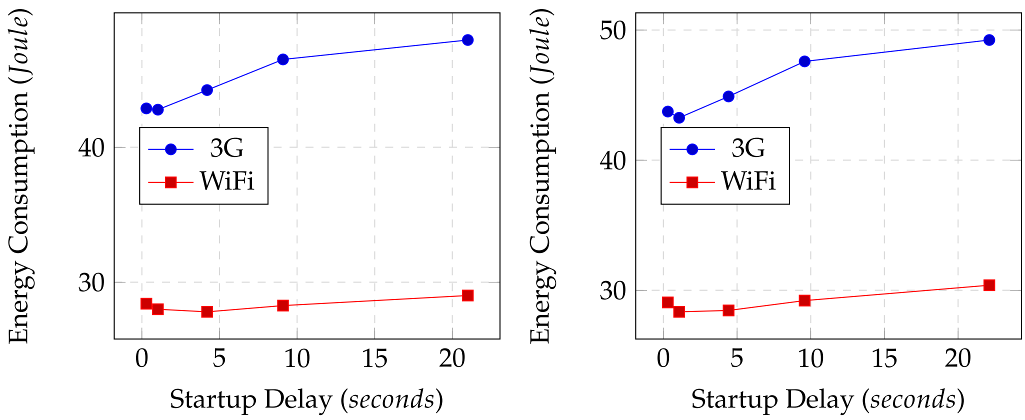


Figure 4.8: Optimized periodic broadcasting, (a) 500 kbit/s, (b) 1000 kbit/s.

Distribution	Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
Even	500 kbit/s	47.95	46.51	44.24	42.79	42.88
Even	1000 kbit/s	49.23	47.59	44.90	43.26	43.74
Burst	1000 kbit/s	24.54	29.78	30.75	30.03	31.01

Table 4.4: Energy Consumption Optimized Periodic Broadcasting 3G

Distribution	Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
Even	500 kbit/s	29.01	28.27	27.80	27.99	28.41
Even	1000 kbit/s	30.39	29.21	28.45	28.34	29.07
Burst	1000 kbit/s	11.99	11.99	11.85	12.16	13.82

Table 4.5: Energy Consumption Optimized Periodic Broadcasting WiFi

Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
500 kbit/s	21.001	9.093	4.201	1.033	0.289
1000 kbit/s	22.121	9.587	4.425	1.070	0.303

Table 4.6: Startup Delay Optimized Periodic Broadcasting

4.3 Fast Broadcasting

Energy consumption over channels

The results presented below are also available in Tables 4.7 and 4.8 it is shown the energy consumption for 500kbit/s and 1000kbit/s for evenly distributed packages and packages sent in burst mode.This data points are corresponds to the data shown in Figures 4.9, 4.10 and 4.11. A front-loading solution such as FB helps with keeping the energy consumption as low as possible. FB divides chunks of the video onto each available channel and tries to download the file as fast as possible. This gives an advantage in power consumption as the receiver only has to work for a short amount of time at high effect and then spend the rest of the session in idle. In Figure 4.9 we can observe that the difference in bit rate on the video file has a small impact on the overall energy consumption. The big difference is when switching interface from 3G to WiFi which reduces energy consumption greatly. The amount of channels used to transfer data has a great affect on energy consumption as the more channels used, the more sections are transferred in parallel making the transfer time shorter. Thus, making it possible for the receiver to go into idle faster. When testing FB in burst mode there were complications, when running the simulation at higher channel counts combined with a 1ms delay between packet transmission. This resulted in a huge drop in successfully retrieved packets as the network adapter could not handle the load and decided to drop packets instead. This was solved by increasing the δ time between packets transmissions to 2ms.

Startup delay over channels

Startup delay is also greatly reduced as can be noticed in, when the amount of channels are increased. This results in the sections becoming smaller and thus contains fewer UDP packets. This greatly reduces the amount of time it takes to receive the first section necessarily to start playing the transferred video. In FB the first section of each channel arrive at about the same time as the first package on all other available channels. It will therefore be slower to start than other protocols as bandwidth is shared between channels. In our case the video is just 2 minutes but the best startup time at 8 channels are still about one second as seen in Table.

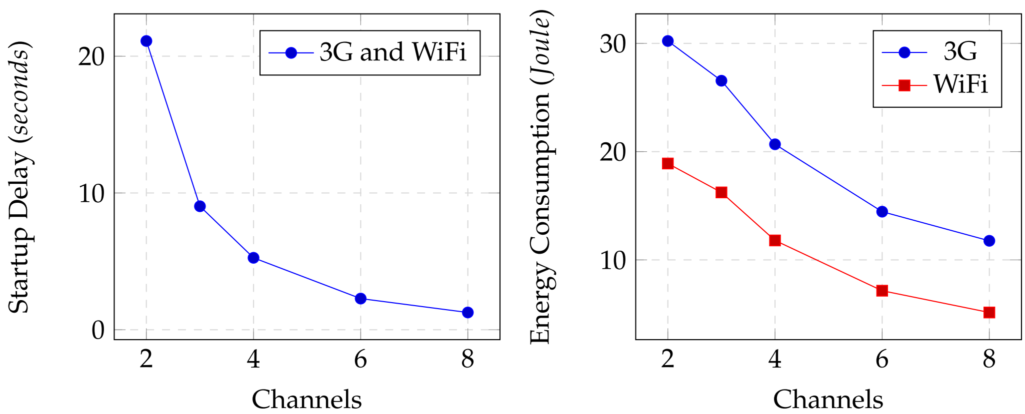


Figure 4.9: Fast broadcasting 500 kbit/s, (a) Startup Delay, (b) Energy Consumption.

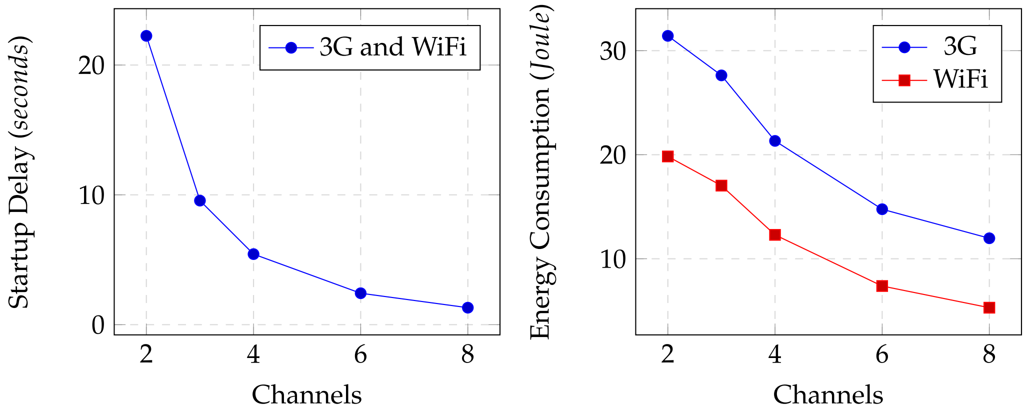


Figure 4.10: Fast broadcasting 1000 kbit/s, (a) Startup Delay, (b) Energy Consumption.

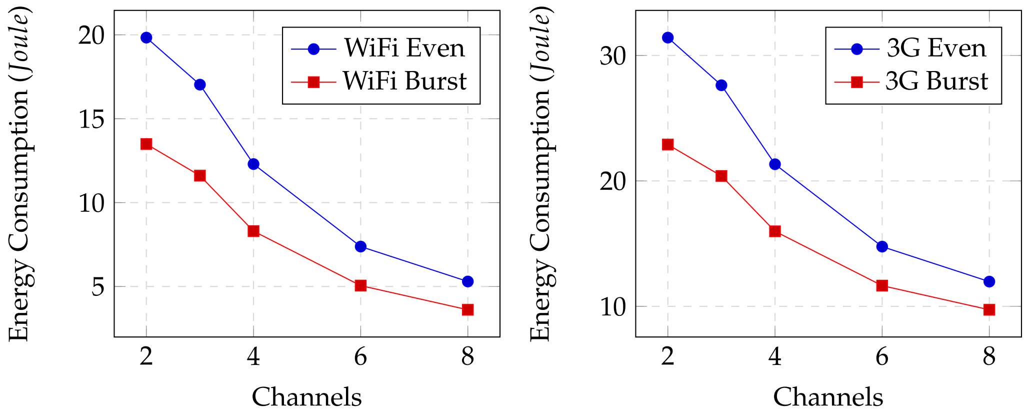


Figure 4.11: Fast broadcasting 1000 kbit/s Even compared to Burst mode, (a) Energy Consumption WiFi, (b) Energy Consumption 3G.

Energy consumption over startup delay

Energy consumption before having the first section completly downloaded are greatly reduced as more channels are used. It can bee seen in Figure 4.12 that the longer the user have to wait in order to get the first section, the higher the power consumption for getting the first section will be.

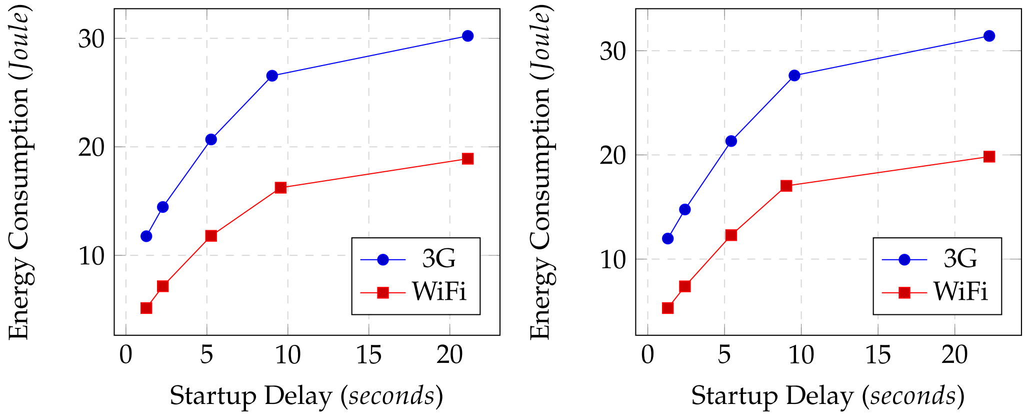


Figure 4.12: Fast broadcasting, (a) 500 kbit/s, (b) 1000 kbit/s.

Distribution	Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
Even	500 kbit/s	30.22	26.55	20.68	14.45	11.76
Even	1000 kbit/s	31.42	27.63	21.33	14.76	11.97
Burst	1000 kbit/s	22.90	20.39	15.98	11.65	9.73

Table 4.7: Energy Consumption Fast broadcasting 3G

Distribution	Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
Even	500 kbit/s	18.90	16.23	11.80	7.15	5.14
Even	1000 kbit/s	19.83	17.03	12.30	7.38	5.30
Burst	1000 kbit/s	13.49	11.61	8.30	5.05	3.61

Table 4.8: Energy Consumption Fast broadcasting WiFi

Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
500 kbit/s	21.112	9.028	5.260	2.279	1.262
1000 kbit/s	22.246	9.555	5.435	2.422	1.308

Table 4.9: Startup Delay Fast broadcasting

4.4 Skyscraper Broadcasting

Energy consumption over channels

As seen in Tables 4.10 and 4.11 the energy consumption is shown for transmission rates of 500kbit/s and 1000kbit/s, and for both evenly distributed packages and packages sent in burst mode.This data points are corresponds to the data shown in Figures 4.13(b), 4.14(b) and 4.15(b). The energy consumption when switching from 500 kbit/s to 1000 kbit/s increases faster as channels are added, but the growth evens out as it reaches the peek of 8 channels. At two channels the consumption is lower but not capable of competing with most other protocols. SB increases segment length as more channels are added and this forces the receiver to stay on for a longer time with each added channel, which can be shown in the Table mentioned above. When measuring the consumption on WiFi we can see in energy figures that the energy consumption starts at a much lower level and keeps growing much slower than the 3G simulation.

Startup delay over channels

Table 4.12 corresponds to the data found in Figures 4.13 and 4.14, the start-up delay is greatly reduced as more channels are introduced. This is to be expected as the segments in the beginning of the file is greatly reduced in size. Skyscraper grows the segments on the later channels fast but as an result the early channels get smaller segments which are faster transferred, which leads to a decreased startup time. There is some difference between 500 kbit/s and 1000 kbit/s as the later is somewhat slower to start, but the difference decreases as more channels are added, as seen in the table mentioned above.

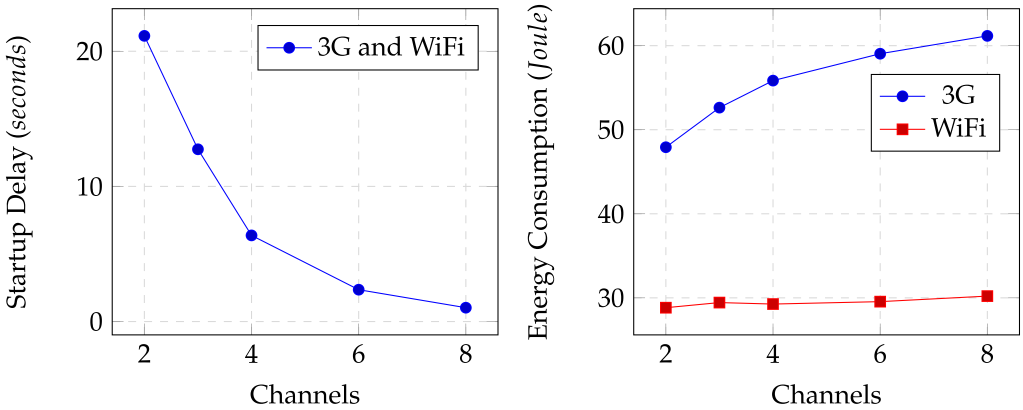


Figure 4.13: Skyscraper broadcasting 500 kbit/s, (a) Startup Delay, (b) Energy Consumption.

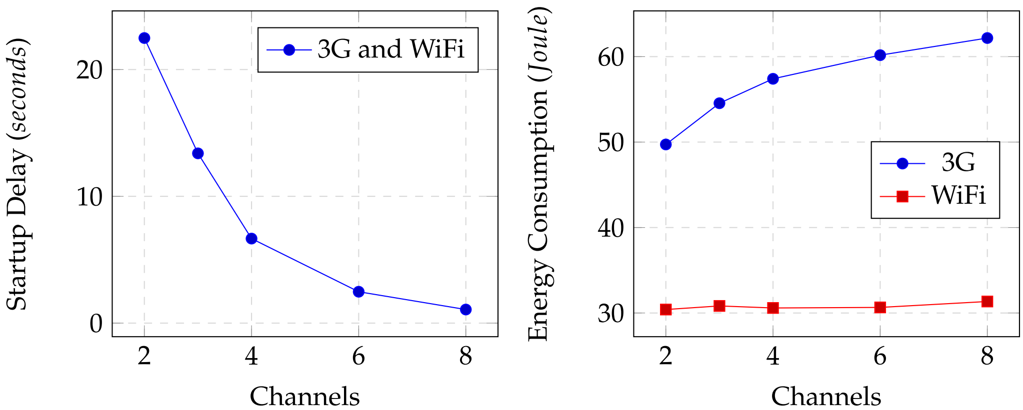


Figure 4.14: Skyscraper broadcasting 1000 kbit/s, (a) Startup Delay, (b) Energy Consumption.

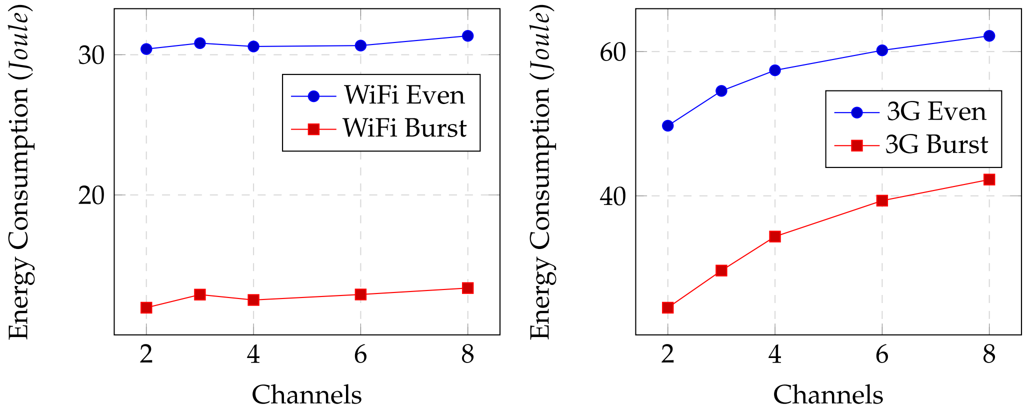


Figure 4.15: SkyScraper broadcasting 1000 kbit/s Even compared to Burst mode, (a) Energy Consumption WiFi, (b) Energy Consumption 3G.

Energy consumption over startup delay

In Figure 4.16 it is possible to distinguish that the longer startup delay results in much less consumed energy. This as the energy consumption grows greatly with each newly added channel when using 3G. This is not true when observing the behaviour of WiFi which keeps the energy consumption at a relative even level when more channels are added.

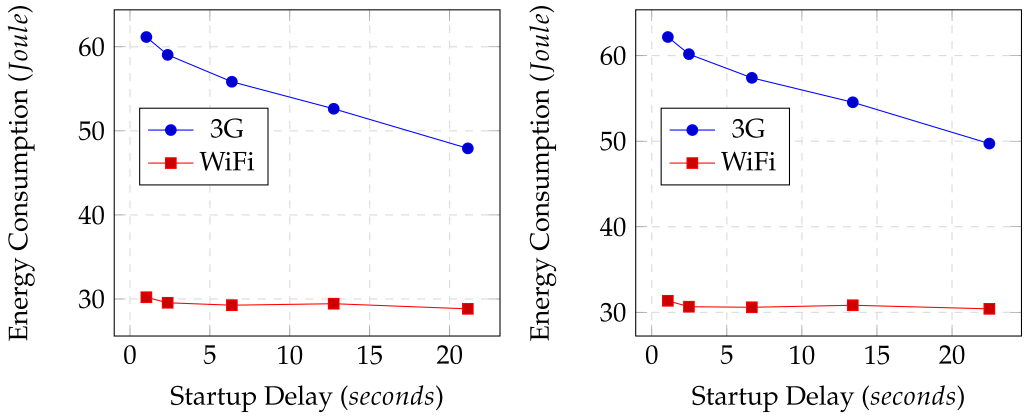


Figure 4.16: Skyscraper broadcasting, (a) 500 kbits, (b) 1000 kbit/s.

Distribution	Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
Even	500 kbit/s	47.92	52.63	55.84	59.04	61.16
Even	1000 kbit/s	49.72	54.55	57.40	60.17	62.17
Burst	1000 kbit/s	24.50	29.64	34.36	39.34	42.26

Table 4.10: Energy Consumption Skyscraper broadcasting 3G

Distribution	Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
Even	500 kbit/s	28.83	29.44	29.26	29.55	30.21
Even	1000 kbit/s	30.41	30.83	30.59	30.65	31.35
Burst	1000 kbit/s	11.96	12.89	12.51	12.90	13.36

Table 4.11: Energy Consumption Skyscraper broadcasting WiFi

Bit rate	2 channels	3 channels	4 channels	6 channels	8 channels
500 kbit/s	21.139	12.746	6.374	2.357	1.025
1000 kbit/s	22.485	13.389	6.667	2.475	1.071

Table 4.12: Startup Delay Skyscraper broadcasting

4.5 Protocol Comparison

Figures 4.17 and 4.18 compare the energy-delay tradeoff for each protocol with each other when the transmission of packets are evenly distributed. As seen in the graphs the energy-delay tradeoff for FB is a lot better, for both 3G and WiFi, than for the other protocols. Furthermore, the results for PB, OPB and SB are pretty even, where OPB diverge a little from PB and FB in the 3G tests. Presented separately in the bar diagrams below are the energy consumption and startup delay for each protocol, both for 3G and WiFi with a bit rate of 500 kbit/s and 1000 kbit/s. The data in the bar diagrams is combined to get the data presented in Figures 4.17 and 4.18.

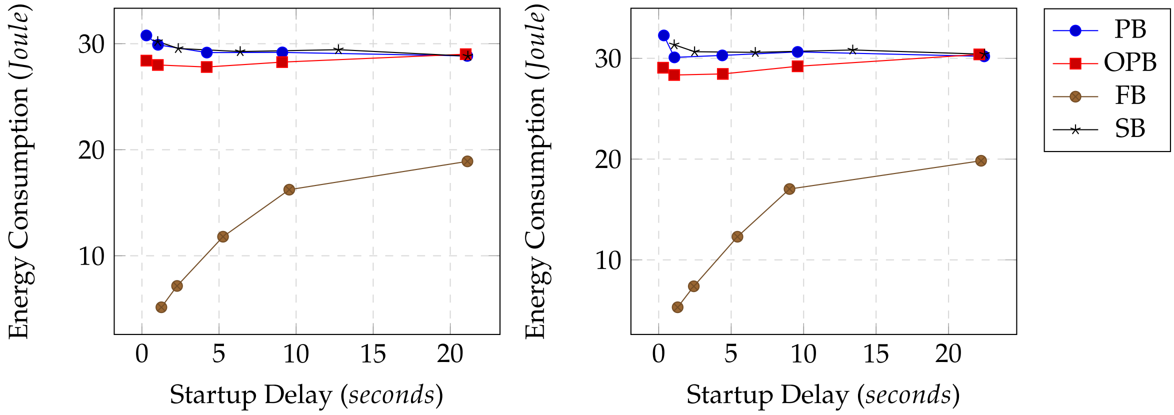


Figure 4.17: Protocols WiFi energy-delay tradeoff, (a) 500 kbit/s, (b) 1000 kbit/s.

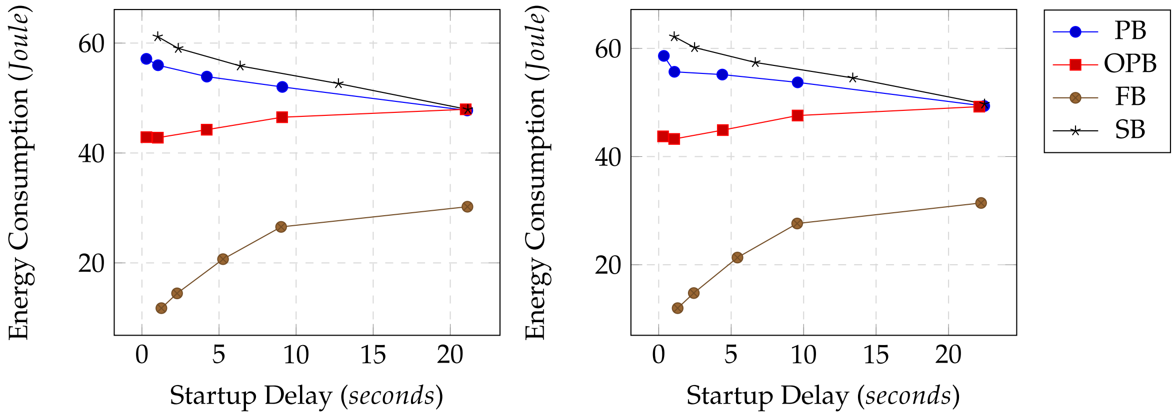


Figure 4.18: Protocols 3G energy-delay tradeoff, (a) 500 kbit/s, (b) 1000 kbit/s.

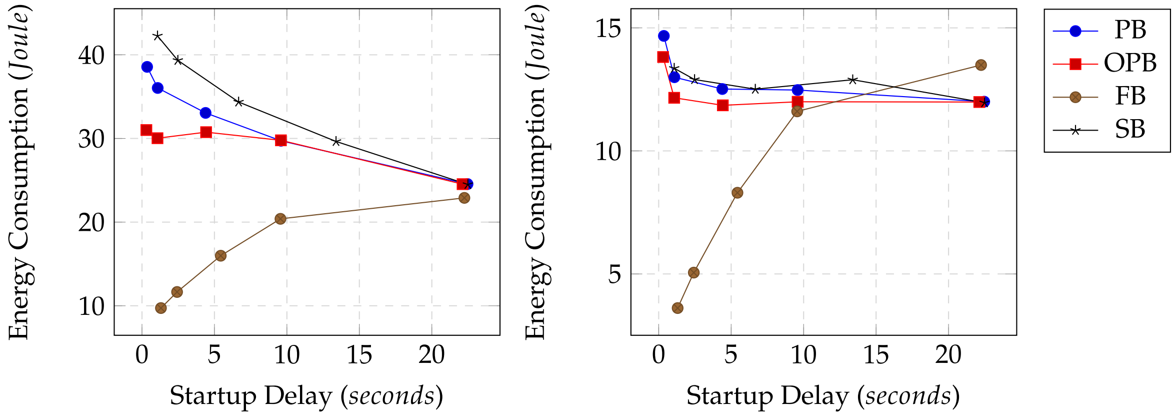


Figure 4.19: Protocols energy-delay tradeoff 1000 kbit/s Burst mode, (a) 3G, (b) WiFi.

Energy consumption over channels

All four Figures 4.20, 4.21, 4.23 and 4.24 represent the energy consumption with the four protocols that we have analyzed and simulated. As can be seen FB has a far lower energy consumption than the rest of the protocols. Unlike SB that almost has a energy consumption six times that of FB when transmitting on 8 channels.

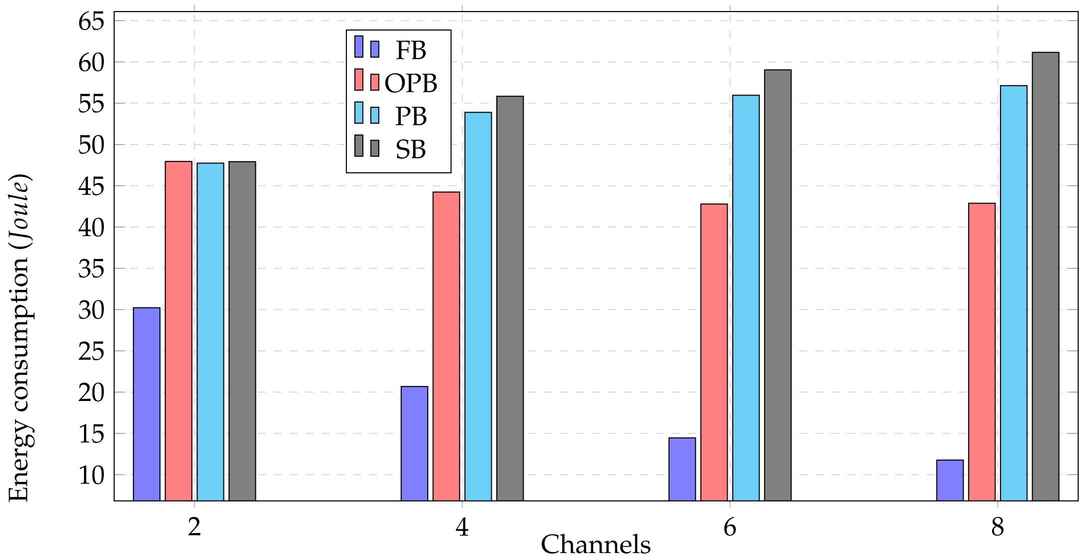


Figure 4.20: Protocols 3G 500 kbit/s.

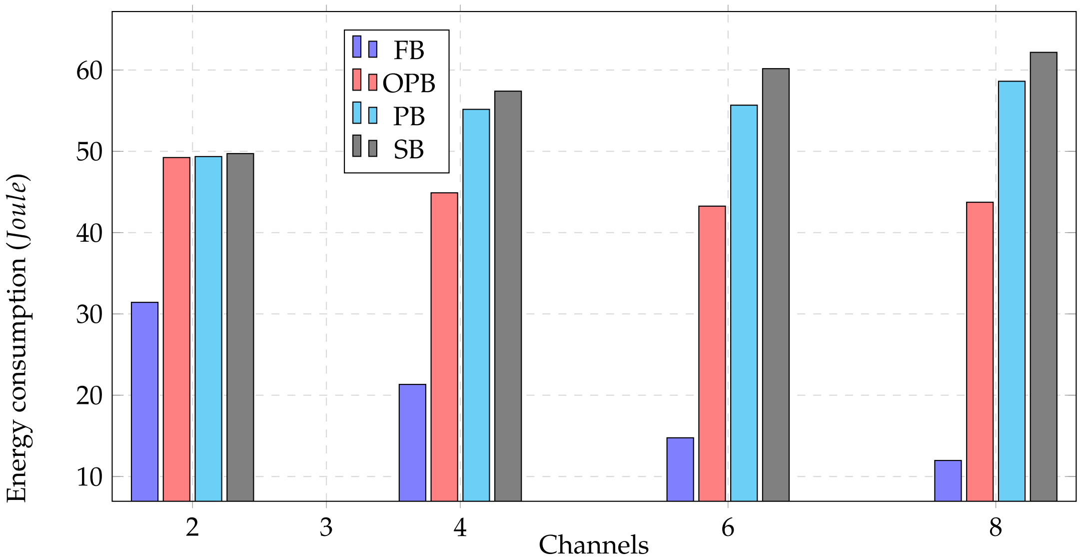


Figure 4.21: Protocols 3G 1000 kbit/s.

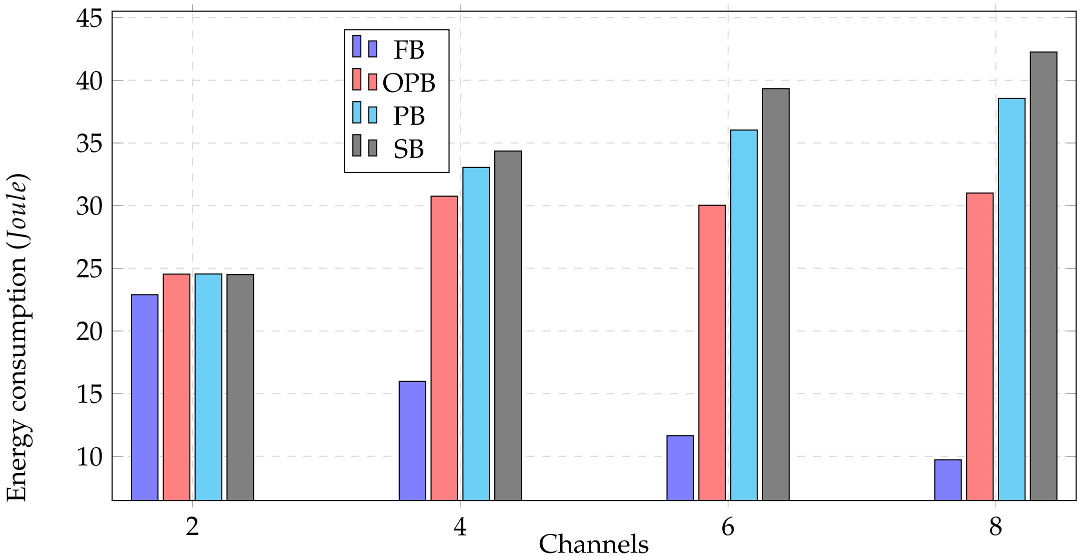


Figure 4.22: Protocols 3G 1000 kbit/s Burst mode.

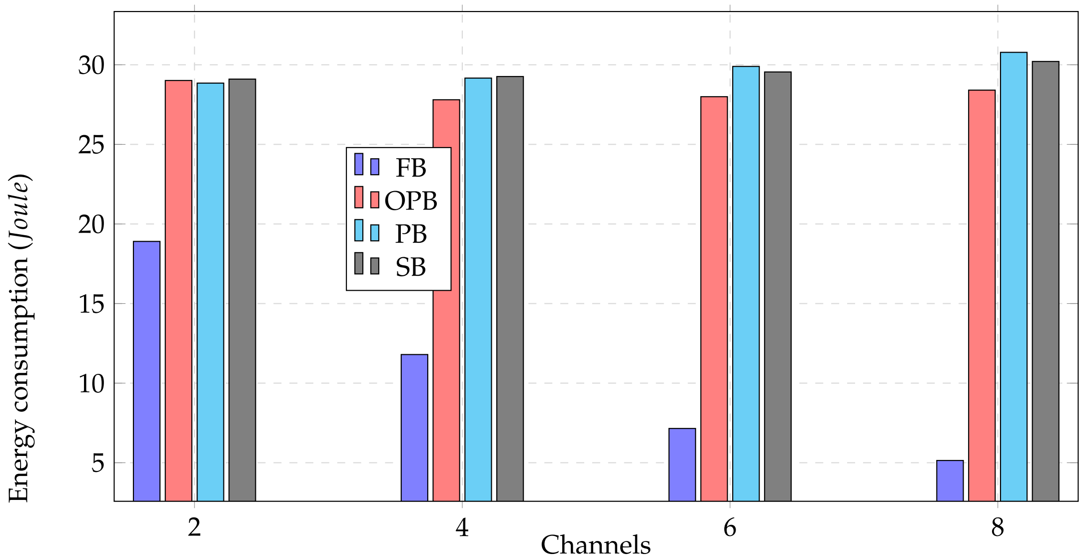


Figure 4.23: Protocols WiFi 500 kbit/s.

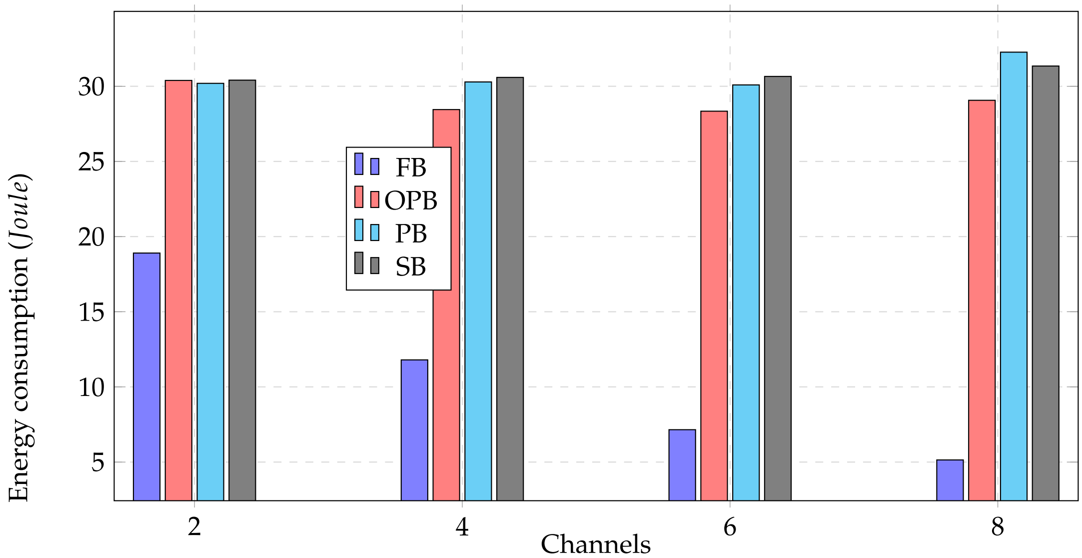


Figure 4.24: Protocols WiFi 1000 kbit/s.

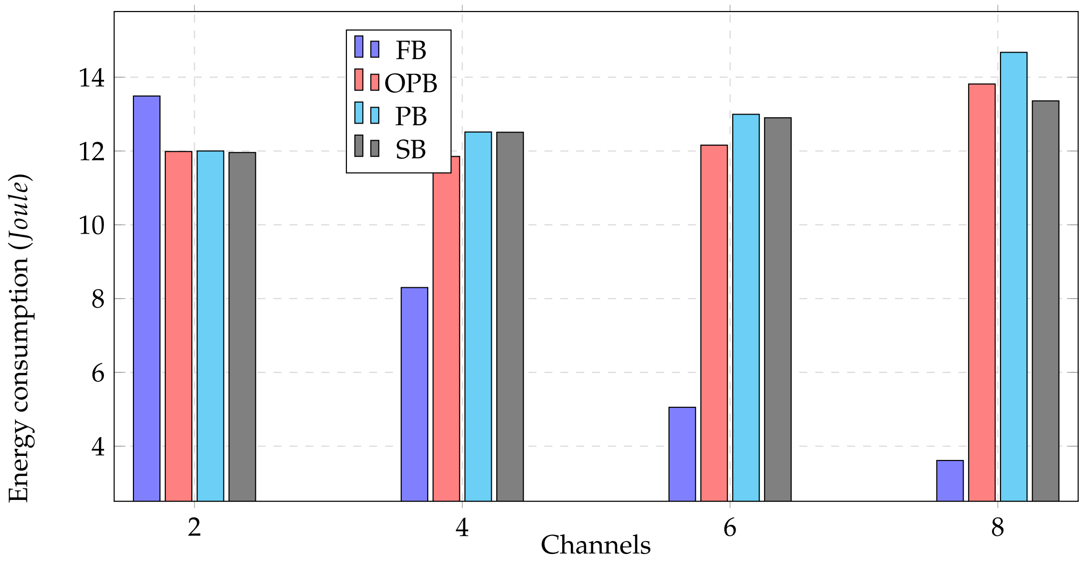


Figure 4.25: Protocols WiFi 1000 kbit/s Burst mode.

Startup delay over channels

As can be seen in Figures 4.26 and 4.27 more channels reduces start up time greatly on all protocols with no exception. This is as expected as early segments gets smaller with more channels. The ones taking the most benefit is OPB and PB as their earlier segments shrink faster than the others. When using just 2 channels the difference between protocols are almost negligible but when we reach 4 channels the gains are huge and when 8 channels are reached some are having startup times within less than one second, which could be considered good since the simulated video is 2 minutes long.

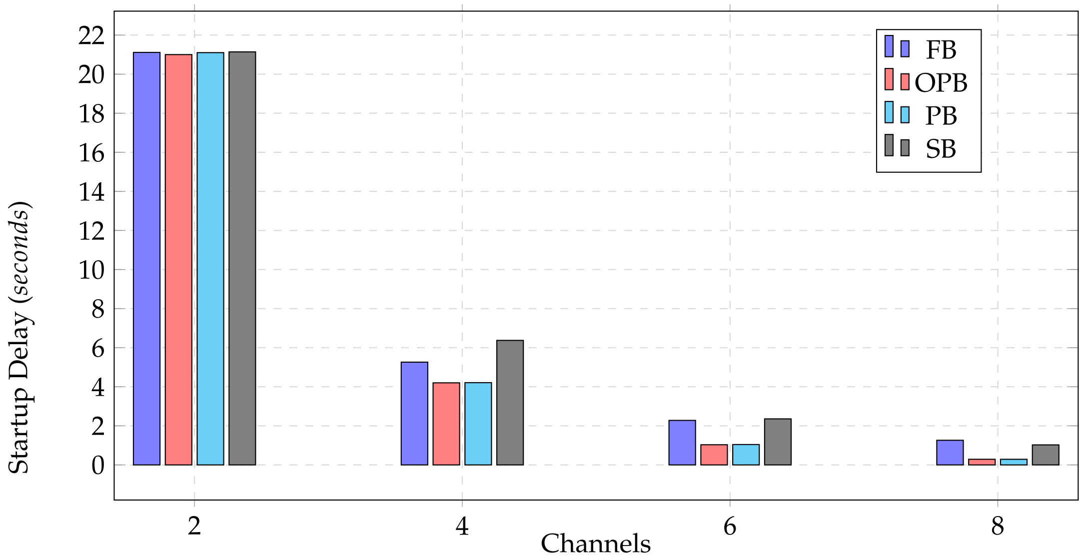


Figure 4.26: Protocols 500 kbit/s startup delay.

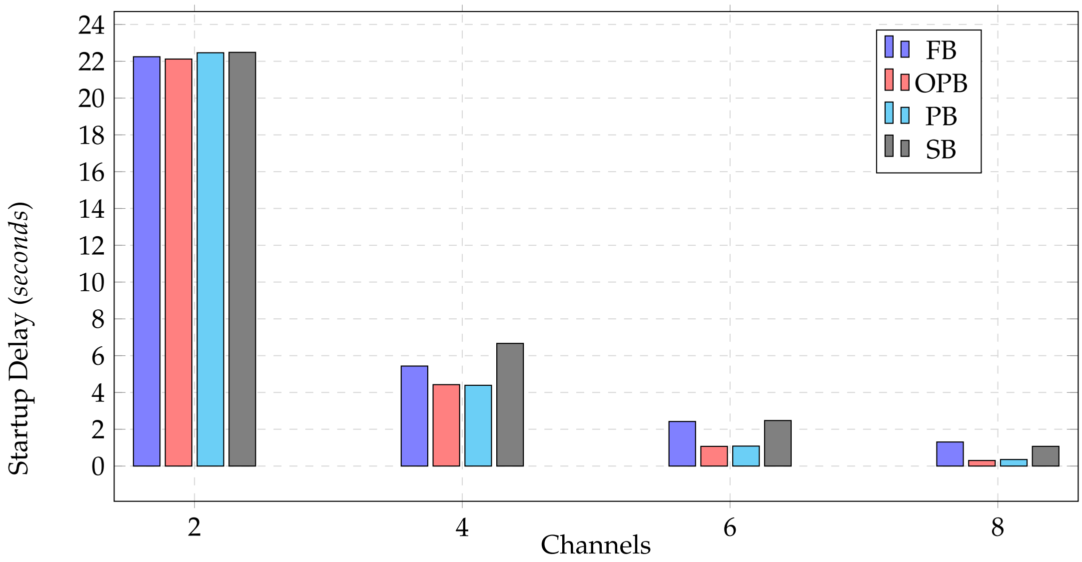


Figure 4.27: Protocols 1000 kbit/s startup delay.

5 Discussion

In this chapter we mainly discuss the two major parts of this thesis, the result and the methodology. However, we also address some of the limitations we have taken into account and how these have influenced our work.
We will also be referring to figures found in the Appendix(numbered as A.X, where X is the figure number within the Appendix).

The first restriction we did was to decide upon how many protocols we would wanted to work with. There was not enough time to evaluate each and every broadcasting protocol, so we decided to evaluate four that we found interesting. Thus, limiting both our study and our results. Even though we got a clear result of which protocol, of the ones we studied, is the best when it comes to saving energy on a receiving device the study could be expanded to evaluating more protocols.
Another delimitation we decided upon before we began this study was to only analyze the receiver and how the data transmission affected the energy consumed with the end client. We decided on this approach because we found it more interesting and more important to have a knowledge about how to optimize a data transmission from the end users point-of-view.

All delimitations were not decided upon starting this thesis. There were also several ones that had to be taken into account as we encountered problems that forced us to change our approach. For example we encountered that EnergyBox had a limit on how large the files were allowed to be for EnergyBox to process them. This had us restricting our video file to two minutes as well as keeping the bit rate bellow 1200 kbit/s.

5.1 Results

Quite a few results turned out as expected. The preliminary belief was that a front loading protocol would get advantages as it is more efficient to transfer data as fast as possible and then turn off the receiver entirely, than to spread it out and keep the receiver working for a longer time. Additional tests where done here to further test this. This is where we simulated to transmit packages in bursts, meaning that we tried to transfer them as fast as possible in all transfer windows in-order to let the receiver start resting even within the transfer window itself. The other testing states were done to evenly distribute the packages, with the theory that if few enough packages were to be received after each other, the system would revert to a lower energy state, as too few packages were transmitted thus not breaking the threshold keeping the receiver in the DCH state.

Fast broadcasting where believed to be very efficient from the start, it were also expected that it would gain greatly from additional channels as it is capable of downloading segments faster, the more channels it get available. In Figure A.6 it is possible to distinguish from the graph that the download rate makes the receiver go into the DCH state and when the requests stops it stays there for a while before going down to FACH. This behaviour can be found in all figures representing Fast broadcastings states. The device will stay in FACH for a predetermined time that is predetermined and therefore will always be the same, the same applies for the higher power level DCH, where it will stay a while after the last request has been received before stepping down. In Fast broadcasting this is used efficiently as most packages can be transmitted during very few transfer windows, this is where the receiver is fully active. It were even further improved upon when allowing it to send packages in bursts. This as multiple channels are transmitting in parallel whilst the receiver keeps active in DCH state with very short delays between the packages. As it transfers everything fast it can then quickly allow the receiver to go down to PCH/Idle again. An increase in channels transmitting parallel greatly reduced start up delay as-well as it is using the receiver more efficiently. But the downsides are that it could never get close to the startup times like some of the other protocols tested in this thesis and as the video we simulated was only 2 minutes long. This could become annoying if the video were to be longer than the one simulated in this test, which is most likely the case for most real life applications.

PB where one of the highest power consumers and it kept growing as more channels where added, especially bad is it when the transmission are done at 3G where the energy consumed raises aggressively, as can be seen at Figure 4.1. This is most likely due to the receiver never being able to go down to PCH/Idle. When the amount of channels are increased so are the spread of the DCH states. PB is spreading its sections very well due to how it is constructed. When one section is done the receiver starts to download the next one on a different channel. As the sections are not ever overlapping the efficient usage of the receiver becomes poor. This were much improved upon when making use of bursts instead, allowing the receiver to step down from DCH to FACH and even PCH. If A.13 and A.15 are compared it can be distinguished that the time the receiver is allowed to stay in PCH/Idle is greatly increased, the same results can be found when analyzing A.14 and A.16 where the transmission at 8 channels had almost no opportunity in wide mode to reach the lower power levels, which were changed when allowing burst mode. The power savings in this case where close to 20 *joule* which is a reduction of 35% which can be considered a quite good improvement. PB benefits greatly from having more channels when observing the start up delay. This as it is growing its sections very fast and thus making the earlier ones very small which is good when you want to get the playback going as fast as possible. Taking a look at Table 4.3 it can be noticed that it is having one of the best start up times of all the protocols this thesis have conducted tests upon and when working over WiFi the energy consumption is acceptable. The question would be on how much more channels would decrease start up delay and how much the power consumption would increase as a result.

SB is also among these high consumers, as the energy consumption grows fast with more added channels especially on 3G. It keeps around PB levels of power consumption when running over WiFi. This could be explained in Figure A.10 where the power states are changing often between DCH, FACH and PCH. The more channels that are added the more time the protocol spends in DCH as can be seen in Figure 4.13 this greatly increases power consumption. At lower channel counts the receiver actually get to rest and powers down to PCH/Idle which is why fewer channels actually are consuming less power. This is greatly improved upon when running in burst mode, allowing everything to get done as soon as possible and the amount of time spent in PCH/Idle is therefore greatly increased as seen in A.9 where it is spread over the transmission window, and A.11 where it is transmitting as fast as possible on 2 channels. When analyzing the 8 channel configuration it can be noticed that in A.10 there is almost no PCH/Idle occurring at all and at most cases the receiver is not allowed to step further down than the FACH stage. This is changed a great deal when looking at the same channel configuration again but instead with using burst mode A.12. As seen in the graph is the receiver able to not only reach FACH but PCH/Idle as well and therefore reducing the consumed power from around 61.2 *joule* to approximately 42.3 *joule* this giving a reduction of around 18 *joule* which is an improvement of about 31% which indicates that sending data as fast as possible is greatly advantageous.

A higher amount of channels helps getting the startup delay down, but it is still the worst competitor in this test. When comparing the startup delay between protocols it shows quite bad results with only FB getting worse results in this test. As SB grows slower than PB as a result of its design. It will therefore require much more channels in order to get the earlier sections small enough to be transferred within a reasonable timeframe.

OPB is giving quite different results when comparing the 500 kbit/s and 1000 bkit/s, see Table 4.5 and 4.6. At 2 channels at 500 kbit/s OPB gets to rest between the DCH states as fewer requests results in it being able to reach a more FACH states. Which should explain why it is consuming less power than the ones using more channels. When using 8 channels we get another dip in power consumption and this could be related to that the receiver gets everything done in one go. As seen in Figure A.2, this is similar to the behaviour of Fast Broadcasting which would explain the reduced power drainage as the transfer time is reduced. It should be noted that that four channels have a disadvantage caused by the built in delay feature in 3G data transmissions. This resulting in the receiver not being able to step down to FACH as the gaps between transmissions are shorter than the timeout for changing energy state. When the receiver enters a resting window, its shorter then the timeout so it stays in DCH for the entire session, even when not sending data as seen in previous figure. This is much improved upon when running OPB, in burst mode instead as seen in Figure A.4 the advantage gets even more obvious when comparing Figure A.1 and A.3 and the advantage in energy consumption between the 8 channel configurations goes from 29.1 *joule* when evenly distributed to 13.8 *joule* when transmitting in bursts. This is a difference of about 15.3 *joule* giving a 53% advantage, which is the biggest gain of all protocols when switching in between evenly distributed and burst distributed transmissions. When observing the startup delay it is obvious that OPB is very good at delivering the first session fast as the startup delay is very low which can be observed in Table 4.9. This gets better as more channels are added resulting in one of the lowest startup delays in this test. When looking at energy consumption compared to startup delay we notice that it comes with a cost as longer startup delay in this case results in lower power consumption.

When comparing the tested protocols with each other we find that from an energy perspective Fast Broadcast is the one with the lowest energy consumption and OPB is the one with the shortest startup delay. This can be seen easily in Figures 4.21 and 4.20, the difference is obvious when analyzing the WiFi graphs as most are keeping themselves around the same energy consumption levels except Fast Broadcasting which gains a lot by the WiFi interface as seen in Figures 4.23 and 4.24. When taking this values and comparing them to the ones generated when receiving data in bursts, the energy consumption shrinks quite distinctively but it is still FB which is the one consuming the least amount of power as seen in Figure 4.22. The advantage has shrunken as both Pyramid and Skyscraper were consuming about 30% more energy when transmission where evenly distributed compared to the ones transmitted in bursts. OPB makes a great gain as it was consuming approximately 50% more when transmission where evenly distributed. When comparing the consumption using WiFi even greater benefits can be found as most protocols are consuming about 50% more when evenly distributed, except for FB as its gains are planning out. We expect that FB is reaching a lower bound on consumption, it might be reduced somewhat more if more channels where to be added but not with as much gains as earlier reductions. When analyzing startup delay it is OPB and PB who are the winners as both FB and SB do not keep up as more channels are added which can be seen in Figures 4.26 and 4.27. In our simulation there were also no gains in startup delay when using burst as the simulator only declared transmission complete when the transmission window where over which in bursts case would occur a while after the actual data transmission were completed.

In Figures 4.17 and 4.18 there is a comparison for how energy consumption related to startup delay behaves for all protocols. As earlier discussed FB got a big advantage over the entire range of channels. This advantage grows fast compared to the other protocols who have a more modest gain. This is of course because of the heavy front loading, where FB gets more done whenever activating the receiver. There are also some gains for OPB as more channels are added with a small peak at 6 channels, this behaviour can be found in the result for both 3G and WiFi. If additional channels where to be added our theory is that energy consumption will continue to rise. In both SB and PB a rise of consumption can be observed as additional channels are added, this is caused by how it is distributing packages over the transmission windows. In Figure 4.19 similar results as above can be observed but as bursty mode gives an advantage in energy consumption related to shorter transmission windows, gives the graphs the same behaviour, but at a lower energy level.

5.2 Method

The method used is measurement and data driven. A simulator have been theorized and coded, and this simulator have been used to simulate large streams of data packets imitating real time video traffic, these data have then been analyzed. To capture the data transmission we used the tool Wireshark presented in Section 3.3 and later analyzed this data using EnergyBox presented in Section 3.1.

Even though we are satisfied with our result and think that the thesis is reliable, a way to get a more reliable result could be to analyze larger data files as the bit rates that was analyzed in this thesis is much lower than what can be anticipated in real-life applications. But this requires using another tool than EnergyBox or in some way rewriting the program making it capable of processing larger files.
Our simulator allows us to do several simulations in a short period of time, which means that we can repeat our experiments a number of times, allowing us to confirm our results. This increases the thesis reliability even more, and if someone were to replicate this study they could expect to get similar results as the ones presented in this thesis.
One could argue that a broader range of protocols would be beneficial, as this study is limited to just four protocols. This is a reasonable thought. However, considering the fact that we created our own data sets it would consume to much time to theorize and write the code in order to create additional protocol simulations.

5.3 The Work in a Wider Context

This thesis presents how four different broadcast/multicast protocols affect the energy consumption and startup delay with the end clients during a data transfer. These results could be used in further work or as mentioned in Section 5.2 could be improved and extended by using larger data files, using a bit rate which are are more common in real-life situations and to study additional protocols. Furthermore, including Long Term Evolution (LTE) in a future study would be interesting since the 4G network is expanding and overtaking 3G as the default one for data transactions. There are some protocols that could benefit from using additional channels in order to get its startup delay down, this includes SB, which grows slowly and would therefore greatly benefit from being able to spread the data over more channels.

There could be several untapped benefits from using multicast techniques, such as eMBMS. One example is when transmitting advertisement. A multicast solution for a shorter transmission could be used, as the server keeps an open broadcast for these shorter video clips. This would be beneficial from a power perspective as the local transmitter is able to shut down these channels when no one is listening to them. This could also be used in order to send specialized information for certain customer groups. For example, in a student district more focused/personalized advertisements could be used. These transmissions would be broadcasted in parallel with multiple other personalized video stream.

6 Related work

Broadcasting protocols are mostly conceptually equal. One example is that almost all protocols are splitting the streamed video file into smaller segments [15]. In spite of this their real world performance vary greatly from each other. Skyscraper is very good at minimizing the buffer footprint on the device receiving the stream but it is also wasting bandwidth according to Carter et al. [15]. This as it has duplicate segments transmitted on different channels at the same time. According to Hu [9] Skyscraper broadcasting is not alone in this endeavor as Pyramid broadcasting and Permutation-based broadcasting also has this problem. This problem is not to critical for the receiver as it can simply ignore the duplicates which is where our focus is. There are other protocols that are downloading big sections of the content at initiation. As a result they might miss to deliver segments in time. This will force the video to pause in order to buffer [9]. There are many books and papers focusing on optimizing the buffers and making variations on protocols in order to keep data stored on the client as low as possible [7, 3, 12]. There are also several papers that looks into which protocols to use and how to make the startup delay as short as possible, this will neither be the focus of this thesis [3]. Carlsson and Eager [1] discusses the research that digital fountains could be a solution to get costs down when distributing video using OPB. Digital fountains is an erasure coding technique, meaning it could reconstruct lost data by using key data blocks it has previously received. This would be considered to be a technique for packet loss recovery. Packet loss and packet recovery is not something that can be considered to be within the scope of this thesis.

The popularity of using mobile devices to stream videos have increased [13]. Making the need to understand how power consumption is affected by different types of data traffic essential, as mobile devices still are very limited by their battery life.
In this report we have studied how the power consumption of several different Multicast/Broadcast protocols perform compared to each other.

7 Conclusion

This thesis presents a comparison between different broadcasting protocols. By observing the results we are capable of drawing several conclusions. Primarily we can see that Fast broadcasting has the absolute lowest power consumption when broadcasting over 2, 3, 4, 6 and 8 channels compared to the other three protocols. The reason for this is that Fast broadcasting is a very front-loaded protocol, meaning that it tries to transfer most of the data in the beginning of the session. However, fast broadcasting is one of the inferior protocols when it comes to startup delay. From this we draw the conclusion that bursting out packets and transmitting data from all channels at the same time is not optimal when distributing a data file to end users. We can also see that there is a very small difference between Pyramid broadcasting and Skyscraper broadcasting in all cases. We conclude that receiving data through one channel at a time while listening and pre-loading data from a second channel is the optimal way for a protocol to work to improve the startup delay and energy usage with mobile devices. There is also a big difference in power consumption when using 3G versus using WiFi. The highest energy used when receiving data using 3G reaches a value of about 60 *Joule*, while WiFi only have a maximum power usage of about 30 *Joule*.

7.1 Future Work

The results in this study could be used in future work. When it comes to developing multicasting systems, choosing the most optimal protocol to use when distributing data is crucial when optimizing the power consumption and keeping the startup delay low. There is still further work to be done such as how 4G/LTE transmissions changes consumption behaviours for these protocols. There are also a wide variation of protocols not analyzed in this thesis that could prove useful in this context. However this study shows that none of the analyzed protocols are optimal from a power consumption to startup delay perspective. There could therefore be further studies to find a protocol which has both a short startup delay as well as low power consumption. Our thesis finds that front loading as much as possible is the most beneficial from a energy consumption perspective. The disadvantage of this approach is the increasingly growing startup delay. We would therefore suggest that further studies look into how to take advantage of Fast broadcastings front loading approach in other protocols such as OPB or Pyramid/Skyscraper. This would be to use the parallelism of FB where multiple channels are downloading at the same time during certain transfer windows. As the penalty of shutting down is huge on 3G we would suggest that few transfer windows are to be used, and those used are to be used in their entirety with the receiver working on multiple channels simultaneously. The structure of the other protocols could give this new solution both good energy consumption as receiver usage is optimized as well as much shorter startup delay as earlier sections becomes small enough to transfer quickly as can be seen in PB or OPB. Therefore, studying the possibilities of merging these protocols would be highly interesting.

Bibliography

[1] N. Carlsson and D. L. Eager. "Content Delivery Using Replicated Digital Fountains". In: *Proceedings IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*. Aug. 2010, pp. 338–348.

[2] Niklas Carlsson, Anirban Mahanti, Zongpeng Li, and Derek Eager. "Optimized Periodic Broadcast of Non-linear Media". In: *Proceedings IEEE Transactions on Multimedia*. 2008, pp. 871–884.

[3] S. H. G. Chan and S. H. I. Yeung. "Client buffering techniques for scalable video broadcasting over broadband networks with low user delay". In: *IEEE Transactions on Broadcasting* 48.1 (Mar. 2002), pp. 19–26.

[4] J. Chen, M. Chiang, J. Erman, G. Li, K. K. Ramakrishnan, and R. K. Sinha. "Fair and optimal resource allocation for LTE multicast (eMBMS): Group partitioning and dynamics". In: *IEEE Conference on Computer Communications (INFOCOM)*. Apr. 2015, pp. 1266–1274.

[5] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2015–2020 White Paper*. URL: `http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html`.

[6] "Encyclopedia of Multimedia". In: ed. by Borko Furht. Boston, MA: Springer US, 2008. Chap. Harmonic Broadcasting Protocol, pp. 280–280.

[7] Jerry D. Gibson. *Multimedia Communications: Directions and Innovations*. Dallas, Texas: Academic Press, 2001.

[8] Phillipa Gill, Liqi Shi, Anirban Mahanti, Zongpeng Li, and Derek L. Eager. "Scalable On-demand Media Streaming for Heterogeneous Clients". In: *ACM Trans. Multimedia Comput. Commun. Appl.* 5.1 (Oct. 2008), 8:1–8:24.

[9] A. Hu. "Video-on-demand broadcasting protocols: a comprehensive study". In: *IEEE INFOCOM*. 2001, pp. 508–517.

[10] Kien A. Hua and Simon Sheu. "Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems". In: *Proceedings. ACM SIGCOMM*. 1997, pp. 89–100.

[11] C. P. Lau, A. Alabbasi, and B. Shihada. "An Efficient Live TV Scheduling System for 4G LTE Broadcast". In: *IEEE Systems Journal* (2016), pp. 1–12.

[12] W. C. Liu and J. Y. B. Lee. “Constrained consonant broadcasting - a generalized periodic broadcasting scheme for large scale video streaming”. In: *Proceedings International Conference on Multimedia and Expo.* July 2003, pp. 805–808.

[13] J. Ma, X. Deng, Y. Liu, and D. Wu. “Power consumption of mobile video streaming under adverse network conditions”. In: *Proceedings IEEE/CIC International Conference on Communications in China (ICCC)*. Aug. 2013, pp. 106–111.

[14] A. Mahanti, D. L. Eager, M. K. Vernon, and D. J. Sundaram-Stukel. “Scalable on-demand media streaming with packet loss recovery”. In: *IEEE/ACM Transactions on Networking* 11.2 (Apr. 2003), pp. 195–209.

[15] J. F. Paris, S. W. Carter, and D. D. E. Long. “Efficient broadcasting protocols for video on demand”. In: *Proceedings. International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems.* July 1998, pp. 127–132.

[16] Sanjoy Paul. *Digital Video Distribution in Broadband, Television, Mobile and Converged Networks: Trends, Challenges and Solutions*. Wiley Publishing, 2010.

[17] Ekhiotz Jon Vergara, Simin Nadjm-Tehrani, and Mihails Prihodko. “EnergyBox: Disclosing the wireless transmission energy cost for mobile devices”. In: *Sustainable Computing: Informatics and Systems* 4.2 (2014). Special Issue on Selected papers from EE-LSDS2013 Conference, pp. 118–135.

A Appendix

The figures presented here in the Appendix are figures created using EnergyBox. These graphs present how much time that is spent in each energy state discussed in Section 2.8. The value on the x-axis is always time.

The first graph shows how much power that is drained over time. The thicker orange parts show when the mobile device is receiving data, while the thinner parts represent the device resting. The second graph indicates which state the device is in during the transmission, and the last graph shows the size of each packet that is being transmitted.

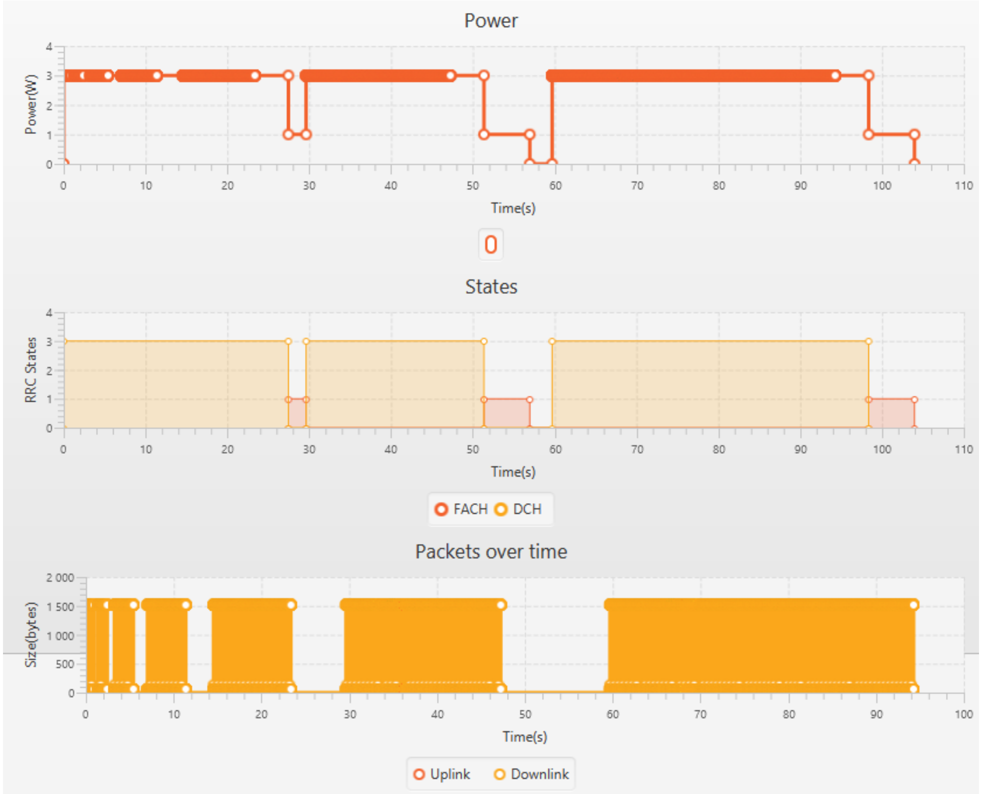


Figure A.1: OPB 2 channels energy state 1000 kbit/s evenly distributed

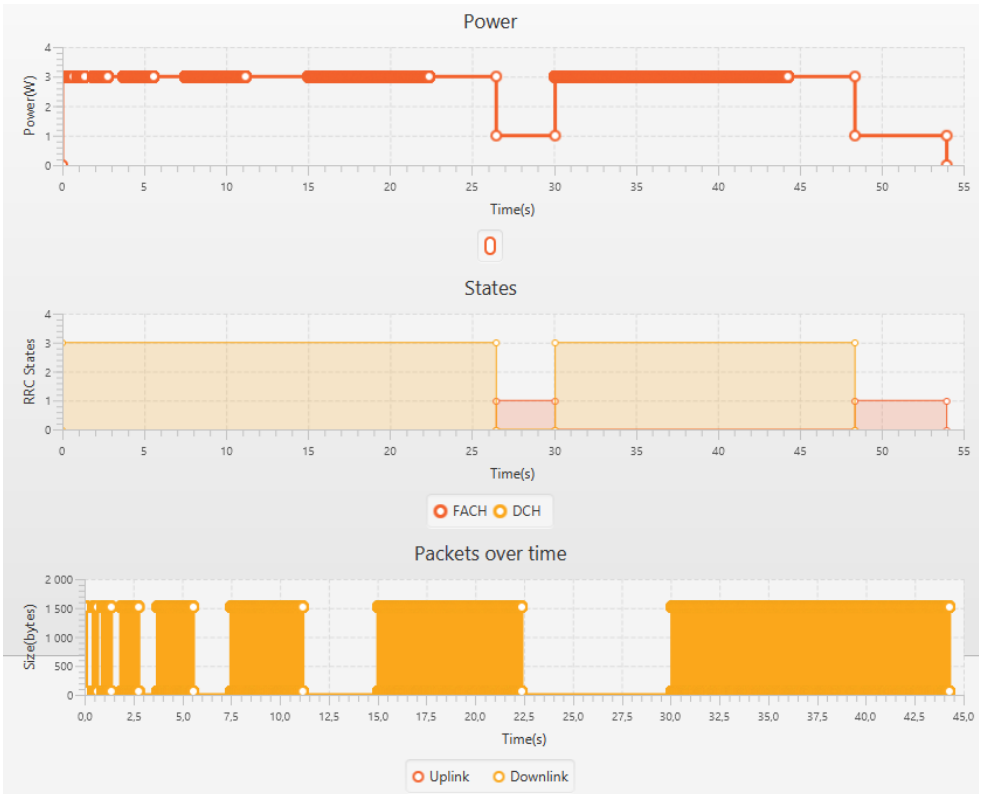


Figure A.2: OPB 3G 8 channels energy state 1000 kbit/s evenly distributed

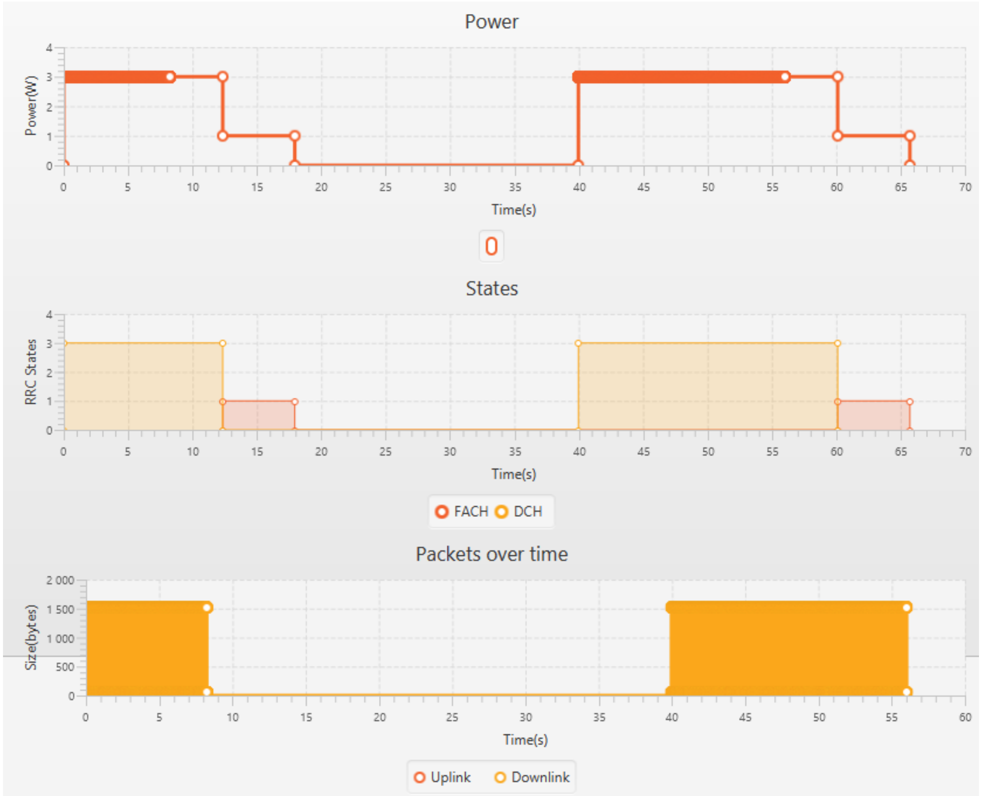


Figure A.3: OPB 3G 2 channels energy state 1000 kbit/s burst mode

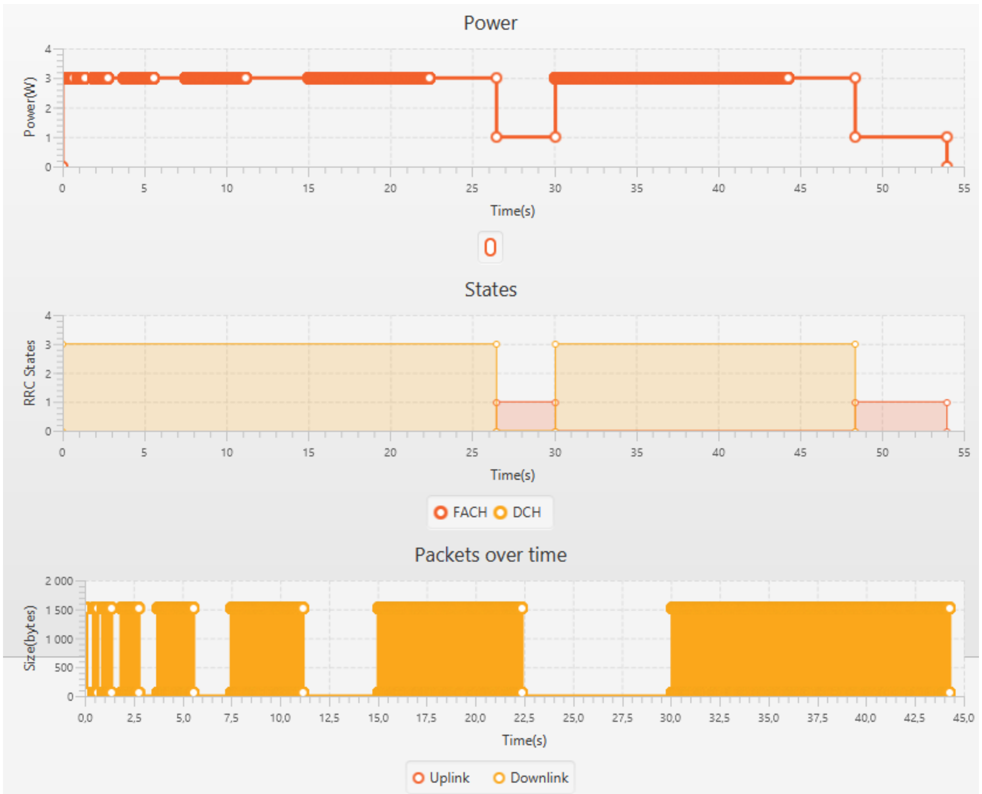


Figure A.4: OPB 3G 8 channels energy state 1000 kbit/s burst mode

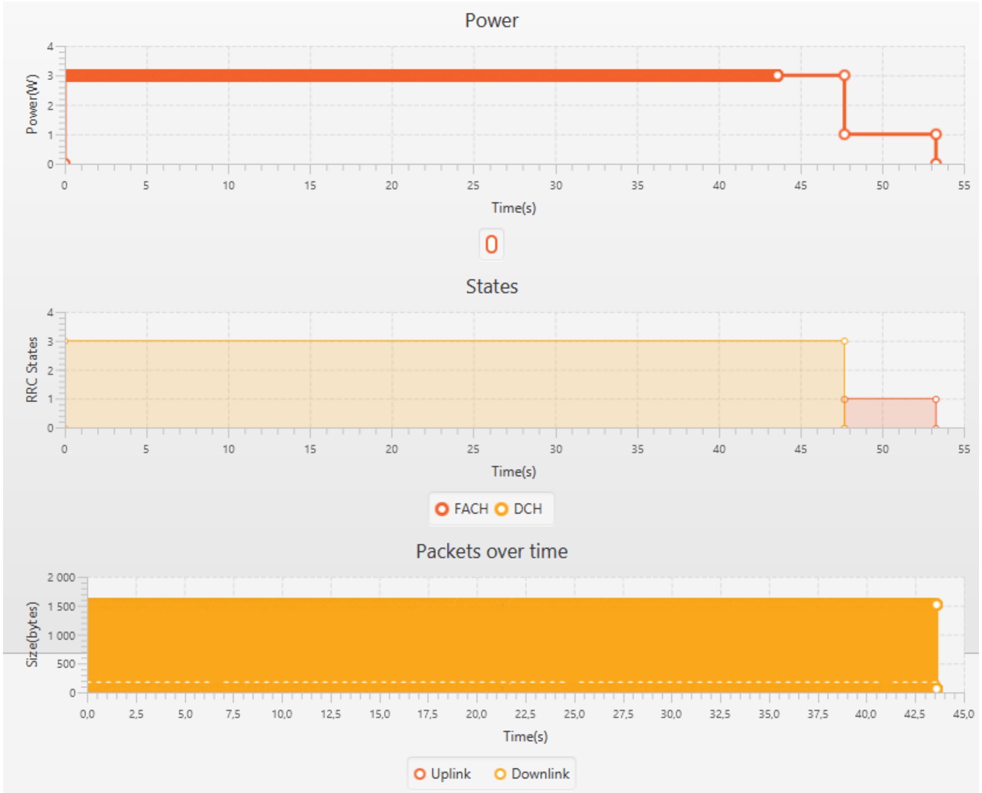


Figure A.5: FastBroadcasting 3G 2 channels energy state 1000 kbit/s evenly distributed

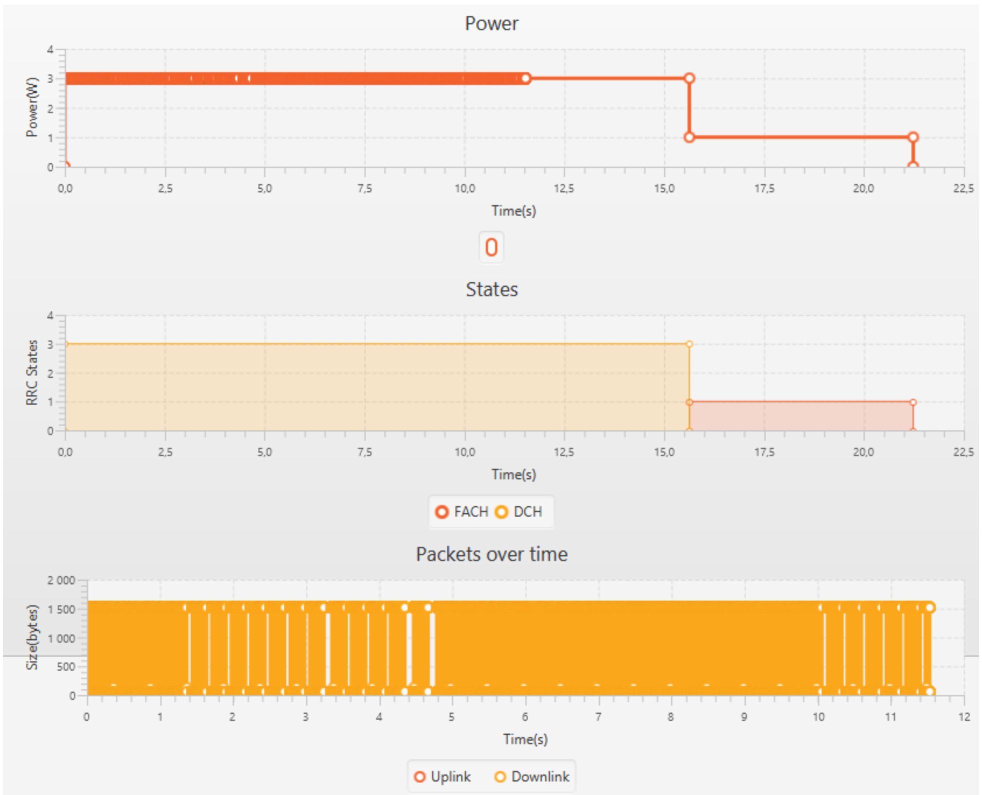


Figure A.6: FastBroadcasting 3G 8 channels energy state 1000 kbit/s evenly distributed

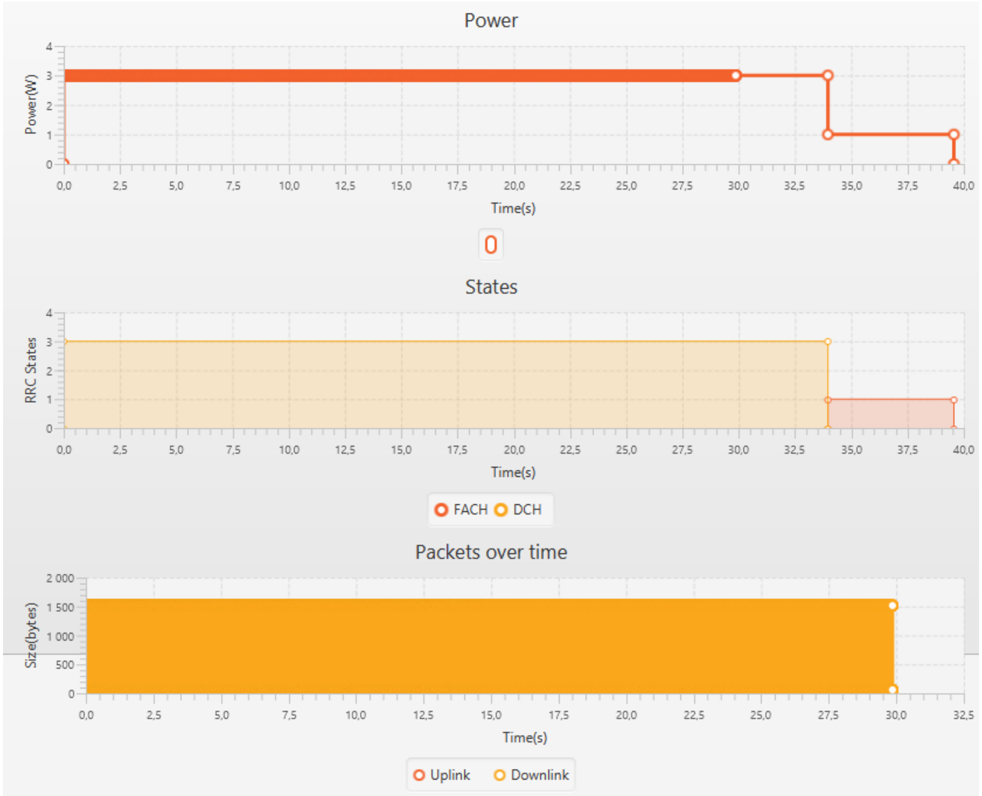


Figure A.7: FastBroadcasting 3G 2 channels energy state 1000 kbit/s burst mode

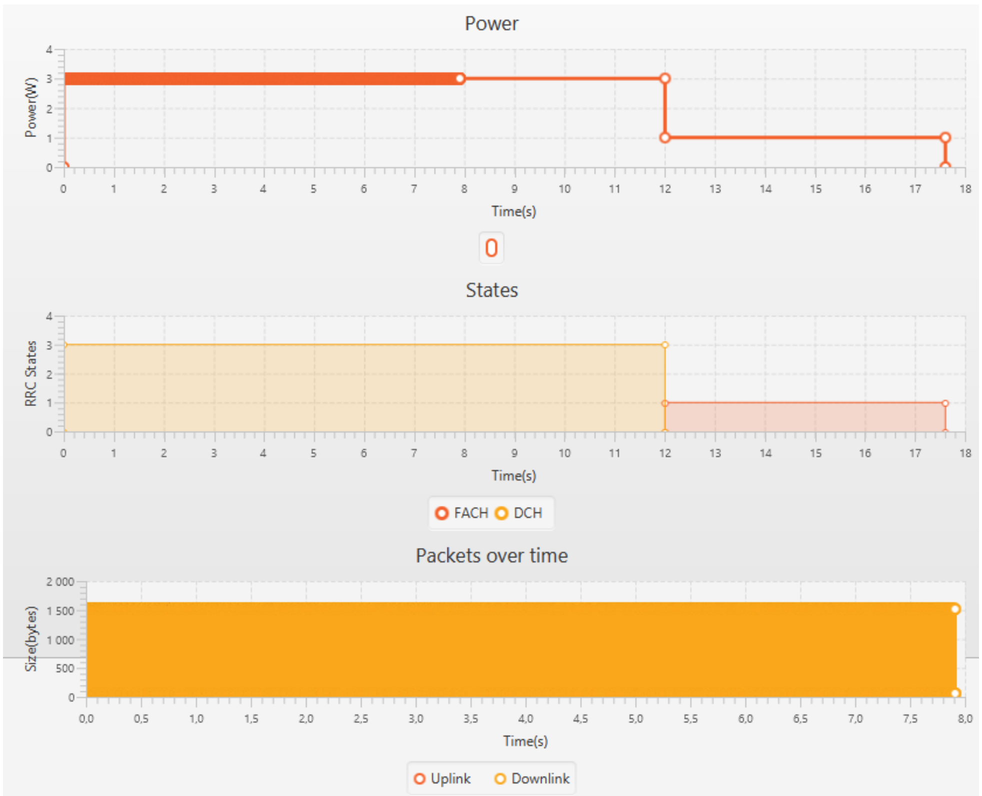


Figure A.8: FastBroadcasting 3G 8 channels energy state 1000 kbit/s burst mode

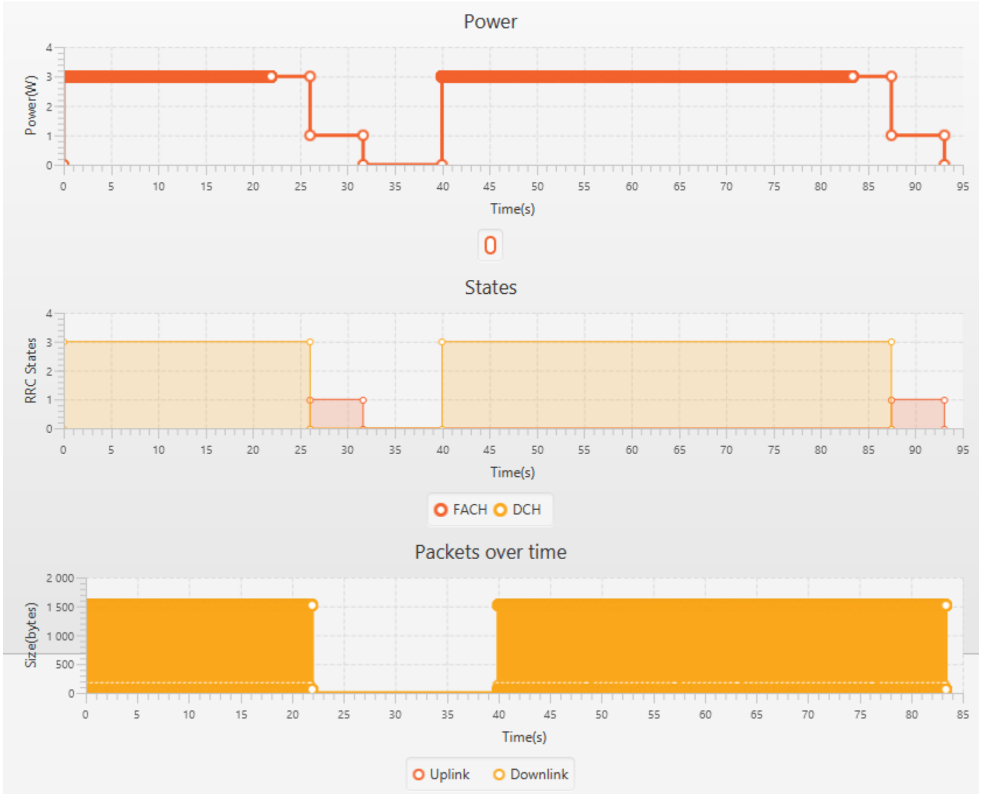


Figure A.9: Skyscraper 3G 2 channels energy state 1000 kbit/s evenly distributed

Figure A.10: Skyscraper 3G 8 channels energy state 1000 kbit/s evenly distributed

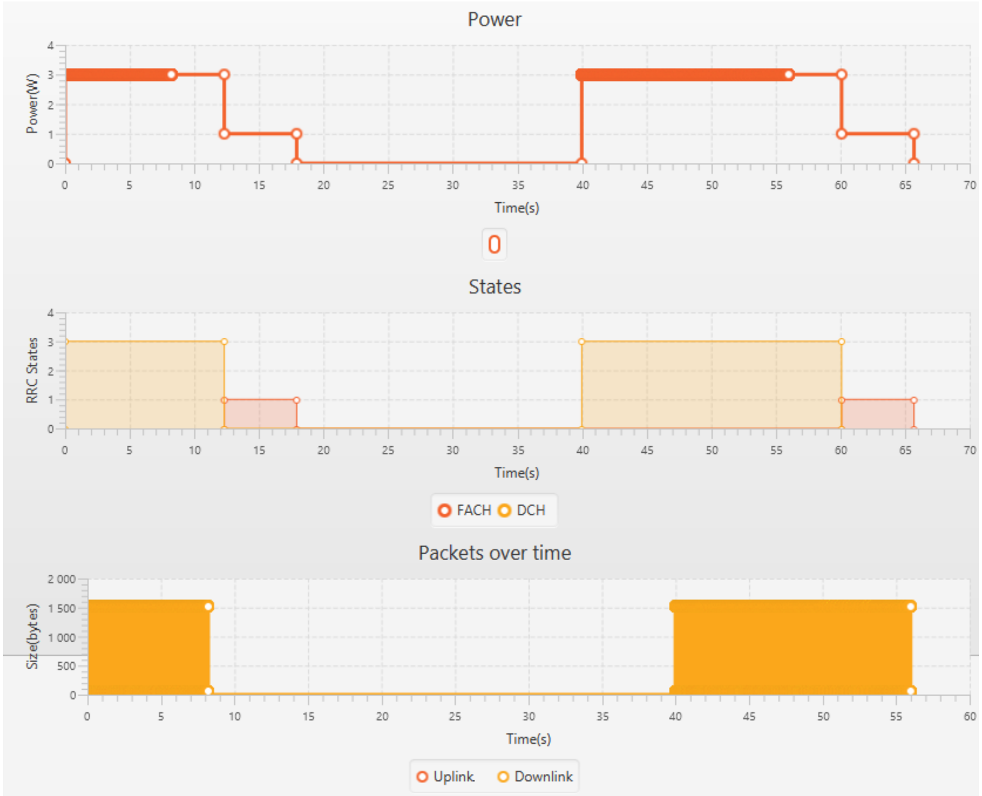


Figure A.11: Skyscraper 3G 2 channels energy state 1000 kbit/s burst mode

Figure A.12: Skyscraper 3G 8 channels energy state 1000 kbit/s burst mode

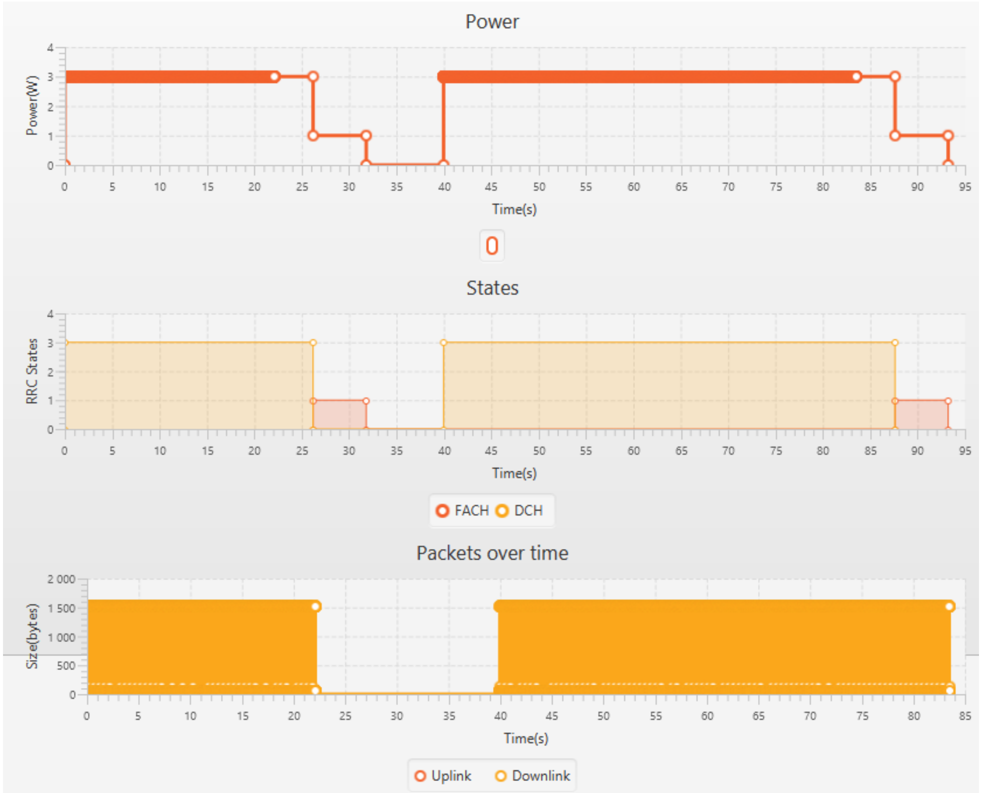


Figure A.13: Pyramid 3G 2 channels energy state 1000 kbit/s evenly distributed

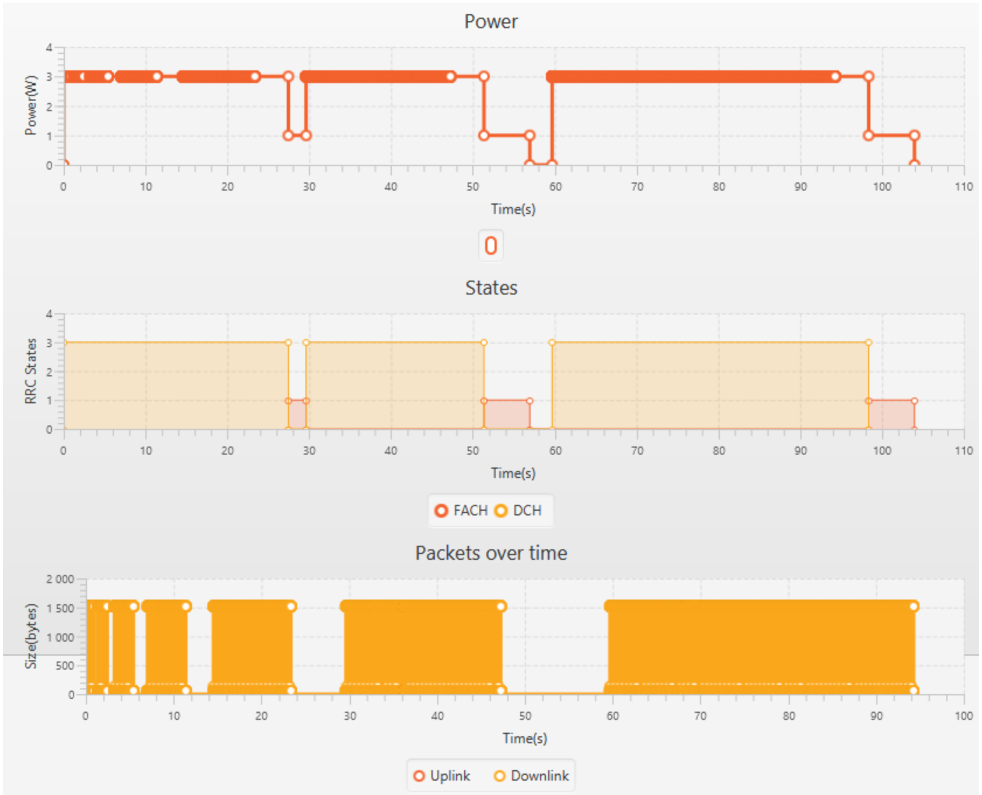


Figure A.14: Pyramid 3G 8 channels energy state 1000 kbit/s evenly distributed

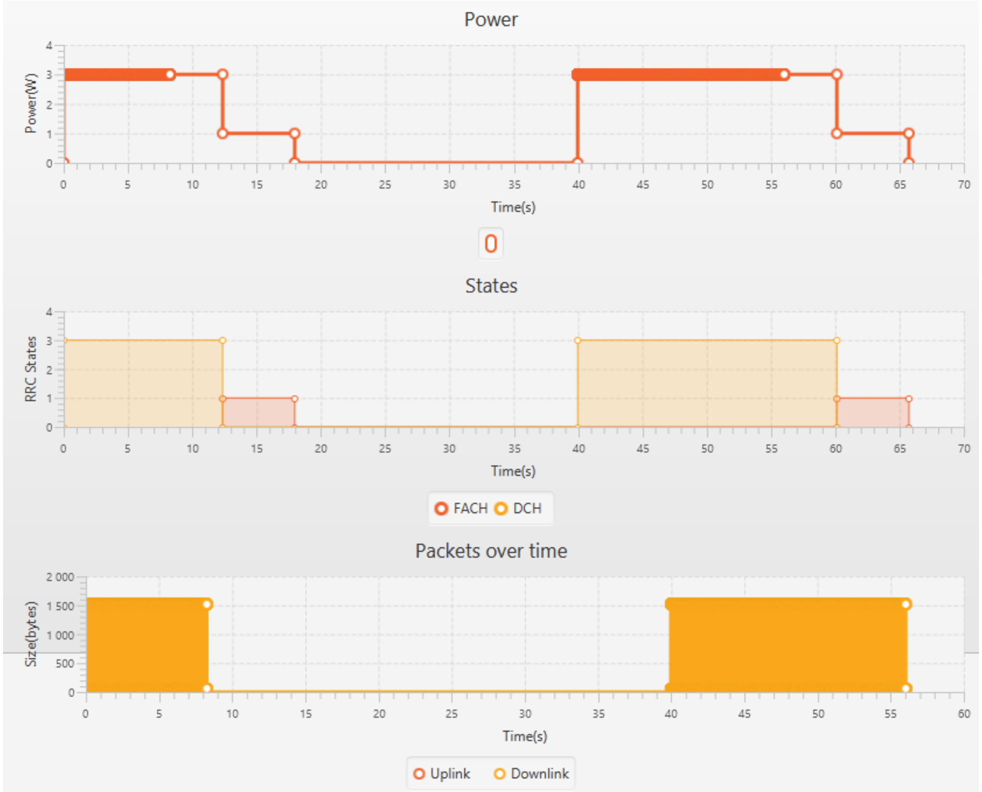


Figure A.15: Pyramid 3G 2 channels energy state 1000 kbit/s burst mode

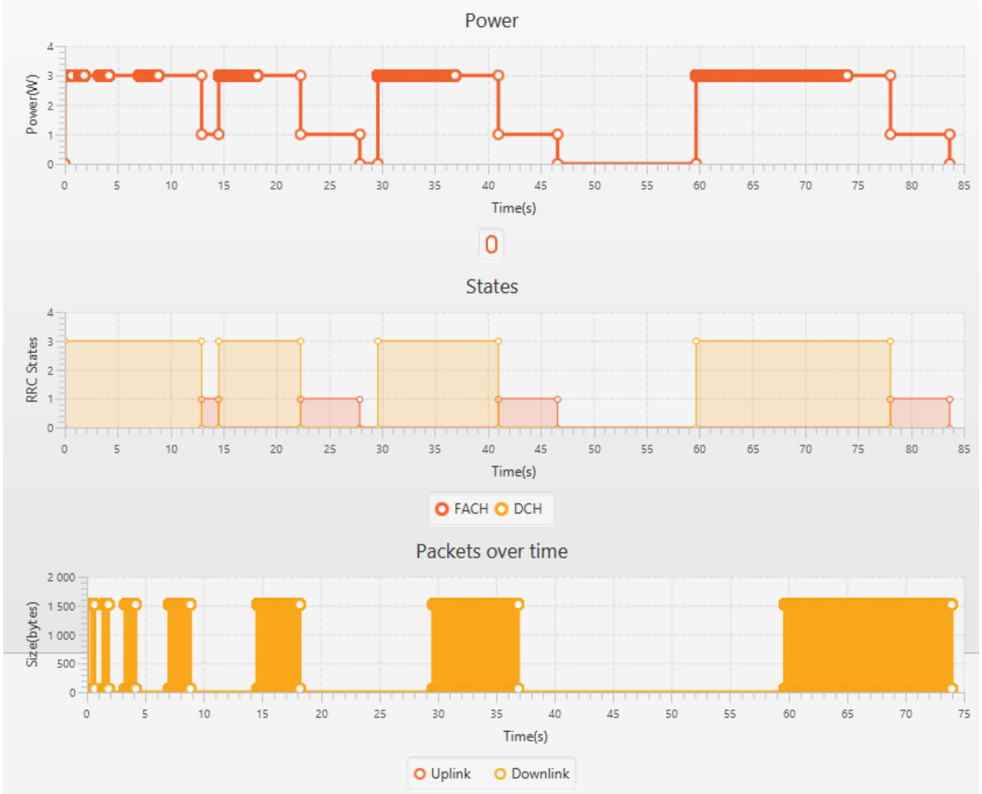


Figure A.16: Pyramid 3G 8 channels energy state 1000 kbit/s burst mode

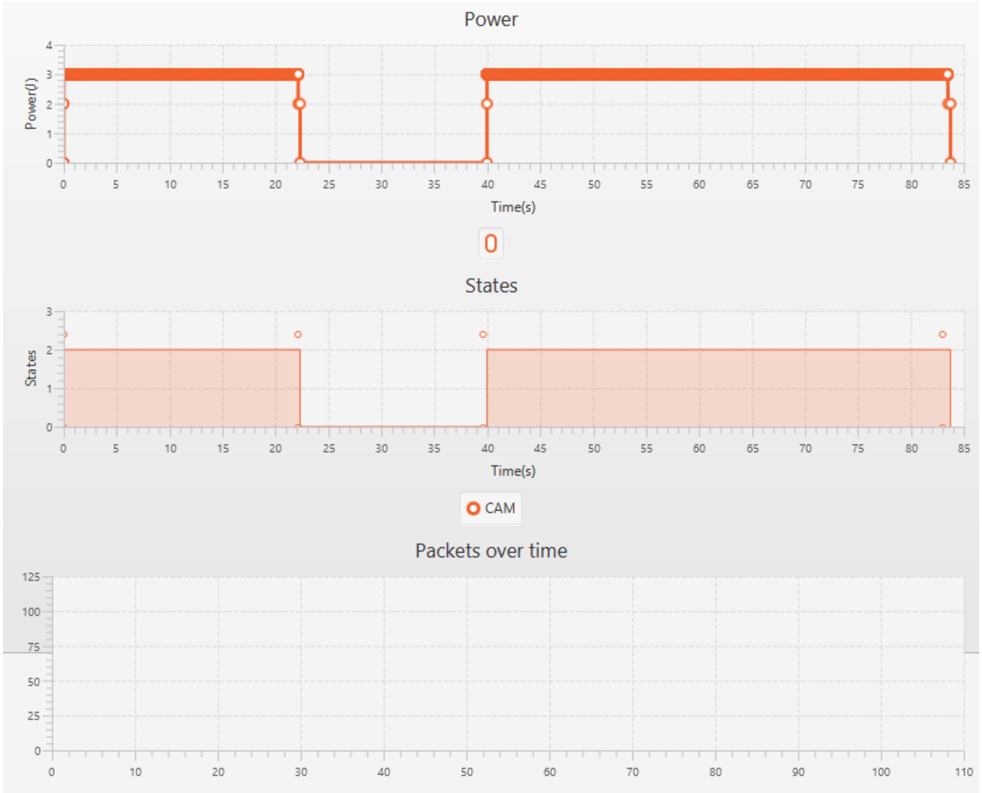


Figure A.17: Pyramid WiFi 2 channels energy state 1000 kbit/s evenly distributed

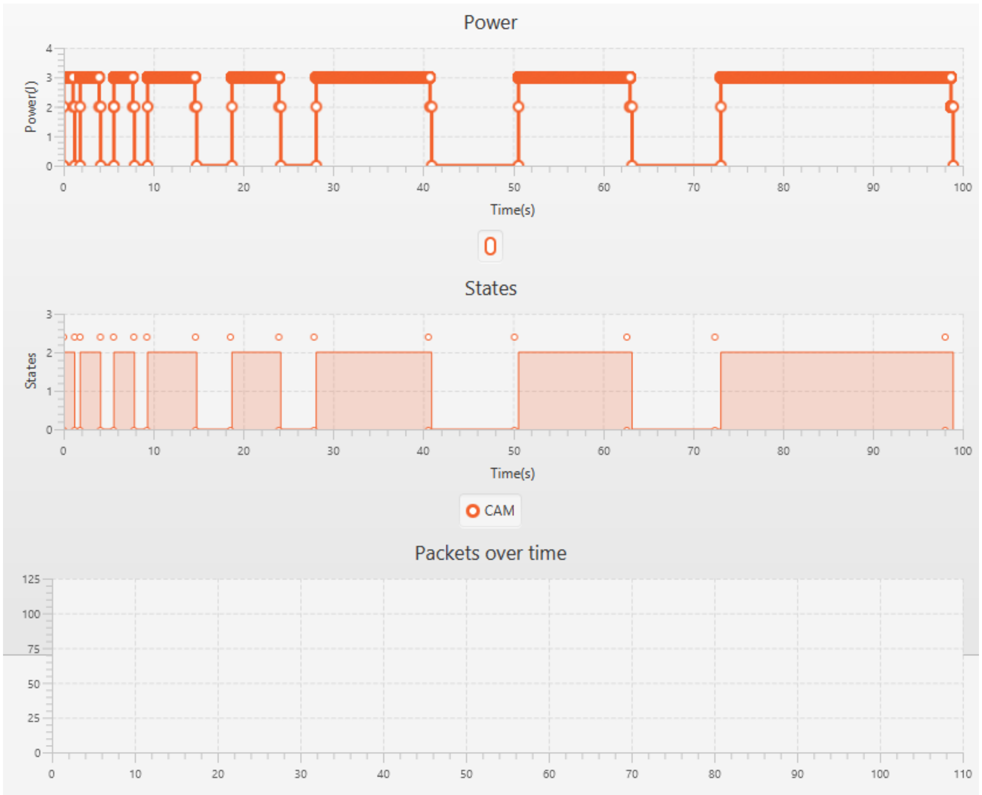


Figure A.18: Pyramid WiFi 8 channels energy state 1000 kbit/s evenly distributed