HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Systems Analysis
Mat-2.4177 Seminar on Case Studies in Operations Research

Optimizing Energy Consumption in Mobile BitTorrent Networks

Subject 1

Project Report

29.4.2009

Client:
Nokia Research Center
Dr. Jukka Nurminen

Project Manager:
Marko Kotilainen

Project Group:
Turkka Anttonen
Lars Kurkinen
Ilkka Mansikkamäki
Jukka Ylitalo
Jutta Ylitalo

Abstract

This research project examined the optimization of energy consumption in mobile BitTorrent networks. The energy consumption is an important factor for mobile devices as it sets operational restrictions. The goal for the project was to derive heuristic rules for the optimal sharing behavior. The sharing in BitTorrent networks was examined with mixed-integer linear programming (MILP) model that represents ideal situation.

Two cases were used for the analysis. In the first case, we analyzed situations where the download bandwidth is significantly larger than the upload bandwidth. In the second case, download bandwidth equals upload bandwidth. Analysis was made by examining the number of devices possessing each file part; the number of file parts in possession of each device; average upload and download speeds; the total number of file parts in the network; activity time for devices, and capacity utilized.

In the first case, we made the following observations: an active device uploads at maximum speed, maximum upload speed requires some planning and energy consumption increases linearly as the number of devices increase. In the second case following observations were made: total activity time was more evenly distributed than in the first case, active devices reached maximum utilized capacity slower than in the first case but it was sustained to the end of sharing; sharing behavior seemed to form block of files, and initial seed remained active only the first time steps that were required to upload all of the file parts to network.

Table of Contents

1 Introduction

1.1 Background

BitTorrent is a file sharing protocol. It is more efficient than traditional file transferring systems because files are transferred directly from one user to another. In a BitTorrent network, there is first one device with a file which is to be shared to all others. The device that has the whole file and is sharing it, in other words uploading the file, is called a seed. Device that is receiving or downloading is called a leech. Furthermore, a device possessing some file parts but not the whole file is called a peer. The shared file is divided into small pieces (file parts) and these pieces are then sent to leeches in the network. After the device has downloaded a file part, it starts to upload the file part to different users. This way the size of the network expands rapidly and several peers are uploading and downloading file parts simultaneously. Each device has some restrictions, for instance how fast it can download or upload from the network. (Cohen 2008)

BitTorrent was originally developed for PC network and it has been designed to minimize time consumption so that all peers would get the file as quickly as possible. However, BitTorrent can also be used in mobile networks. In such a case, energy becomes an important factor as batteries restrict the energy available.

BitTorrent has been widely researched mostly focusing on time consumption or the theoretical behaviour of the network. Energy consumption has been investigated without any proposed solutions concerning the sharing behaviour (Nurminen and Nöyränen 2008).

1.2 Objectives

The purpose of this project was to research the optimal way to share a file in the BitTorrent network so that energy consumption of devices is minimized. The objective of our project was to find heuristic rules on how to reach the optimal energy. The idea was to examine the optimal energy consumption behaviour in different sized networks with different parameters.

Optimal energy in this project is described as the minimum amount of time steps which devices need to stay in total in the network to receive the whole file. We approached the problem by creating a linear optimization problem which was implemented with Matlab.

We used a Matlab interface to solve the problem with Xpress-MP, a program developed to solve large-scale linear problems. Using Matlab gave us better means for interpreting and analyzing the results.

2 Model

This chapter describes the characteristics of the used model. First we consider the realized assumptions. Section 2.1 describes the implemented model in the form we finally left it. Section 2.2 reviews the alternative models that were considered to enhance the model and reduce computational effort but did not actually work. Finally, in section 2.3, we revise some of the issues that emerged in the modelling process.

The main objective of the model is to optimize energy consumption of mobile devices. The following assumptions are made to simplify the model:

1. All participating devices are energy-sensitive.
2. In idle phase power consumption is about zero. A device is in idle phase when it does not download or upload anything.
3. In active phase power consumption is constant and does not depend on the download and/or upload speed. A device is in active phase when it uploads or downloads at least one file part during the considered time step.
4. No devices enter or exit the network during the process.
5. In the considered network, there is an initial seed and no other devices possess any file parts in the beginning.
6. All devices follow a common strategy to share the file.

The assumption 3 is close to reality as an increase in download bandwidth increases energy consumption only slightly. Likewise, parallel uploading increases energy consumption only slightly according to the client of the project.

The assumption 6 means that a device may not optimize its own behaviour, for example, choose when to download or upload. Instead, the network is optimized as a whole.

The course staff, Antti Punkka and Juuso Liesiö, provided us with a model that we could use as a starting point in our work. The model considers the whole network. This means that if the same file part is uploaded and downloaded by two devices at the same time step, the model does not take a stand on which of the uploading devices uploaded the file part to which downloading device.

Optimization provides only one of the solutions that lead to the optimal energy consumption. Usually there are several solutions that give the same optimal energy consumption.

2.1 Final Model

2.1.1 Parameters

d_i = download bandwidth (file parts per time step) of the device *i*

u_i = upload bandwidth (file parts per time step) of the device *i*

The number of mobile devices, *I*, the number of file parts, *K*, and the number of time steps, *T*, are also given to the optimization algorithm.

2.1.2 Variables

f(i,k,t) indicates whether the device *i* has file part *k* at time *t* (1) or not (0)

r(i,k,t) indicates whether the device *i* downloads file part *k* at time *t* (1) or not (0)

s(i,k,t) indicates whether the device *i* uploads file part *k* at time *t* (≥1) or not (0), integer variable

z(i,t) ϵ {0,1} indicates whether the device *i* is active at time *t* (1) or not (0)

2.1.3 Constraints

The same amount of file parts is downloaded and uploaded during every time step:

$$\sum_i r(i,k,t) - \sum_i s(i,k,t) = 0 \;\forall k, t \tag{1}$$

At a time step, a device has a file part if it already had it at the previous time step or if it downloaded the part during previous time step:

$$f(i,k,t) - f(i,k,t-1) - r(i,k,t-1) = 0 \;\forall\; i, k, t \tag{2}$$

A device may upload file parts only if it is active, and even then its upload bandwidth cannot be exceeded:

$$\sum_k s(i,k,t) - z(i,t) * u(i) \leq 0 \;\forall i,t \tag{3}$$

A device may download file parts only if it is active, and even then its download bandwidth cannot be exceeded:

$$\sum_k r(i,k,t) - z(i,t) * d(i) \leq 0 \;\forall i,t \tag{4}$$

A device can upload a file part only if it already has it. If the device has the file part, it can upload it at maximum at its upload bandwidth.

$$-u(i) * f(i,k,t) + s(i,k,t) \leq 0 \;\forall\; i,k,t \tag{5}$$

In the end, each device has to have all file parts:

$$\sum_t r(i,k,t) + f(i,k,0) = 1 \;\forall i,k \tag{6}$$

2.1.4 Adjusted Constraints for Special Cases

Some of the mentioned constraints are changed in some special cases described below.

If the first device already has all the file parts at the beginning, it never downloads anything. So the constraint (4) becomes:

$$\sum_k r(i,k,t) - z(i,t) * d(i) \leq 0 \;\forall t\,, i \geq 2 \tag{7}$$

In addition, if no other device has any file parts, they cannot upload anything at the first time step. So the constraint (3) changes as follows:

$$\sum_k s(i,k,t) - z(i,t) * u(i) \leq 0, \text{when } i = 1, \forall\; t \tag{8}$$

$$\sum_k s(i,k,t) - z(i,t) * u(i) \leq 0, \text{when } i \geq 2 \text{ and } t \geq 2 \tag{9}$$

In these special cases, constraints are changed this way to decrease the amounts of variables and constraints in order to make optimization faster.

2.1.5 Objective Function

We ended up using an objective function that forces all the empty time steps in the solution to the end. The objective function also favours solutions that use the minimum number of time steps. It penalizes solutions in which the active time steps of different devices are divided to longer time as the coefficient of the active time step of a device is larger when it occurs later. This makes sure the trade off between minimizing time and minimizing energy consumption does not appear to be larger than it actually is. Still, minimizing the energy consumption stays as the primary objective because its coefficient is much larger than the coefficient of the time minimization in the following function:

$$\min \sum_i \sum_t (1000 + 0.1 * t - 1)) * z(i,t) \tag{10}$$

2.2 Alternative Models

We also tried a different version of the model in which the variables *f(i,k,t)* were deleted. This reduces the number of variables and thus should reduce the computation effort. Instead, we replaced the variables *f(i,k,t)* with the sum of downloaded file parts thus far as a device can only possess a file part if it had it in the beginning or if it has downloaded it before the time step in question:

$$f(i,k,t) = f(i,k,0) + \sum_{\tau=1}^{t-1} r(i,k,\tau) \;\; \forall i,k,t \tag{11}$$

Contrary to what was expected, this change made the optimization slower so the change was discarded.

Originally we had an objective function that just summed the number of all the active devices at each time step:

$$\min \sum_i \sum_t z(i,t) \tag{12}$$

The following constraint was added to the model. It enforces possible empty time steps to the end of the time period. This also decreases the number of feasible solutions which in turn should reduce optimization time. The following constraint was added to the model:

$$\sum_{i} z(i,t) - \frac{1}{I+1}\sum_{i} z(i,t+1) \geq 0 \ \forall t = 1,\dots,T-1 \tag{13}$$

The objective function and constraint described above were replaced by the final objective function (10).

In addition, we tried to change some of the variables continuous. Theoretically, this should speed up computation. In practice, the computation was slower although several changes were made at the same time so it was impossible to say which of the changes caused the increase in computation time. Due to restricted time that was available for the project, it is impossible to say if the computation time could be decreased by changing the types of the variables.

We also tried to add upper bounds for variables as most of the variables may not be greater than one. These new constraints decrese the number of feasible solutions. It is also possible to diminish interchangeability of devices by choosing which devices downloaded which file parts, for example, during first time step. This is one way to decrease the number of optima that have the same value. For example, if devices 2, 3, and 4 are chose to download file parts 1, 2, and 3 during the first time step, the algorithm does not have to check if it was able to find a better optimum for example by changing the mentioned devices to 5, 6, and 7 or the mentioned file parts to 4, 5, and 6. In practice, we had same problems as with changing variables continuous: in the end, we do not know which change alone or which changes together were the reason for the increase in optimization time.

2.3 Choosing Parameters

The purpose was to model an ideal situation as well as possible. For instance, we calculated that if we wanted to examine the distribution of a normal music file, about 3-4 Mbit/s, we would have to have about 50 file parts. One raised question was that what are the normal upload and download bandwidths. After discussing with the client and researching the bandwidths in the Internet (Nokia 2009) we decided to set the upload bandwidth to 6 file parts per time step which would mean 384 kbit/s and download bandwidth to 60 (3849 kbit/s).

After running the first version of our Matlab code, we discovered that we needed to reduce our parameters. The memory capacity in the computer we had in use was not sufficient for problems of this scale. Either we would have a network of three devices, or the relations between different parameters would be corrupted. So we decided to decrease the number of file parts in half as well as the bandwidths. This way we were able to have a network of about ten devices and 20 file parts. Each time we ran the model with different parameters we determined the number of time steps separately guessing and later by using heuristic rules we learned during the work.

We could have started with even smaller upload and download bandwidths. Still, it is reasonable to keep upload and download bandwidths larger than one so that a device can simultaneously upload to several other devices and download from several other devices. Actual networks work this way, and we wanted to include this feature in the model. A problem with discrete time is that, for example, if a device downloads three file parts from another device during one time step, it may not upload any of those file parts during the same time step even though it takes only a fraction of the time step to download the first file part. Another reason to choose upload and download bandwidths of this magnitude was that smaller bandwidths make the problem larger and we had problems with computation capacity.

Another simplification was that only one device served as an initial seed. No other device had any file parts in the beginning. Schedule problems led to this simplification. There was not enough time to examine situations with multiple initial situations properly and choosing one initial situation as the basis for analyzing served in making the different cases comparable to each other.

3 Results and Analysis

In this section we present and analyze the results our model has produced. First, in section 3.1, we study file sharing problems in homogenous networks where the download bandwidth is effectively unlimited. Sections 3.1.1-3.1.6 study different characteristics of the network. In section 3.1.8 we summarize our results and draw conclusions. In section 3.1.7 we perform some further analysis through a simple theoretical approach. The heuristics we derive are presented in section 3.1.9. In section 3.2 we study file sharing problems in homogenous networks in which upload bandwidth is equal to the download bandwidth. As in the previous case, sections 3.2.1-3.2.6 study different characteristics of the network. Summary, conclusions and heuristics are developed in sections 3.2.7– 3.2.9.

The results are illustrated with supporting figures and tables from individual cases that we studied. If nothing else is mentioned these figures and tables illustrate general patterns that were found in all studied cases. The initial and final situations are the same in all cases; at the first time step, the initial seed is the only device to hold all file parts, whereas no other device holds any file parts. At the final state, all devices have all of the file parts.

3.1 Networks with Significantly Larger Download than Upload Speed Limits

In the first case, we study a homogenous network of energy sensitive devices. The upload bandwidth of the devices is fixed, and the download speed of the devices is effectively unlimited (in the future, these cases are also referred to as “ul<<dl” cases). In practice, however, the download speed is limited by the available upload capacity in the network. For example, in a network consisting of 10 devices with an upload bandwidth of 3 each, the maximum speed a single device can achieve is 27. In practice, the download speeds of the devices in the network remained significantly smaller than this, very rarely exceeding two times the upload bandwidth of a device. This case is thus close to real world cases where it is common that maximum download speeds of devices stand between 2 and 10 times the magnitude of the maximum upload speed. For example, the Nokia N95 has, according to the product specifications, a download bandwidth roughly 10 times the maximum upload bandwidth (Nokia 2009).

The behaviour of the network is thus very dependent on the upload bandwidth of the devices. As only one device holds the whole file in the beginning (the initial seed), even the time it takes for this device to upload the whole file into the network is a clear limitation of how quickly the file sharing can be completed. In most cases we studied, the upload bandwidth was set to be three file parts per time step. It is important to note that even the way the upload bandwidth is selected changes some characteristics of the file sharing process. As we chose a upload bandwidth of three, this implies that a device can only send file parts to three devices at a given time step.

3.1.1 Number of Devices Possessing Each File Part

First, we studied how the number of devices in the network that possess the file parts develops over time. Figure 1 illustrates how common the different file parts are in the network. We can for instance observe that the first three file parts that are uploaded into the network become quickly very common in the network. In the case that Figure 1 illustrates parts one to three are held by 9 devices after time step 4, and by all 10 devices after step 5. The same happens with subsequent file parts as well. After a file part is uploaded by the initial seed into the network, it becomes within a few time steps held by the majority of the devices in the network.

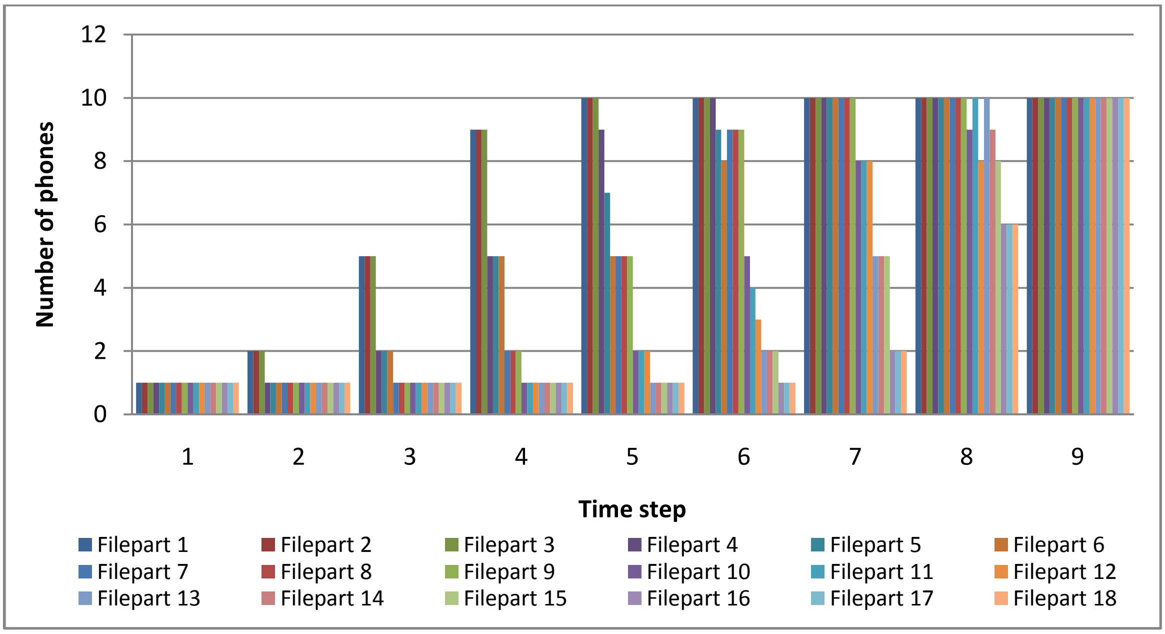

Figure 1: Number of devices possessing each file part at the beginning of each time step, case with 10 devices, 18 file parts and 8 time steps.

3.1.2 Number of File Parts in Possession of Each Device

Figure 2 illustrates a typical example of how the number of file parts each device holds develops over time. From Figure 2 we can observe a characteristic common for all ul<<dl cases: the initial seed shares the file to three devices. Intuitively one may expect the seed to only upload file parts to one device at a time at the maximum speed. If we consider the first time step only, separate from the rest of the sharing process, it would be more energy efficient to upload only to one device. This would mean only two devices are active, compared to the actual case where four devices are active. However, as the sharing happens to multiple devices from the start, it indicates that there may not be large differences in completion times of the energy optimal sharing compared to the "time optimal" sharing.

Generally the devices download the last parts of the file at approximately the same time. The first devices to complete the download of the whole file do this at time step 8, followed by the rest of the devices at time step 9. The file is thus shared quite "fairly" throughout the network with no clear "winners" or "losers" with regard to completion of the download. In the case of Figure 2, the initial seed has sent out all parts of the file at time step 7, which again limits the time when the devices can complete their downloads.

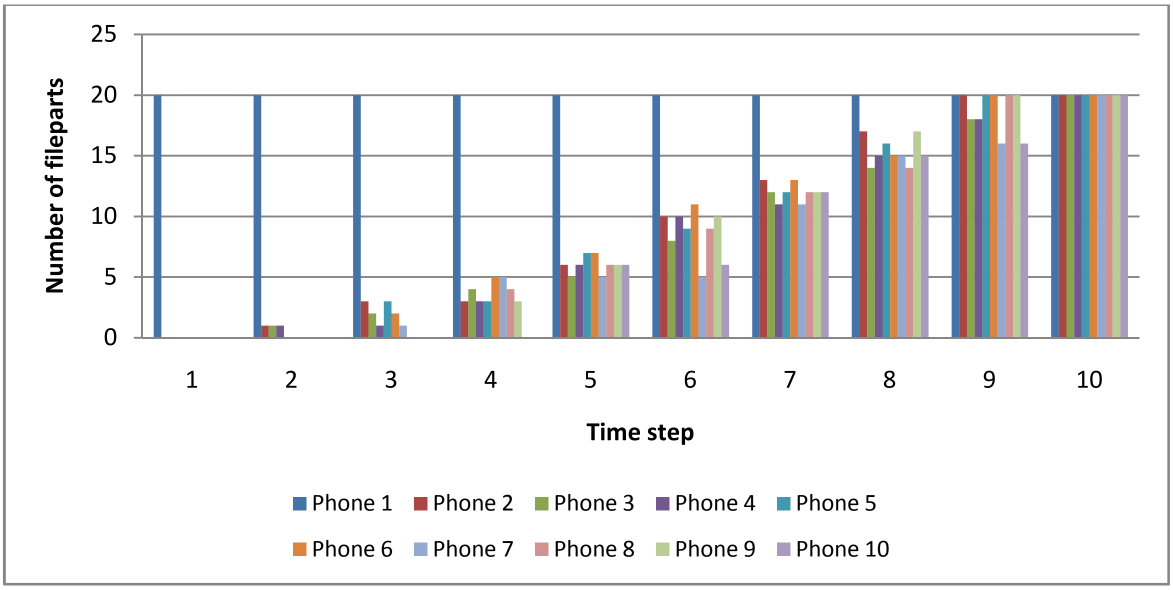

Figure 2: Number of file parts in possession of each device at the beginning of each time step, model with 10 devices, 20 file parts and 9 time steps.

3.1.3 Average Upload and Download Speeds

Table 1 presents the key parameters of the upload and download activity, for example, average speeds and standard deviations of three cases that we studied. The case code 10-20-8 refers to the case with 10 devices, 20 file parts, and 8 time steps. The number of upload time steps is in the three cases of Table 1 very close to the number of download time steps. This does, however, not imply that devices download always when they upload. During the first active time step of a device the device downloads its first file part(s) from the network without being able to upload anything. Also, the initial seed never downloads, but naturally has to upload all file parts. The time steps during which devices have an upload speed of zero have been disregarded in the calculation of average upload speeds and standard deviations. The same has been done in the calculation of the download speed parameters.

A central characteristic for energy optimal solutions seems to be that the upload capacity is always utilized to its maximum, as it is the main bottleneck of the system. In all the cases we studied, the upload bandwidth was always utilized to its maximum when a device was active. An exception occurs if the total number of file transfers needed is not evenly divisible with the upload bandwidth. For example, in the case of 11 devices and 20 file parts the total number of file parts that needs to be downloaded is 10·20=200, which is not evenly divisible with the upload bandwidth three. In this case, one device uploads at the speed of two during one time step. In other cases, the average upload speed equals three and the standard deviation is zero, implying that always when a device uploads, it uploads at full speed. The standard deviation of the download speed is around 1.5 in these cases, which reflects the quite large variations that occur in the download speeds of the individual devices.

Table 1: Key parameters of the upload and download speeds.

	Case 10-20-8	Case 10-20-9	Case 11-20-10
Number of upload time steps	54.00	60.00	67.00
Average upload speed	3.00	3.00	2.99
Standard deviation of upload speed	0.00	0.00	0.12
Number of download time steps	55.00	60.00	67.00
Average download speed	2.95	3.00	2.99
Standard deviation of download speed	1.66	1.37	1.47

3.1.4 Total Number of File Parts in the Network

Figure 3 illustrates how the number of file parts in the network evolves over time. This figure is essentially equal to taking the sum of how many file parts each device possesses at each time step, which was presented in Figure 2. An s-shaped curve can be seen in Figure 3. The slope of the curve tells us the general pattern of file sharing in the network: after a slow start, the file sharing speeds up to a maximum, to again slow down in the end of the sharing process. One thing that limits the upload speed in the start is the number of devices that hold file parts, i.e., are able to upload. Another limitation is the upload speed of the devices, especially the initial seed. Larger upload bandwidths have two consequences. First, and more importantly, the seed can share file parts to more devices at once, which leads to reaching the maximum total upload capacity more quickly. Second, the total upload bandwidth of the network increases.

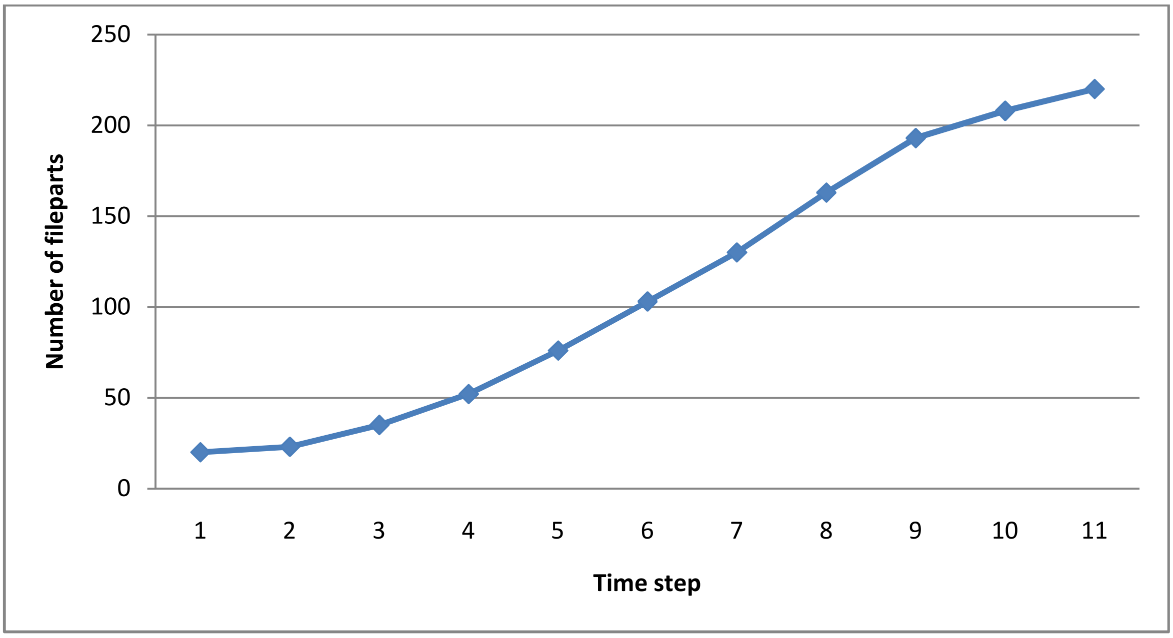

Figure 3: Total number of file parts in the network, model with 11 devices, 20 file parts and 10 time steps.

3.1.5 Activity Time

The activity time of a device refers to the number of times steps during which the device uploads and/or downloads. When considering energy sensitive devices, it is in interest of each individual device to complete the download in as few active time steps as possible. Our model, however, takes a holistic perspective of the energy consumption of the network and thus does not favour cases in which the activity times would be evenly distributed between the devices. Nevertheless, it is clearly of interest to examine what happens with activity times on the individual level of the devices when the total energy consumption is at its minimum.

In Figure 4, a pattern, which occurred in all the cases we studied, can be observed: the devices that are active early on have the highest total activity times, while the devices which have their first downloads last have the lowest total activity times. There are, however, exceptions to this. For instance, in Figure 4 we can observe that device 6 is active in the first two time steps but despite this manages to get away with a lower-than-average activity time. This highlights the fact that the results we are studying are just single representations from the substantially larger group of equally good optimal solutions, and exceptions do exist. Nevertheless, on a general level, early activity of a device leads to a higher total activity time for the device.

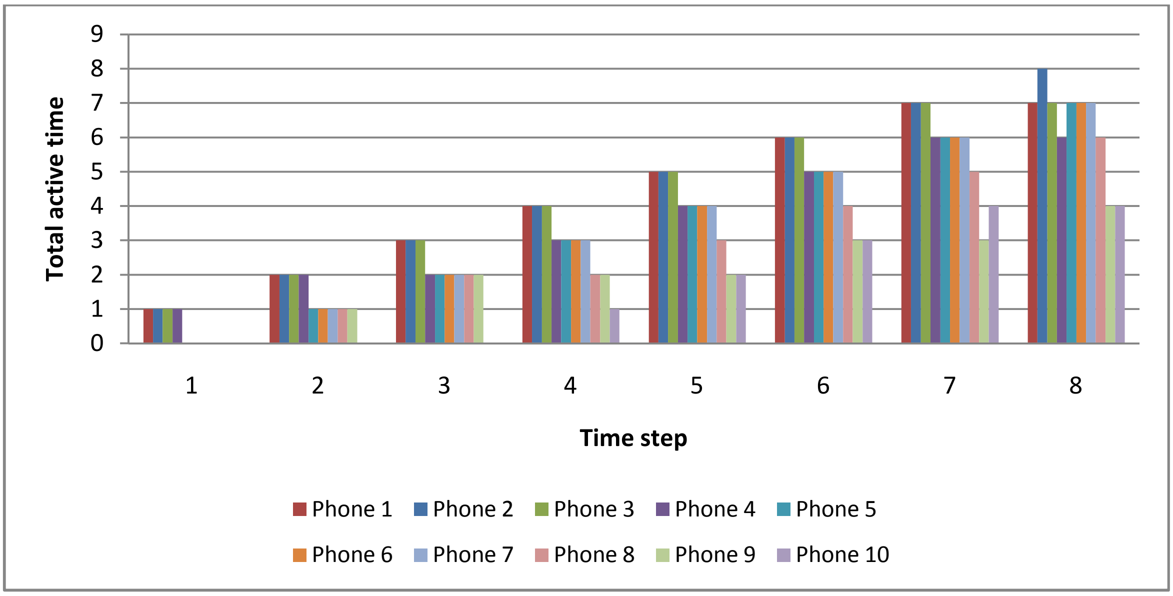

Figure 4: Cumulative activity time for each device, model with 10 devices, 18 file parts and 8 time steps.

3.1.6 Capacity Utilized

The capacity utilized measure reflects how much of the upload bandwidth is utilized in the network. We have already noted that the devices always upload at maximum speed when awake. The *potential* capacity, given the number of active devices at a time step is thus used to 100% in each time step. When the utilized upload capacity of active devices is less than 100%, this reflects that file parts are being sent to a device for the first time (and the device is hence not yet able to upload file parts). Exceptions arise when the total number of file parts is not equally divisible with the upload speed, as discussed in section 3.1.3.

In the analyzed cases, the capacity utilized of all devices stayed between 80% and 100% once new devices had stopped entering the network. 100% of the capacity of all devices was in all ul<<dl cases reached, but not for more than one or a few time steps.

Our model favours solutions that require fewer time steps to complete. It does, however, not differentiate between solutions that complete the task as quickly. Thus, there are often solutions that may be considered more efficient than the solutions we studied, as they achieve a higher percentage of completion earlier on. For example in cases where the total number of file parts is not equally divisible with the upload speed, the upload speed of a device is below maximum at some time step. In the most efficient solution we would expect the device to upload at below maximum speed at the last time step, but in our current solutions the “sub-maximum upload step” seems to occur at a random time. However, the download could not be energy optimally completed in a smaller number of time steps.

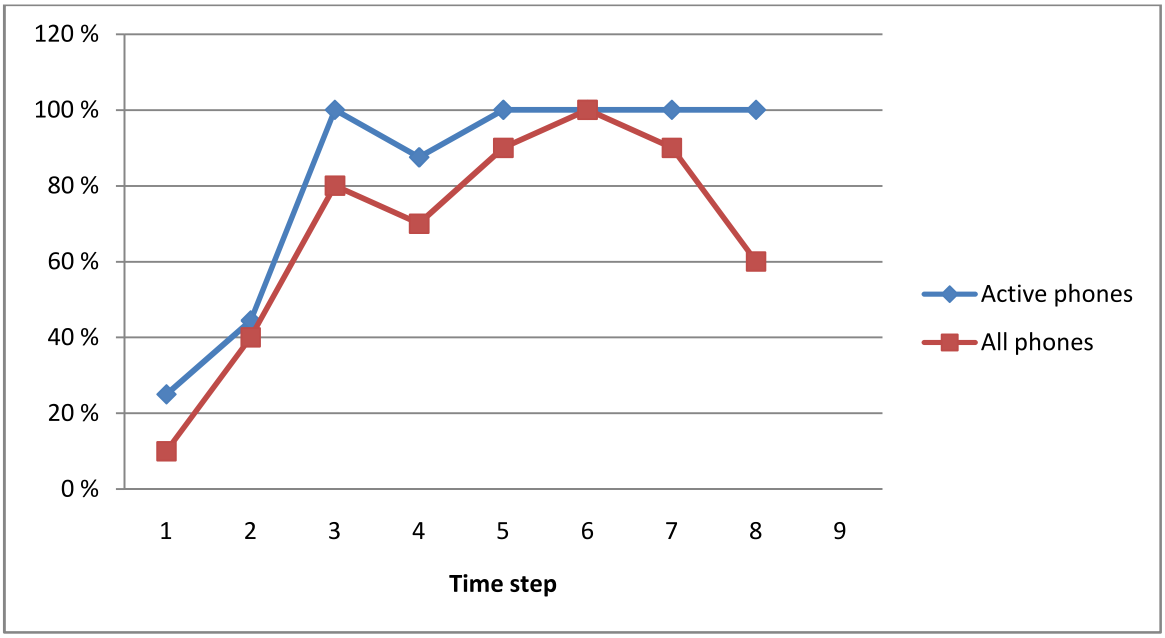

Figure 5: Capacity utilized at each time step, model with 8 devices, 18 file parts and 8 time steps.

In the 8-18-8 case that Figure 5 presents, there is a drop in utilized capacity at the fourth time step. This drop occurs because device 4 enters in the fourth time step, whereas no new devices entered in the third time step. However, the earlier the last device performs its first download the earlier its upload capacity can be utilized which speeds up the sharing process. We suspect that device 4 could have entered the network earlier, e.g., at time step 3, but that this didn't happen in our case because of the way our algorithm functions, as explained above.

3.1.7 Further Analysis through a Simple Theoretical Approach

In this section, we approach the ul<<dl cases of file sharing problems in homogenous networks with an initial seed and peers with no file parts to begin with through a simple theoretical approach. First we will calculate a minimal bound for the total energy consumption of a network. In a network with I devices and K file parts, I-1 devices must in total download (I-1)K file parts.

$$DL_{TOT} = (I-1) \cdot K \tag{14}$$

Similarly, when the upload bandwidth of a device is u the devices must in total upload $I \cdot u \cdot t$ file parts, where t is the average time each device must be awake.

$$UL_{TOT} = I \cdot K \cdot t \tag{15}$$

Because the number of uploaded file parts must equal the number of downloaded file parts, we can combine the two equations above to calculate the average time each device must be awake. We obtain the following equation:

$$t = \frac{(I-1) \cdot K}{I \cdot u} \tag{16}$$

The minimum bound for the required amount of active time steps can be calculated by multiplying the average awake time with the number of devices, and then adding I-1 time steps (because I-1 devices must download at one time step before uploading).

$$min\,bound = I \cdot \frac{(I-1) \cdot K}{I \cdot u} + (I-1) = (I-1)\left(\frac{K}{u} + 1\right) \tag{17}$$

This minimum bound for the energy consumption matches the actual minimal energy consumption very accurately. It provides exact estimates of the energy consumption in all cases, except for case 6 in which the network and number of file parts is very small. As the actual consumption is always an integer the minimal bound needs to be rounded up if it is not an integer, as in case 2 and 5.

Table 2: Total energy consumption and minimum bounds for torrent networks

	Devices	File parts	Time steps	Upload bandwidth	Download bandwidth	Total energy consumption	Minimum bound
Case 1	10	20	9	3	27	69	69.00
Case 2	11	20	10	3	30	77	76.67
Case 3	10	18	8	3	27	63	63.00
Case 4	7	3	6	1	6	24	24.00
Case 5	5	5	4	3	12	11	10.67
Case 6	3	3	6	1	2	9	8.00

We may observe that if we hold I and u constant, the minimum bound equation is linear except for the occasional need for rounding up. In this case, the slope of function (16) is (I/u+1), and the constant is -(I/u+1).

Let's consider case 3 where we have 10 devices, 18 file parts, and an upload bandwidth of 3. If we add one device to the network, equation (16) predicts that the energy consumption increases by (I/u+1) active time steps, i.e., 7 time steps. Similarly, if we add two devices to the network, equation (16) predicts the energy consumption to increase by 14. Now, let's imagine cases where one and two devices are added to the network at a time when all other devices have completed their downloading. When one device is added, the energy consumption is minimized when 6 devices upload at maximum speed (3 file parts each) simultaneously to the new device. The total awake time is thus 7, as predicted. Similarly, when two devices enter they each need to download 18 pieces. This cannot be completed in one time step. In this case the new devices do not even benefit from sharing file parts with each other – contrary to what could be expected it does not reduce the total amount of energy that is spent in the network. The case studied here requires, however, that we allow the download bandwidths to be large enough.

When the total number of file parts is evenly divisible with the upload bandwidth, devices upload at maximum speed always when active. When the total number of file parts is not evenly divisible with the upload bandwidth exceptions occur in which devices upload at a speed below maximum during at least one step.

3.1.8 Conclusions

After a file part has been sent into the network by the initial seed it becomes quickly held by a majority of the devices in the network. Bottlenecks of the network include the number of devices that hold the file initially, and the upload speeds of the devices, especially that of the initial seed.

The average upload and download speeds suggest that when a device is active, it uploads at maximum speed. The download speeds of the devices, on the other hand, vary a lot. However, the average download speed is very close to the average upload speed.

In the analysis of the number file parts in possession of each device, we noted that the initial seed uploads to three devices at a time from the start. This enables the rapid increase in the capacity utilized, and leads to an increasing growth rate of the total number of file parts in the network from the beginning.

Generally the devices that are active early on have the highest total activity times, while the devices which have their first downloads last have the lowest total activity times. However, there are exceptions to this.

The activity time shows that the initial seed keeps uploading also after it has sent the whole file into the network. It seems not to make any difference which devices provide the upload capacity, as long as all devices upload at maximum speed when active. As a result, the total number of time steps each device is active may differ quite a lot from one device to another.

For the devices to be able to always upload at maximum speed, they need to hold the file parts that are not in possession of all the other devices in the network. In order for this to be fulfilled some kind of "planning" may be needed in the sharing process. If the sharing occurs randomly it's easy to imagine devices end up in situations where they are not able to upload because the file parts they hold are already held by all other devices.

3.1.9 Derived Heuristics

The most central heuristics, that we derived from our results and analysis of the cases in which download bandwidth is substantially larger than upload bandwidth, are:

- When a device is active it will upload at maximum upload bandwidth
  - except for in its first active time step during which it only downloads.
  - except if the total number of file transfers is not evenly divisible with the upload speed.
- For a device to be able to always upload at maximum speed, some planning or rules of thumb of which parts each device downloads is likely to be required.

It seems that these two heuristics may be sufficient to guide the devices of a homogenous network to operate in an energy optimal way.

3.2 Networks with Equal Upload and Download Bandwidths

The second case we studied considered network of energy sensitive devices similar to the first case. This time, however, the upload and download bandwidths of devices were set to equal level (further, these cases are referred to as "ul=dl" cases). In the case discussed in section 3.1, the upload bandwidth was significantly smaller than the download bandwidth. More specifically, the download bandwidths of the devices were 10 times the upload bandwidths. In that case, the limited upload speed restricted the sharing efficiency far more than the download speed as it was effectively unlimited. When setting the download bandwidth to the same level as the upload bandwidth, it is expected, that the downloading capacities become more restricting thus changing the behaviour of the network. In the cases considered in this section, the upload and download bandwidths were set to three file parts per time step.

The assumption of equal upload and download bandwidths can be a realistic one. Usually, when considering the ordinary BitTorrent networks between regular desktop computers operating through ordinary Internet connections, the downloading bandwidths greatly exceeds the uploading bandwidths limiting file sharing. However, when considering the mobile BitTorrent network where all devices operate through a 3G network, the situation can be different. According to experiences of the project client, in 3G networks, the devices' abilities to upload and download are equal to each other. Also in some cases, where the devices are connected through WLAN network, the maximum upload and download speeds are approximately equal.

In total we formed two different cases with equal download and upload bandwidths. The first consisted of 8 devices, 15 file parts, and 10 time steps. The second comprised 9 devices, 17 file parts, and 12 time steps.

3.2.1 Number of Devices Possessing Each File Part

As in section 3.1.1, we investigated the development of the amount of devices having each file part at each time step. Figure 6 presents the number of devices having each file part at the beginning of each time step for the case having 8 devices, 15 file parts and 10 time steps. This illustrates a typical ul=dl case.

In section 3.1, the same features were investigated in the ul<<dl case. It was observed, that the copies of file parts were not distributed evenly. They rather followed the pattern of sharing a file part quickly to all devices before distributing other file parts. In other words, the file parts, which were uploaded first in the network, were usually possessed by more devices at each time step than the file parts which were uploaded later. This caused a situation where some of the file parts where in possession of all the devices while some of the parts were still only at the first device (see for example Figure 1 in section 3.1).

When limiting the download and upload bandwidths at the same level, the situation changed. As can be seen from Figure 6, the behaviour observed in the previous section does not apply anymore in this situation. The file parts are clearly more evenly distributed. In the ul<<dl case, the devices were more prone to spread the first file parts quickly to the whole network. In the current case, however, the devices rather extract most of the file parts from the initial device to some of the devices before starting to spread them to further devices. For example, as can be seen from Figure 6, at the first three time steps, only new file parts are shared to the network from the initial seed, but the already copied file parts are not yet shared forward. Not until the beginning of the fifth time step are the file parts shared forward from the initially downloading devices.

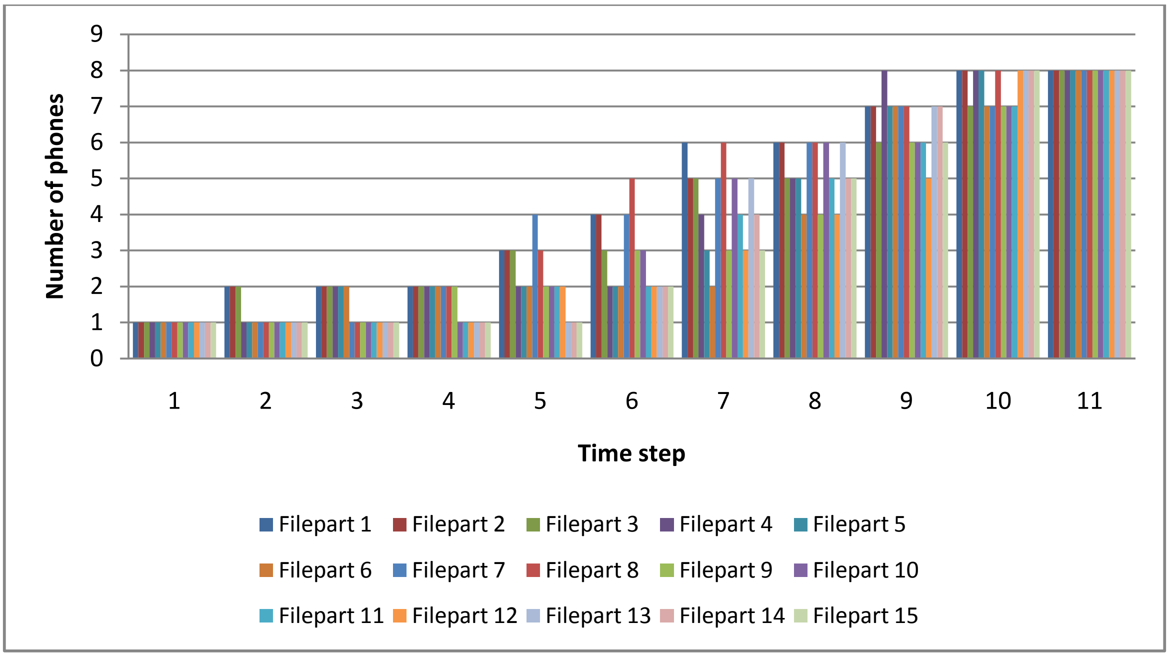

Figure 6: Number of devices possessing each file part at the beginning of each time step, case with 8 devices, 15 file parts and 10 time steps.

3.2.2 Number of File Parts in Possession of Each Device

As in section 3.2.1, we also investigated the number of file parts each device possesses at each time step. Figure 7 illustrates a typical example of how the number of file parts each device holds develops over time for ul=dl case.

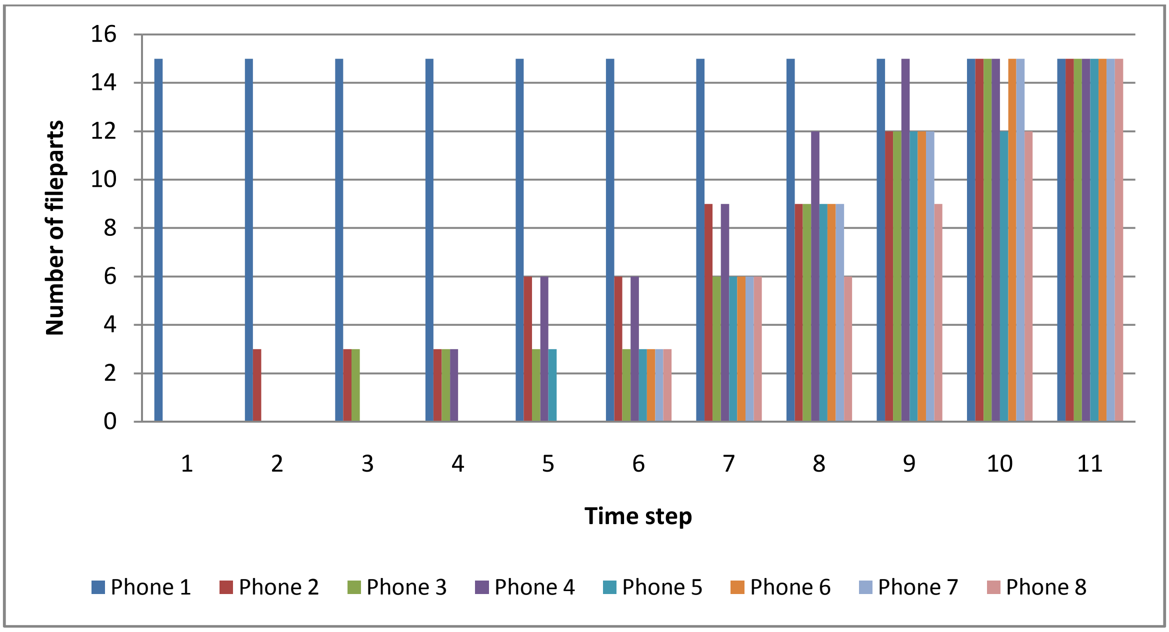

Figure 7: Number of file parts in possession of each device at the beginning of each time step, case with 8 devices, 15 file parts and 10 time steps.

In section 3.2.1, it was noted that during the first time steps file parts from the initial seed were extracted. Only after that were the downloaded file parts shared to further devices. The same behaviour can be observed from Figure 7. At the first time step, three file parts are transferred from the seed to another device. During the next two time steps three file parts are uploaded to another two devices. Based on the Figure 6, these file parts are not the ones transferred at the first time step. This indicates that the initial seed shares file parts that have not yet been uploaded at the earlier time steps. Therefore it seems that during the first time steps, the initial seed remains active sharing new file parts to the devices that have not yet received any file parts.

Another distinctive feature from the earlier ul<<dl case is that this time the sharing behaviour seems to favour the blocks of three file parts. Figure 7 shows that every time a device downloads, it receives three file parts. This indicates that in the case of equal download and upload bandwidths, it is energy efficient to download at the full speed whenever a device is downloading. However, this behaviour only occurred in some of the ul=dl cases. The behaviour tended to take place in models where the number of file transfers was divisible by the upload (and also download) bandwidth. If this was not the case, the block forming behaviour did not appear as strongly as in Figure 7, but it was still detectable.

The fairness of sharing, in terms of possessed file parts at each time step, seems not to differ significantly from the case of download speed limits exceeding upload speed limits. In neither case there was found to be any devices that are lagging in comparison to other devices in terms of downloaded file parts. There does not seem to be any devices that receive all file parts significantly earlier than the other devices.

3.2.3 Average Upload and Download Speeds

In Table 3, key parameters of the upload and download activity are presented, as in section 3.1.3. The numbers of total download and upload time steps are relatively close to each other in all of the cases. In the first case, 8-15-10, the numbers are the same since the amount of file parts is divisible by the upload (and thus download) bandwidth unlike in case 9-17-12 in which the number of downloading time steps is bigger than the number of upload time steps.

Table 3: Key parameters of the upload and download speeds.

	Case 8-15-10	Case 9-17-12
Number of upload time steps	35.00	46.00
Average upload speed	3.00	2.96
Standard deviation of upload speed	0.00	0.21
Number of download time steps	35.00	48.00
Average download speed	3.00	2.83
Standard deviation of download speed	0.00	0.38

3.2.4 Total Number of File Parts in the Network

In section 3.2.2, we investigated the amount of file parts each device possesses at each time step. In this section, the total amount of file parts existing in the network is studied.

Figure 8 presents the total number of file parts in the network at each time step for a typical ul=dl case. Initially, the amount of file parts in the network equals the amount of file parts the initial seed is possessing. At the time when all of the devices have received all file parts, the total amount is the number of devices times the number of file parts. In this case, the initial amount of file parts is 15, and at the end the amount is 15*8=120.

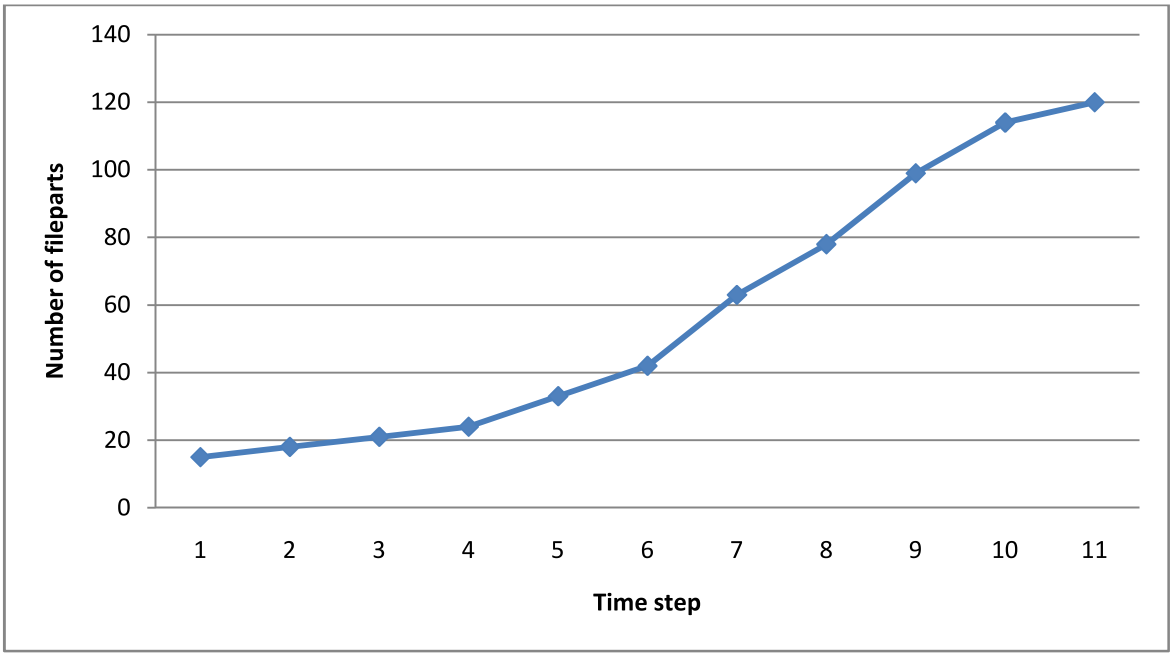

Figure 8: Total number of file parts in the network, model with 8 devices, 15 file parts and 10 time steps.

The same feature was also studied in section 3.1 for the case in which the download bandwidth was greater than the upload bandwidth. The curve indicating the amount of file parts in the network was found to be slightly S-shaped meaning that the increasing speed of the amount was greatest approximately at the midpoint of the total time.

In the current case, the same S-shape can be observed as well. However, during the first time steps the amount of file parts seems to grow linearly. This differs from the earlier cases, since in those the amount of file parts grew at an increasing rate during the first time steps. The linear growth could also have been anticipated based on Figure 6 and Figure 7 in the previous sections. It was noticed, that during the first time steps, the initial seed was the only uploading device and sharing file parts to single devices.

3.2.5 Activity Time

Figure 9 presents the cumulative activity times for each device at each time step for a typical ul=dl case. The figure shows that the initial seed remains active during the first five time steps. After that it remains idle for the rest of the time. The initial seed must upload all of the file parts in the network in order for all the other devices to be able to download them. Since the initial seed is active for only five time steps, there are 15 file parts, and the upload bandwidth is three file parts per time step, it has to upload different file parts at each time steps. Therefore the initial seed seems to be active only the necessary time to share all file parts to some other devices. After that the other devices share the downloaded files to each other. This behaviour seems logical, because the initial seed is not downloading at any time step. In order to minimize energy consumption, it is more energy-efficient if the active devices downloaded and uploaded at the full rate. The initial seed cannot download anything, and therefore it is active only the necessary time steps to upload the file parts once.

This situation differs from the ul<<dl case. In section 3.1.5, it was noted that the initial seed was active during almost all of the time steps. Therefore the initial seed kept uploading files to other devices longer than it would be necessary in order to upload the whole content to the network.

In section 3.1.5, it was noted that the devices seemed to form some sort of groups in terms of activity times. Also the devices tended to remain active after they had been active for the first time. These do not hold well in the current case. The group-forming effect seems to be smaller in the ul=dl case. Also fewer devices implement the same behaviour of remaining constantly active after the initial activity.

The total activity time seems to be distributed more evenly in the current case than in the ul<<dl case. In the latter, it was found that the activity time varied much between the most active and the most inactive devices in the network. However, in the current case of equal upload and download bandwidths, the total activity time of devices was very close to each other. This can be clearly seen in Figure 9 and occurred in all ul=dl cases.

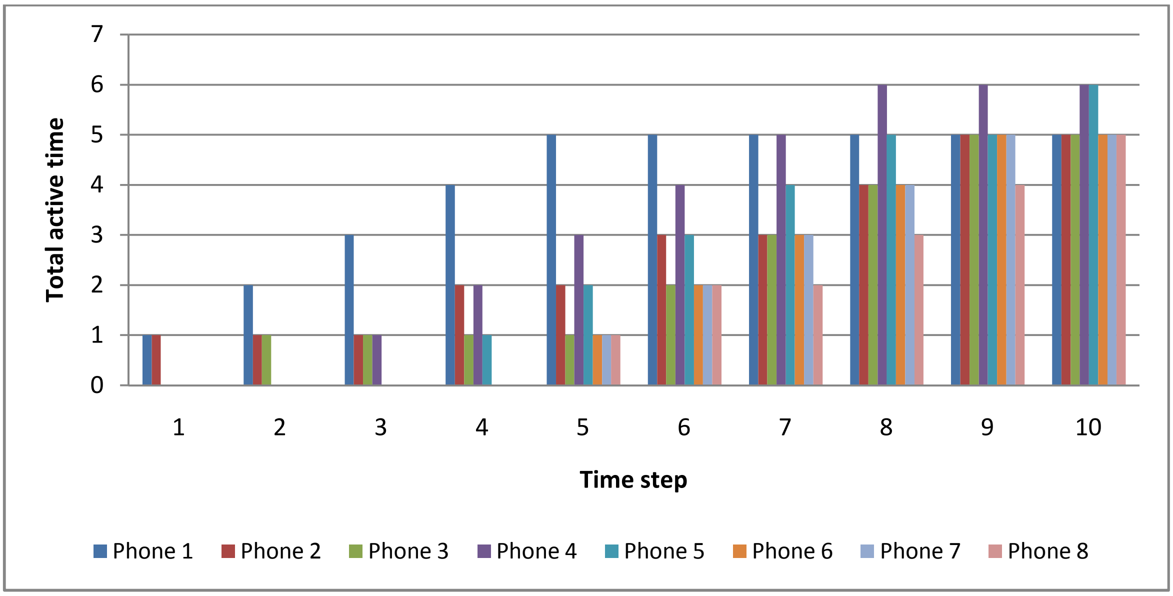

Figure 9: Cumulative activity time for each device, model with 8 devices, 15 file parts and 10 time steps.

3.2.6 Capacity utilized

The capacity utilized refers to in what extent the devices use the allowed upload bandwidth at each time step. Only upload bandwidth is considered, since the total download bandwidth always equals the total upload bandwidth. This can be used to measure the effectiveness of file sharing in terms of energy consumption. In this study two different capacity utilization measures were employed. In the first measure the percent of used bandwidth only considered active devices at each time step. The second measure also included inactive devices, thus providing information on how well the total file sharing potential was utilized at each time step.

Figure 10 presents the capacity utilized at each time step by two previously mentioned measures for a typical ul=dl case.

When considering only active devices, it seems that the full utilization percent is achieved at the sixth time step. At the first three time steps the utilization is only 50 percent. This is due to the behaviour examined in the previous section, in which the initial sharer uploads different file parts to three different devices during the first three time steps. After the maximum efficiency is reached, it is maintained for the rest of the time steps. This, however, is true only for cases where the number of file transfers is divisible by upload (and thus download) bandwidth. In other cases the utilization drops from the maximum before the last time steps.

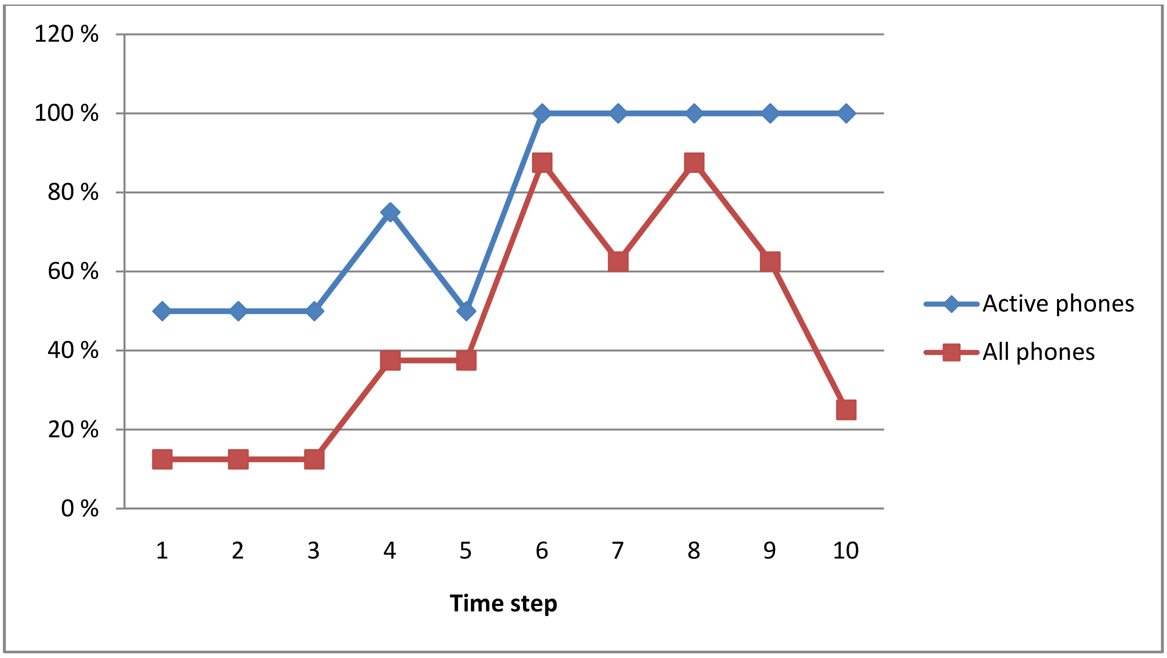

Figure 10: Capacity utilized at each time step, model with 8 devices, 15 file parts and 10 time steps.

If considering the capacity utilization of all devices in the ul=dl case, the percentage remains constantly lower than for the active devices only. This situation holds for all of the ul=dl cases and also mostly in the ul<<dl case. Only in the situations in which all of the devices in the network are active and uploading at full rate, the capacity reaches 100 percent. This is not achieved in either of our ul=dl cases at any point. This is mostly due to

initial seed's behaviour. It tends to share the file parts in the network and remain idle after that. This prevents the utilization of the full potential of the network.

3.2.7 Further Analysis through a Simple Theoretical Approach

This paragraph investigates a similar analytic minimum bound for the energy consumption as paragraph 3.1.7. As in the earlier case, a total uploading and downloading effort (16) must be met. The initial seed contributes Fu units to the upload effort, where F refers to the amount of time steps a single device needs to download or upload the whole file (I/u rounded up to the nearest integer). Peers contribute a total of (I-1)(F-1)u units during the F time steps they are active while downloading the file. If Fu+(I-1)(F-1)u is greater or equal to the total effort needed, (I-1)K, pF time steps are sufficient. Otherwise n additional time steps are needed, where n is the smallest amount of time steps that is required to complete the uploading effort. In all,

$$IF + n, n = \min_x\{x \in \mathbb{Z}_+ | Fu + (I-1)(F-1)u + xu \ \geq (p-1)K\} \qquad (18)$$

time steps are needed.

Table 4 presents the realized energy consumptions and calculated minimum bounds for ul=dl cases. The minimum bound is calculated using function (18). This minimum bound for the energy consumption matches the actual minimal energy consumption.

Table 4: Total energy consumption and minimum bounds for the two cases.

	Devices	File parts	Time steps	Upload bandwidth	Download bandwidth	Total energy consumption	Minimum bound
Case 1	8	15	10	3	3	42	42
Case 2	9	17	12	3	3	54	54

3.2.8 Conclusions

One of the biggest difference in the current case compared to the previous case of download bandwidths exceeding upload bandwidths, was that the initial seed remained active only for the first few time steps uploading all of the file parts to the network. This occurred in both of our ul=dl cases and it clearly differs from the ul<<dl case considered in section 3.1. In the latter, the initial seed kept uploading for almost all of the time steps. Also it did not upload all of the file parts during the first time steps, but some of the parts were not shared until clearly after the midpoint of total time steps.

Another interesting feature was that in the case of equal download and upload bandwidths, the sharing behaviour seemed to form blocks of several file parts. In ul=dl cases where the number of file transfers is divisible by upload bandwidth, every time when a device downloaded, it downloaded at the full rate of several file parts per time step. This was not the case in section 3.1, where download speeds varied a lot.

Total activity times of devices seemed to be more evenly distributed in the case of equal download and upload bandwidths. The differences in total activity times were only one time step at most, while in section 3.1 the total activity time of the most active device was twice the activity time the most inactive device in some cases.

Utilized capacity also differed between the two cases. When considering only active devices, the utilized capacity reached its maximum slower than in the case of section 3.1. However, in ul=dl cases where the amount of file parts was divisible by upload bandwidth, the maximum was sustained for the rest of the time steps after it was reached for the first time. If considering the capacity utilized for all devices, the maximum was not reached at all in the case of equal download and upload bandwidths. This is mostly due to the initial seed remaining idle after it has uploaded all of the file parts once. This did not hold in section 3.1, where the maximum was reached at least in one point for all of the cases.

3.2.9 Derived Heuristics

Based on previous conclusions, some new heuristics can be derived for the ul=dl case. According to our two ul=dl cases, the following heuristics apply when pursuing energy optimal sharing:

- When a device is active it will upload and download at maximum speed
    - except if the total number of file parts is not evenly divisible with the upload speed. This leads to at least $(p-1)$ and at most $(p-1)\cdot(r-u)$ exceptions.
- The initial sharer remains active for only the amount of time steps necessary to upload all of the file parts once to the network.

3.3 Comparison to a client-server model

In this paragraph, the energy consumption of the previously considered BitTorrent (peer-to-peer) model is compared to the energy consumption of a simple client-server model with no information exchange between the clients. This aims at determining the difference between the energies needed to distribute a file to a similar network via these two methods.

The client-server model refers to a situation in which the server initially holds the whole file and clients do not hold any parts of it. The file is then downloaded by the clients in parts similar to the ones in the BitTorrent formulation. The server's upload bandwidth is considered unlimited and the clients' download bandwidth is limited to a value equal to the case it is being compared to. The energy consumption of the client-server model with p-1 clients is compared to a BitTorrent model with p devices, one of which is the initial seed. This refers to a similar increase in the amount of devices possessing the whole file in the end. A decrease in the total energy consumption can be expected as the same downloading effort needs to be met with one energy sensitive device less.

Energy consumption of the client-server model is

$$E = (p - 1)F \tag{18}$$

where F is the number of active time steps needed to download the whole file (f/d rounded up to the nearest integer). Energy consumption of the BitTorrent model is assumed to be the minimum bound suggested by the theoretical approaches in paragraphs 3.1.7 and 3.2.7.

The results of the comparison are presented in Table 5 and Table 6. In the ul<<dl case, the energy consumption of the BitTorrent model experiences a dramatic increase compared to the client-server model. This is the result of devices' upload bandwidths becoming the bottleneck in terms of efficient use of download bandwidth. Ul=dl case shows that a network with devices capable of downloading and uploading at the same speed works almost as efficiently as a peer-to-peer network and a client-server network.

Table 5: Energy consumptions in the case of download bandwidth 10 times the upload bandwidth

Case	Devices	File parts	Active time steps for client-server model	Active time steps for BitTorrent model	Increase in energy consumption (%)
Case 1	50	50	98	866	784
Case 2	50	100	196	1683	759
Case 3	100	50	198	1749	783
Case 4	100	100	396	3399	758

Table 6: Energy consumptions in the case of equal download and upload bandwidths

Case	Devices	File parts	Active time steps for client-server model	Active time steps for BitTorrent model	Increase in energy consumption (%)
Case 1	50	50	833	866	3.96
Case 2	50	100	1666	1700	2.04
Case 3	100	50	1683	1749	3.92
Case 4	100	100	3366	3400	1.01

The assumption, that the minimum bound for energy consumption is always possible to achieve, may hold well for the ul=dl case but caution must be taken when it is being made for the ul<<dl case especially when the amount of file parts and upload bandwidth in the whole network exceed the download bandwidth of a single device. Cases that were investigated showed that analytic minimum bounds were achievable for small networks. However, further analysis is needed to determine whether download bandwidth may become a restricting factor in the case of ul<<dl and whether the minimum bounds can be achieved in the case of larger networks. Furthermore, a total energy perspective would require taking account of the energy consumption of the server.

4 Alternative Approaches

The model used for optimization of energy consumption experiences problems typical for ILPs (Margot 2007). Several symmetrical optimal solutions exist, because devices and file parts are identical and thus interchangeable. For example, as the upload bandwidth of the initial seed is smaller than the amount of feasible receivers at the first time step, it is possible to upload to any 1 to u_i of these and still achieve the same optimal value for the ILP. It is not known whether the Xpress-MP uses any symmetry-breaking methods so deploying a solving algorithm with such capabilities could decrease the optimization time significantly. This can be done in several ways.

First, symmetry-breaking constraints can be added in order to reduce the amount of feasible solutions to the ILP. Constraint (13) deals with the symmetrical solutions containing empty time steps allowing only one order of the active and non-active time steps. Similar constraints were formulated in order to reduce symmetries of the devices and file parts but none proved to be efficient in terms of optimization time. Such constraints do exist but they involve logic programming and cannot be formulated as linear constraints.

The problem with branch and bound algorithm is that it cannot terminate optimization until it has found all of the optima or one that is the smallest possible integer solution based on the solution of the LP relaxation. For example, for the case of 10 devices, 20 file parts and effectively unlimited download bandwidth, LP relaxation gives an optimal value of 60 active time steps and the ILP 69 time steps. This means that the branch and bound algorithm must have gone through all feasible solutions in order to be certain that 69 is the optimal value because it could have stopped only at 60 knowing that a better one cannot be found. Logic programming approach would suggest considering at each node whether assigning a decision variable to a certain integer value (in most cases 0 or 1) would result in a situation for which a symmetrical one has already been explored (Caprara, et al. 1998). In this way, the branches could be cut early on and the time needed to complete the optimization possibly reduced dramatically.

Second, the problem could be formulated as a constraint satisfaction problem (CSP). CSPs are a class of problems that has been studied a lot and for which several efficient algorithms have been documented. The CSP approach to energy consumption optimization would mean discarding the objective function and instead adding the amount of active time steps as an equality constraint. Optimization would be conducted as finding the minimum amount of time active time steps which still would make the CSP feasible. The strength of this approach is that it extracts a solution that can be analyzed while the algorithm is still searching for a better one. In some cases a solution can be found optimal by comparing it to an analytic minimum bound (17). As the case was with the method used in this research, no information other than the optimal value found so far could be extracted from the optimization software. During the optimization process, it seemed that the final optimal value was achieved rather early in the optimization process so the CSP approach would save a lot of time by allowing several parallel tasks and reducing the computational effort compared to the ILP (Smith, et al. 1996).

5 Conclusions

This research project examined the optimization of energy consumption in mobile BitTorrent networks.

First, we analyzed situations where the download bandwidth is significantly larger than the upload bandwidth. The most important observations were as follows: an active device uploads at maximum speed, maximum upload speed requires some planning and energy consumption increases linearly as the number of devices increase.

Second topic of analysis was the case with equal upload and download bandwidths. We made following observations: total activity time was more evenly distributed than in the first case, active devices reached maximum utilized capacity slower than in the first case but it was sustained to the end of sharing; sharing behavior seemed to form block of files, and initial seed remained active only the first time steps that were required to upload all of the file parts to network.

6 Bibliography

Caprara, A, et al. "Integrating constraint logic programming and operations research techniques for the Crew Rostering Problem." *Software: Practice and Experience* (John Wiley & Sons, Ltd.) 28, no. 1 (January 1998): 49-76.

Cohen, Bram. *BitTorrent protocol specification.* January 10, 2008. http://www.bittorrent.org/beps/bep_0003.html (accessed April 18, 2009).

Margot, François. "Symmetric ILP: Coloring and small integers." *Discrete Optimization* 4, no. 1 (March 2007): 40-62.

Nokia. *Nokia N95 - Specifications.* 2009. http://www.nokia.co.uk/find-products/all-phones/nokia-n95/specifications (accessed April 18, 2009).

Nurminen, J K, and J Nöyränen. "Energy-Consumption in Mobile Peer-to-Peer - Quantitative Results from File Sharing." *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE.* Las Vegas, Nevada, 2008. 729-733.

Smith, Barbara M, Sally C Brailsford, Peter M Hubbard, and H Paul Williams. "The progressive party problem: Integer linear programming and constraint programming compared." *Constraints* (Springer Netherlands) 1, no. 1-2 (September 1996): 119-138.

Appendix A: The Model in Brief

Parameters

d_i = download speed (file parts per time step) of the device *i*

u_i = upload speed (file parts per time step) of the device *i*

Variables

f(i,k,t) indicates whether the device *i* has file part *k* at time *t* (1) or not (0)

r(i,k,t) indicates whether the device e *i* downloads file part *k* at time *t* (1) or not (0)

s(i,k,t) indicates whether the device *i* uploads file part *k* at time *t* (≥1) or not (0), integer variable

z(i,t) ϵ {0,1} indicates whether the device *i* is active at time *t* (1) or not (0)

Objective Function and Constraints

$$\min \sum_i \sum_t (1000 + 0.1 * t - 1)) * z(i,t)$$

s.t. $\sum_i r(i,k,t) - \sum_i s(i,k,t) = 0 \ \forall k,t$

$f(i,k,t) - f(i,k,t-1) - r(i,k,t-1) = 0 \ \forall \ i,k,t$

$-u(i) * f(i,k,t) + s(i,k,t) \leq 0 \ \forall \ i,k,t$

$\sum_t r(i,k,t) + f(i,k,0) = 1 \ \forall i,k$

$\sum_k s(i,k,t) - z(i,t) * u(i) \leq 0, \text{when } i \geq 2 \text{ and } t \geq 2$

$\sum_k s(i,k,t) - z(i,t) * u(i) \leq 0, \text{when } i = 1, \forall \ t$

$\sum_k r(i,k,t) - z(i,t) * d(i) \leq 0 \ \forall t \ , i \geq 2$

Appendix B: The Code

```
function [fopt,ropt,sopt,zopt,opt,status,info,dl_speed,ul_speed]=...
    bittorrent7(num_devices,num_parts,timesteps,dl_speed,ul_speed)
%This function searches the energy optimum when sharing file parts to
all
%devices.
%
%[fopt,ropt,sopt,zopt,fend,opt,status,selite]=bittorrent7(num_devices,..
.
%    num_parts,timesteps,dl_speed,ul_speed)
%
%num_devices = the number of devices
%num_parts = the number of file parts
%timesteps = the number of time steps
%dl_speed = how many file parts each device may download during one time
%            step (a vector)
%up_speed = how many file parts each device may upload during one time
step
%            (a vector)
%
%Information about one of the optimal solutions:
%fopt = which file parts each device has (1) and has not (0) at each
time step
%            (a 3-dimensional matrix: device|file part|time step)
%ropt = which file parts each device downloads (1) or not (0) at each
time step
%            (a 3-dimensional matrix: device|file part|time step)
%sopt = which file parts each device uploads (>0) or not (0) at each
time step
%            (a 3-dimensional matrix: device|file part|time step)
%zopt = which devices are active (1) and which are not (0) at which time
steps
%            (a 2-dimensional matrix: device|time step)
%opt = the optimum value of objective function
%status = status of the solution
%info = information about parameters etc

%If the first device has all the file parts in the beginning and other
devices
%have no file parts, the uploading possibility (variable s) has been
removed
%from other devices, besides the first, at the first time step.

%If the first device has all the file parts in the beginning, it will
never
%download anything (r have been removed)


%defining which devices have which file parts at the start (first device
has
%all the file parts)
f0=zeros(num_devices,num_parts);
f0(1,:)=1;

%if the first device has all the file parts and no other device has no
```

```
%file parts, we may form constants p and q, with which we can later
remove
%constraints and make x shorter
if sum(f0(1,:))==length(f0(1,:)) && sum(sum(f0(2:end,:)))==0
    p=1;
    q=1;
%if the first device has all the file parts we can only use constant p
elseif sum(f0(1,:))==length(f0(1,:))
    p=1;
    q=0;
else
    p=0;
    q=0;
end

%The problem is a linear optimization problem Ax ? B, where ? can be
either
%=, <= or >=.

%variable lengths in vector x
f=num_devices*num_parts*timesteps;
r=f-p*timesteps*num_parts;
s=f-q*(num_devices-1)*num_parts;
z=num_devices*timesteps;
x=f+r+s+z;

%Constraints

%f(t)=f(t-1)+r(t-1) if the device uploads a file part, it will have the
part
%all the time
%has is one part of the constraint matrix A
has=zeros(f,x);
%b_has is the part of B corresponding has in A
b_has=zeros(f,1);
b_has2=zeros(s,1);
%the type of the constraint ('E':Ax=B 'L':Ax<=B 'G':Ax>=B)
c_has(1:1:f,1)='E';
%s(t)<=f(t) device can only upload the file part if it already has it
has2=zeros(s,x);
c_has2(1:s,1)='L';
for i=1:num_devices
    for j=1:num_parts;
        for k=1:timesteps

            %setting the multipliers for f(t)
            has(1+(i-1)*timesteps*num_parts+(j-1)*timesteps+k-1,...
                1+(i-1)*timesteps*num_parts+(j-1)*timesteps+k-1)=1;

            %multipliers for s(t)
            if i==1
                has2(1+(i-1)*num_parts*timesteps+(j-1)*timesteps+k-1,...
                    f+r+1+(i-1)*num_parts*timesteps+timesteps*(j-1)+k-
1)=1;
            else
                %at each time step the device can only upload the file
parts
                %it already has
```

```
                has2(1+(i-1)*num_parts*timesteps+(j-1)*timesteps+k-1-...
                    q*(i-1)*num_parts,f+r+1+(i-
1)*num_parts*timesteps+...
                    timesteps*(j-1)+k-1-q*(i-1)*num_parts)=1;
            end

            %setting -ul_speed to be the multiplier of f(t)
            if i==1
                has2(1+(i-1)*timesteps*num_parts+(j-1)*timesteps+k-1-...
                    q*(i-1)*num_parts,...
                    1+(i-1)*timesteps*num_parts+(j-1)*timesteps+(k-1))=-
ul_speed(i);
            elseif k~=1
                has2(1+(i-1)*timesteps*num_parts+(j-1)*timesteps+k-1-...
                    q*((i-1)*num_parts-num_parts+j),...
                    1+(i-1)*timesteps*num_parts+(j-1)*timesteps+k-1)=-
ul_speed(i);
            end
            if k==1
                %the starting situation is set to be in the right-hand
side
                %in the constraints (B)
                b_has(1+(i-1)*num_parts*timesteps+(j-
1)*timesteps)=f0(i,j);
                if i==1
                    b_has2(1+(i-1)*num_parts*timesteps+(j-
1)*timesteps)=...
                        ul_speed(i)*f0(i,j);
                end
            else
                %f(t-1) has multiplier -1
                has(1+(i-1)*timesteps*num_parts+(j-1)*timesteps+k-1,...
                    1+(i-1)*timesteps*num_parts+(j-1)*timesteps+k-2)=-1;
                %r(t-1) has multiplier -1
                if i==1 && p==1
                else
                    has(1+(i-1)*timesteps*num_parts+(j-1)*timesteps+k-
1,...
                        f+1+(i-1-p)*timesteps*num_parts+(j-
1)*timesteps+k-2)=-1;
                end
            end
        end
    end
end

%forming the constraint matrix
A=[has;has2];
B=[b_has;b_has2];
ctype=[c_has;c_has2];
clear has has2 c_has c_has2 b_has b_has2

% device can only upload if it is awake and upload speed is the maximum
ul_ub=zeros(num_devices*timesteps-q*(num_devices-1),x);
b_ul=zeros(num_devices*timesteps-q*(num_devices-1),1);
c_ul(1:1:num_devices*timesteps-q*(num_devices-1),1)='L';
% device can only download if it is awake and download speed is the
maximum
```

```
dl_ub=zeros((num_devices-p)*timesteps,x);

for i=1:num_devices
    for j=1:timesteps

        if i==1
            %multiplier of s
            ul_ub(1+(i-1)*timesteps+(j-1),...
                f+r+1+(i-1)*timesteps*num_parts+(j-1):timesteps:...
                f+r+1+i*timesteps*num_parts+(j-1)-1)=1;
            %multiplier of z
            ul_ub(1+(i-1)*timesteps+(j-1)-q*(i-1),...
                f+r+s+1+(i-1)*timesteps+j-1)=-ul_speed(i);
        elseif j~=1
            %multiplier of s
            ul_ub(1+(i-1)*timesteps+(j-1)-q*(i-1),...
                f+r+1+(i-1)*timesteps*num_parts+(j-1)-q*((i-1)+(i-
2)*num_parts-...
                max(0,i-2)):timesteps-1:f+r+1+i*timesteps*num_parts+(j-
1)-1-...
                q*((i-1)+(i-1)*num_parts-max(0,i-2)))=1;
            %multiplier of z
            ul_ub(1+(i-1)*timesteps+(j-1)-q*(i-1),...
                f+r+s+1+(i-1)*timesteps+j-1)=-ul_speed(i);
        end

        if i==1 && p==1
        else
            %multiplier of r
            dl_ub(1+(i-1-p)*timesteps+(j-1),...
                f+1+(i-1-p)*timesteps*num_parts+(j-1):timesteps:...
                f+1+(i-p)*timesteps*num_parts+(j-1)-1)=1;
            %multiplier of z
            dl_ub(1+(i-1-p)*timesteps+(j-1),...
                f+r+s+1+(i-1)*timesteps+j-1)=-dl_speed(i);
        end
    end
end

b_dl=zeros((num_devices-p)*timesteps,1);
c_dl(1:1:(num_devices-p)*timesteps,1)='L';

A=[A;ul_ub;dl_ub];
B=[B;b_ul;b_dl];
ctype=[ctype;c_ul;c_dl];
clear ul_ub dl_ub b_ul b_dl c_ul c_dl

%every file part is downloaded exactly once if the device doesn´t have
it
%already in the beginning
r_once=zeros((num_devices-p)*num_parts,x);
b_once=zeros((num_devices-p)*num_parts,1);
c_once(1:1:(num_devices-p)*num_parts,1)='E';
for i=1:num_devices
    if i==1 && p==1
    else
        for j=1:num_parts
```

```
            r_once(1+(i-1-p)*num_parts+(j-1),f+1+(i-1-
p)*timesteps*num_parts+...
                (j-1)*timesteps:1:f+1+(i-1-
p)*timesteps*num_parts+j*timesteps-1)=1;
            if f0(i,j)==0
                b_once(1+(i-1-p)*num_parts+(j-1))=1;
            end
        end
    end
end

A=[A;r_once];
B=[B;b_once];
ctype=[ctype;c_once];
clear r_once c_once b_once

%the file part has to be uploaded as much as downloaded at some time
step
%(there can´t be any 'free' file parts)
same=zeros(num_parts*timesteps,x);
b_same=zeros(num_parts*timesteps,1);
c_same(1:num_parts*timesteps,1)='E';
for i=1:num_parts
    for j=1:timesteps
        %r(t)=1
        same(1+(i-1)*timesteps+(j-1),f+1+(i-1)*timesteps+(j-1):...
           timesteps*num_parts:f+r)=1;
        %s(t)=-1
        if q==1
            for k=1:num_devices
                if k==1
                    same(1+(i-1)*timesteps+(j-1),f+r+1+(i-
1)*timesteps+(j-1))=-1;
                elseif j~=1
                    same(1+(i-1)*timesteps+(j-1),f+r+1+(i-
1)*timesteps+(j-1)+...
                        (k-2)*(timesteps-
1)*num_parts+timesteps*num_parts-i)=-1;
                end
            end
        else
            same(1+(i-1)*timesteps+(j-1),f+r+1+(i-1)*timesteps+(j-1):...
                timesteps*num_parts:f+r+s)=-1;
        end
    end
end

A=[A;same];
B=[B;b_same];
ctype=[ctype;c_same];
clear same c_same b_same

%creating the parameters for the optimization
H=[];
%variable lower bounds
lb=zeros(x,1);
%variable upper bounds
ub=[];
```

```
%variable types (I=integer, B=binary, C=constant)
vartype(1:f,1)='I';
vartype(f+1:x-z)='I';
vartype(x-z+1:x)='B';
%do want to see all the optimizing steps (1) in the output or not (0)
params.msglev=1;
%minimizing (1) or maximizing (0) the solution
sense=1;

%objective function
C=zeros(x,1);
for i=1:num_devices
    for j=1:timesteps
        %pushing the empty timesteps to the end
        C(f+r+s+1+(i-1)*timesteps+(j-1),1)=1000+0.1*(j-1);
    end
end

disp('--- The problem has been sent to Xpress. ---');

%sending the problem to Xpress
[xopt,opt,status,extra]=mexpress(sense,H,C,A,B,ctype,lb,ub,vartype,param
s);
clear A B ctype C lb ub

fopt=zeros(num_devices,num_parts,timesteps);
ropt=fopt;
sopt=fopt;
zopt=zeros(num_devices,timesteps);

%gathering the optimal solution xopt to better readable matrixes
for i=1:num_devices
    for k=1:timesteps
        for j=1:num_parts
            fopt(i,j,k)=xopt(1+(i-1)*num_parts*timesteps+(j-
1)*timesteps+(k-1));
            if i==1 && p==1
                ropt(i,j,k)=0;
            else
                ropt(i,j,k)=xopt(1+f+(i-1-p)*num_parts*timesteps+(j-
1)*timesteps+(k-1));
            end
            if i==1 || q==0
                sopt(i,j,k)=xopt(1+f+r+(i-1)*num_parts*timesteps+(j-
1)*timesteps+(k-1));
            elseif q==1 && k~=1
                sopt(i,j,k)=xopt(1+f+r+(i-2)*num_parts*(timesteps-1)+...
                    num_parts*timesteps+(j-1)*timesteps+(k-1)-j);
            end
        end
        zopt(i,k)=xopt(1+f+r+s+(i-1)*timesteps+(k-1));
    end
end

info=['Time    taken:    '    num2str(extra.time)    '    sec    =    '
num2str(extra.time/3600)...
```

```
    ' hours. The problem had ' num2str(num_devices) ' devices, '
num2str(num_parts)...
    ' fileparts, ' num2str(timesteps) ' timesteps. Used model was
bittorrent7.'];

%rounding the results
zopt=round(zopt);
fopt=round(fopt);
sopt=round(sopt);
ropt=round(ropt);

%autosaving the results
nimi=['workspace' num2str(num_devices) '_' num2str(num_parts) '_'...
    num2str(timesteps) '.mat'];
save (nimi)
```