

Bursty Content Sharing Mechanism for Energy-Limited Mobile Devices

Imre Kelényi
Budapest University of Technology and Economics
Budapest, Hungary
imre.kelenyi@aut.bme.hu

Jukka K. Nurminen
Nokia Research Center
Helsinki, Finland
jukka.k.nurminen@nokia.com

ABSTRACT

The strategy to send data in high speed bursts has been observed to be useful for energy-efficient communication of wireless applications. In this paper we investigate how to apply this approach to peer-to-peer (P2P) content sharing. Our main focus is on battery savings that can be achieved by properly shaping the traffic at the application level. Our primary target is BitTorrent based content sharing on mobile phones connected with 3G cellular networks. However, the results also apply to WLAN communication as well as to other energy-limited devices in addition to mobile phones. We propose a novel BitTorrent extension and evaluate it via flow-level simulations. In comparison to standard BitTorrent, the energy-limited peers can achieve 50% energy savings without significantly affecting the download speeds of regular peers.

Categories and Subject Descriptors

C.2.1 [Computer-Communication Networks]: Network Architecture and Design – *wireless communication*

C.2.2 [Computer-Communication Networks]: Network Protocols – *application, protocol verification*

General Terms

Algorithms, Measurement, Performance, Design

1. INTRODUCTION

The gap between the performance of mobile devices and desktop computers is becoming narrower. Improved processing and networking capabilities of mobile phones have opened up the possibility of their use in peer-to-peer networks. Our previous research on P2P applications for mobile phones, such as Symella [1] and SymTorrent [2], shows that today's mobile phones can effectively join and contribute to distributed content sharing networks.

However, the major issue that prevents mobile peer-to-peer content sharing from becoming ubiquitous is battery consumption. The energy measurement results with the SymTorrent BitTorrent client on Symbian based mobile phones show that SymTorrent

uses about the same amount of power as a voice call [3], meaning that a modern mobile phone can be used for two to five hours for peer-to-peer content sharing before its battery is completely depleted. This is clearly far from the optimal operation time.

The energy consumed by a mobile phone is the result of several factors. Communication via wireless radio, such as transferring data over the cellular network or wireless LAN, is power hungry and clearly dominates the energy consumed by data processing [4]. Therefore, we would need new solutions that, instead of speed, optimize the energy consumption of the wireless communication in P2P content sharing scenarios.

The essence of this research is the efficient use of the wireless interface. As observed in [5], in the context of mobile web access, data should be transferred at the maximum throughput when the wireless link is active. Then the wireless interface should remain off until the next download. We would thus like to reach an alternating sequence of states, where the wireless link either communicates with its maximum speed or is idle in the sleep state. The mechanisms that can be used to reach this target behavior in a distributed P2P system are the key objectives of our research.

We use BitTorrent as the base of our research since it is widely used and already supported by a wide range of platforms. Although it has been the target of extensive research, there is very little prior work on mobile BitTorrent and its energy consumption. Our goal is to propose a new protocol extension, BurstTorrent, which allows content transfer for mobile peers in a way that consumes less energy than the standard BitTorrent protocol. Furthermore, our target is that the new protocol would require only minor modifications to the existing clients.

The rest of the paper is structured as follows. In Section 2 we discuss the energy consumption in wireless P2P networks in detail. Section 3 presents how the battery saving ideas can be applied to BitTorrent based content transfer. Section 4 presents the proposed BurstTorrent protocol in detail. In Section 5 we discuss the flow-level simulations we have used to analyze the behavior of BurstTorrent, and compare it with the standard BitTorrent. In Section 6 we discuss our findings and in Section 7 reflect how our work is related to prior research. Finally, in Section 8 we summarize and conclude our work with future research directions.

2. ENERGY CONSUMPTION IN WIRELESS P2P NETWORKS

In order to understand the energy consumption of the wireless communication we performed a series of measurements with

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
PM2HW2N'09, October 26, 2009, Tenerife, Canary Islands, Spain.
Copyright 2009 ACM 978-1-60558-621-2/09/10...\$10.00.

Nokia N95 mobile phones and Nokia Energy Profiler. Although we used only one phone model, other measurements carried out with different mobile devices also resulted in similar energy consumption characteristics [4].

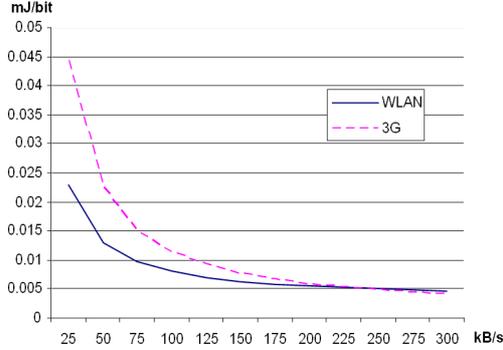


Figure 1. Power consumption per bit as a function of communication speed

Figure 1 shows the energy consumption per bit as a function of communication speed. The shape of the curves clearly shows that the higher the bitrate the more energy-efficient the communication is. This suggests that in order to save battery we should try to arrange the content download activity in a way that the mobile device is able to experience as high bitrates as possible.

Our results, which naturally depend on the used device models and the test environment, show that as bit rate grows by 100 kB, the power consumption only increases by 0.3W in WLAN and by 0.04W in 3G cellular. An important exception to this linear dependency is the case when there is no communication which allows the wireless interface to enter an idle power-saving state.

In the following we model the energy consumption of a wireless link with a binary approach using a constant value for the active state and another constant for the idle state. This model is not fully accurate but, since the growth coefficient is so small, the simplification causes only a small error, especially in the 3G cellular case which will be our focus in the rest of the paper.

$$E = t_{idle} \cdot P_{idle} + t_{active} \cdot P_{active} \quad (1)$$

Equation 1 shows our energy model where E is the overall energy consumption of a peer, P_{idle} and P_{active} are the idle and active state power consumptions, and t_{idle} and t_{active} are the times the peer spends in idle and active states. In 3G cellular the average active state power P_{active} was 1.3W and the idle state power P_{idle} was 0.07W.

Equation 1 and the major power consumption difference between the active and idle states can be used to derive goals to minimize the energy consumption.

- The device should spend as little time as possible in active state.
- When in active state it should use the maximum bit rate for communication.

Clearly spending time in idle state is desirable for the energy consumption. However, in idle state there is no activity and no data transfer so in order to do useful work the device should be in active state. For energy-efficient behavior, the device should thus alternate between the two states. In active state, it should perform a communication burst with very high data rate, and between the bursts, it should remain in idle state.

The energy saving strategy presented in this paper have a measurable effect only if the peers cannot utilize their full bandwidth using standard protocols. If the peers can download at the maximum speed, there is no space for improvements. SymTorrent measurements in [3] report a download speed of around 50 kB/s, which is far below the 250 kB/s average maximum bit rate observed in live 3G networks [6]. As a result, there is potential for conserving energy.

3. ENERGY EFFICIENT BITTORRENT

The previous section highlighted the fact that significant energy conservation can be achieved by performing data transfer in a way that it occurs as fast as possible, which results in the peer spending the shortest amount of time being active. Although this paradigm can be harnessed in several ways, the research presented here focuses on how it can be used to make BitTorrent more energy efficient.

Standard BitTorrent transfers content via a request-response mechanism: peers are constantly requesting pieces of the data from each other, and serving the requests whenever they have free upload slots. The peers that are currently served are selected by the use of a sophisticated tit-for-tat mechanism, which favors peers actively contributing to the swarm. In terms of energy efficiency, the main problem is that if the available upload bandwidth of the sender is lower than the required download capacity of the receiver, it cannot download at full speed. This is especially true for recently connected peers, which will receive even less bandwidth due to the tit-for-tat mechanism. Mobile devices, which often cannot accept incoming connections and generally have slower network connections, are even more likely to be stuck at slow download speeds.

Our system model includes two types of peers: energy-limited and regular. Energy limited peers are devices with limited battery capacity. Their goal is to receive the content in the most energy efficient way, even at the cost of a longer download time. This does not mean that download time is insignificant for energy-limited peers, but they prioritize energy-efficiency over download speed. In contrast with this, the term regular is used for peers that are not concerned about battery consumption. Their primary goal is to receive the content in the shortest amount of time. It should be noted that in addition to desktop computers, mobile devices plugged into the electrical outlet belong to this category. Based on this categorization today's BitTorrent networks show that the majority of the peers are regular, providing most of the available bandwidth.

The requirements for the energy-efficient BitTorrent-based protocol can be summarized as follows:

- The energy consumption of energy limited peers must be less compared with using standard BitTorrent. The download time for limited peers can be longer.

- Regular peers must not be penalized: they must receive the content in about the same amount of time as with using standard BitTorrent.
- Preserve compatibility with standard BitTorrent as much as possible. Peers not supporting the protocol and regular peers still use standard BitTorrent to transfer data between each other.

To satisfy these requirements, the proposed protocol uses the concept of scheduled transfers. An energy-limited peer, which is downloading content from a BitTorrent swarm, negotiates time intervals with regular peers when the regular peers would promise to use all the necessary resources to send content to the downloading peer with the agreed speed. This way it can be ensured that when an energy-limited device is in active state, it receives data at full speed.

Regular peers maintain an upload schedule in which they store points of time when data is needed to be sent to limited peers. Similarly to this, energy limited peers use a download schedule to calculate times when they can request new pieces of data. Scheduled transfers require regular peers to control and regulate their upload speed. They need to be able to reserve a portion of their bandwidth at certain times for the scheduled uploads. Nevertheless, this can be implemented relatively easily with a flow control mechanism, holding back packets to regular peers if the upload speed to energy limited peers is lower than required. If a regular peer cannot transfer at the agreed speed, the schedule can become corrupted, which can result in loss of energy, but it has only a temporary effect and does not ruin the whole transfer process.

It should be highlighted that we only investigate the case when energy-limited peers do not serve each other, and receive data only from regular peers. Energy-limited peers are usually behind firewalls and use NAT, which makes it difficult to accept connections and exchange data between each other.

4. BURSTTORRENT

BurstTorrent is an extended version of BitTorrent that allows using scheduled transfers to achieve energy efficient operation for energy-limited peers. Regular peers use standard BitTorrent to serve each other, and energy-limited peers are served via scheduled transfers, as defined by the new protocol.

Although regular peers could schedule multiple transfers, the protocol allows only one for each regular peer to minimize the size of the schedules both on the downloaders' and on the uploaders' side. This means that the regular peers know only the next scheduled upload they are going to perform, and energy limited peers need to wait until this upload is done to issue new timed requests to the regular peer. This way the excessive growth of the schedules can be prevented. A small schedule, in our case containing only one entry, is much easier to maintain. Also, if there were more transfers scheduled, the risk of the regular peer leaving the network before serving the request would also increase.

Regular peers serve other regular peers and energy limited peers alternating in time. Each scheduled transfer period is followed by a period in which the regular peer serves only other regular peers. The length of the latter period is determined based on the length

of the period serving the energy limited peer. This guarantees that the regular peers always "get back" the bandwidth that was taken away from them during the scheduled transfer time. An example can be observed in Figure 2, depicting the initial period of the lifetime of a regular peer serving BurstTorrent requests. The grey bars mark the time periods reserved for energy-limited peers, and regular peers are served between these.

Each regular peer maintains the following two values:

T_s (earliest serving time) the earliest time when the next scheduled transfer can take place. The value of T_s is updated at the time the next scheduled transfer is negotiated. The next value of T_s is calculated based on the bandwidth used by the scheduled transfer and the previous value of T_s (see the algorithm later). When the peer joins the network T_s is set to the actual time (so that it can start serving a timed request immediately).

T_r (earliest request time) is the earliest time when a regular peer can accept a new timed request. T_r is always set to the ending time of the actual scheduled upload of the peer. Thus, energy limited peers can start issuing new timed requests to the regular peer at T_r .

Therefore, a regular peer is always in one of the following two states:

- FREE: no scheduled upload is negotiated, waiting for a timed request.
- OCCUPIED: a scheduled upload has been negotiated and it has not been finished yet.

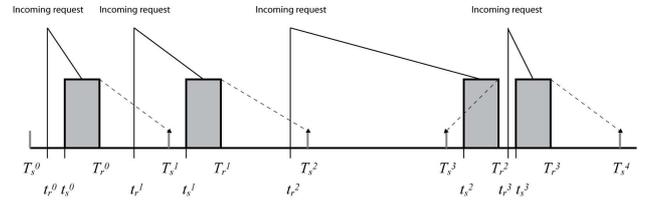


Figure 2. Regular peer serving BurstTorrent requests

It can be seen in Figure 2 that when an incoming request arrives from an energy limited peer, a new scheduled transfer is created (marked with a grey box). The actual scheduled transfer is always performed later than the earliest serving time ($t_s \geq T_s$) to ensure that the time spent serving regular peers and energy limited is kept in balance. The next value of T_s is set based on the last scheduled transfer and the previous value of T_s . In the figure, the scheduled transfer and the T_s value that is set based on it are connected with a dashed arrow.

BurstTorrent applies a two-phase negotiating process to schedule new transfers. The energy limited peers must always send a *REQUEST EARLIEST SERVING TIME* message first, which literally requests the earliest possible serving time (T_s) from the regular peer. If the peer is in FREE state, it sends back an *EARLIEST SERVING TIME* message. However, if it is in OCCUPIED state, it sends a *REFUSE EARLIEST TIME* message, which contains the next time when the energy limited peer can try to request a new serving time. The returned time must be later than T_r , but not necessarily equal to it. This is very important for determining which energy limited peer is served in the next period. Since the first request received in FREE state is served,

the regular peer can assign priorities to energy limited peers by sending them different earliest request time values. For example if one peer receives T_r and another T_r+10 , the former can send a new request earlier than the latter. It does not necessarily mean that the peer with higher priority will always make a new request since it is possible that by that time it has already received the missing pieces from a different source or created another request to a different peer, but the mechanism guarantees that if it still needs the pieces from this particular regular peer, it will be preferred to the other energy limited peer receiving the later request time. To avoid cheating, regular peers maintain the list of earliest request times sent out to energy limited peers, thus upon receiving a new request, it can be verified that the requester is not trying to obtain the time earlier than specified by the regular peer.

If the earliest *EARLIEST SERVING TIME* is received by an energy limited peer, it determines the actual serving time (t_s), which must be a later time than the earliest serving time (T_s), and sends back the time with the other details of the request (the list of subpieces, etc.) to the regular peer in a *TIMED REQUEST* message, which eventually results in the creation of a new scheduled transfer and the regular peer entering *OCCUPIED* state.

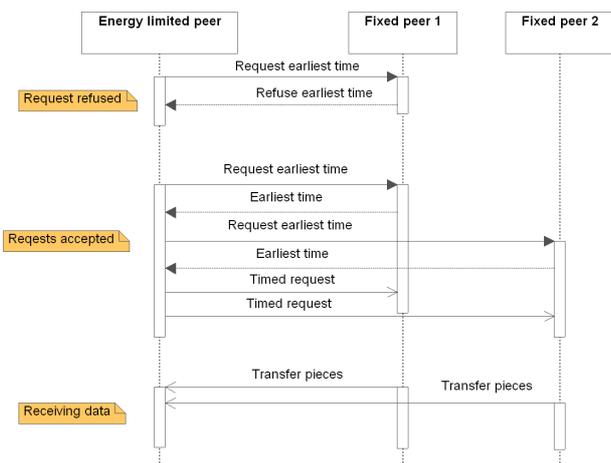


Figure 3. BurstTorrent schedule negotiation

The process of negotiating two scheduled transfers with two regular peers is illustrated in Figure 3. It can be seen that the first request to regular peer 1 is refused, but the second request is accepted. Upon receiving the earliest request time, the energy limited peer sends a request to peer 2, which is also accepted. By doing so, the peer can receive data from the two peers simultaneously at a later time. This is required because in this case, receiving data from one regular peer cannot fully fill the download capacity of the limited peer. To achieve the most energy efficient operation, the full download bandwidth must be utilized, which can be only done by receiving data from multiple peers simultaneously. To calculate the number of simultaneous transfers needed, the energy-limited peer need to be aware of the upload capacity of regular peers it is connected to. The means of doing this is out of the scope of this paper, but one possible solution is initially using a default value, then adapting it as data is received from the regular peers and the transfer speed can be measured.

It should also be noted that sending out the requests to regular peers happens sequentially in a very short amount of time. The

regular peers lock their next serving period for the requester until a timed request message is received or a certain timeout has been passed. Furthermore, although the two transfers depicted as if they were performed with some delay in the sequence diagram, in reality they should start at the same time.

In the currently implemented version of the BurstTorrent protocol, the priorities assigned to the energy limited peers are based on the last time they were served, giving lower priority to those peers that were served recently. The actual request times, which correspond to the priorities, are scattered uniformly in the period between t_r and t_s . The larger exclusive periods are available for energy limited peers to make the next request, the less important is the correct timing, making the system less error-prone.

4.1 Peer and Piece Selection

In a network, where the upload bandwidth of the regular peers is lower than the download bandwidth of the energy limited peers, multiple transfers need to be scheduled for the same time to achieve optimal bandwidth utilization. Thus, energy limited peers can only send out timed requests if enough regular peers are available as download sources. Furthermore, although in standard BitTorrent it is enough if a peer has at least one piece that the downloader is interested in, in BurstTorrent, one request contains several pieces. Thus, the energy limited peer can only select regular peers from which it can request a distinct set of pieces. Scanning all peers to find the ones that meet this condition is NP-hard, since this is a constrained version of the classic NP-complete set cover problem, with the pieces as the universe and the pieces already downloaded by each peer as the subsets.

Performing such complex algorithm might only be possible in a smaller network. Thus, we used a different strategy. If a limited peer wants to make x parallel requests, each with p pieces, it starts making request only if it finds at least x regular peers, each with at least $\beta * p$ missing pieces, where $\beta > 1$. This way, the probability that the missing pieces of the selected peers are not overlapping is increased. The limited peer starts making the request to the selected regular peers sequentially, skipping the next regular peer if it does not have at least p pieces that have not been requested. This means that in some cases it is possible that less than the desired number of parallel requests (x) will be made, which decreases the energy efficiency.

Energy limited peers need to notify regular peers connected to them whether they are in active or idle state, so that regular peers can stop sending messages when the energy limited peer is idle and start sending again when it is active. The notification is carried out by sending a maintenance message to each regular peer when the energy-limited peer changes state. The message only contains a flag indicating whether the peer is active or not.

4.2 Determining the Earliest Serving Time

4.2.1 Choking in standard BitTorrent.

Choking means transferring data only to a limited set of peers chosen according to their contribution to the network, and temporarily stopping transfers to the other peers until the next time the choking algorithm is executed. Standard BitTorrent invokes the choking algorithm at every $\omega=10$ seconds. The mechanism works differently if the choking peer is a seed or it is still downloading. If the peer is a seed, the U least recently served

candidates are served. In leechers, the choking mechanism selects $U_{reg}=3$ peers with the fastest upload speeds, and $U_{opt}=1$ peers randomly. The first set of peers is called the regularly unchoked peers, while peers belonging to the second set are called optimistically unchoked peers. It should be noted that peers that have not uploaded any data recently are excluded from the candidates for regular unchoking. This means that it is possible that less than U_{reg} peers are unchoked as regularly unchoked peers, even if there are more than U_{reg} interested peers. This mechanism implements BitTorrent's Tit-for-tat philosophy, giving data to those that contribute the most. Optimistic unchoking serves the purpose of bootstrapping newly joined peers, giving them a chance to download a part of the data which they can then share [7].

4.2.2 Serving time calculation in BurstTorrent

When determining serving times, BurstTorrent categorizes each interested peer as either regular or energy limited. The principal goal of the algorithm is to give regular peers the same amount of bandwidth as if standard BitTorrent were used. This is carried out by calling BitTorrent's choking algorithm, and based on the ratio of limited and regular peers returned, it can be calculated how much time it would take for standard BitTorrent to serve the energy-limited peer. Our model treats all energy-limited peers as if they were free riders, always excluding them from the regular unchokes. This does not affect the bandwidth allocation significantly since energy-limited peers can only upload during active periods, resulting in that they are outperformed by regular peers anyway. Furthermore, we assumed that at least U_{opt} peers are available for the optimistic unchoking. Since by default $U_{opt}=1$, and energy limited peers can only be unchoked optimistically, the assumption is always valid if at least one energy-limited peer is interested in the choking peer's data.

BurstTorrent does not distinguish the individual energy-limited and regular peers. When determining the next serving time, the only thing that matters is how much bandwidth would be allocated to these two groups if standard BitTorrent were used. Based on the formula used by BitTorrent's choking algorithm to select the peers to be served, the fraction of the bandwidth reserved for regular peers is as follows:

$$b_r(n_{ra}, n_{ri}, n_e) = \frac{u_{reg} + U_{opt} \cdot \frac{n_r - u_{reg}}{n - u_{reg}}}{u_{reg} + U_{opt}} \quad (2)$$

where n_{ra} , n_{ri} and n_e are the number of active interested regular, the number of inactive interested regular and the number of interested energy-limited peers respectively. $n_r = n_{ra} + n_{ri}$ is the total number of interested regular peers. Peers are considered inactive if they did not upload any data to the peer executing the choking algorithm during the last three choking periods. Furthermore, $u_{reg} = u_{reg}(n_{ra})$ is the actual number of regular unchokes ($0 \leq u_{reg} \leq U_{reg}$):

$$u_{reg}(n_{ra}) = \min(U_{reg}, n_{ra}) = \frac{U_{reg} + n_{ra} - |U_{reg} - n_{ra}|}{2} \quad (3)$$

Let $N(t) = (n_{ra}(t), n_{ri}(t), n_e(t))$ be a triple corresponding to the number of active regular, inactive regular and energy limited peers that are interested in the choking peer at time t . The earliest serving time in BurstTorrent is determined according to the following formula:

$$\Delta t_s^i = \frac{d^{i-1}}{[1 - b_r(N(t_r^{i-1}))] \cdot S_{u, \max}} \quad (4)$$

$$t_s^i = t_s^{i-1} + \Delta t_s^i \quad (5)$$

where Δt_s^i is the length of the i th serving period, d^i is size of the data transferred during the period and $S_{u, \max}$ is the maximum upload speed of the serving. It can be observed that BitTorrent's choking algorithm is executed when the request arrives (t_r), and by using the calculated bandwidth ratio, the algorithm ensures that the regular peers get enough time to be compensated for the period when the energy limited peers were served.

5. SIMULATION AND RESULTS

5.1 Simulation Framework

In order to evaluate the proposed protocol, we performed flow-level simulations with a simulation framework based the BitTorrent simulator created by K. Eger et al. [8], which uses the ns-2 simulation system. In addition to implementing the new protocol, we also needed to modify and add several features to the original system. One of the most important changes is limiting the peers' download capacity. In the original simulator, only the upload capacity was limited. Since peer-to-peer traffic is transferred via TCP connections, we assume that the upload capacity of a user will be fairly shared among concurrent upload connections, if the maximum possible download rate of each connection is larger or equal to the fair share. All of the standard features of the BitTorrent protocol were simulated, except the end-game mode.

In our case, flow-level simulation means that the TCP behavior is ignored and the simulation takes only the application-layer into account and uses the access link bandwidth as the speed of the network bottleneck. Furthermore, there is no packet loss due to congestion, and the transfer of maintenance messages, whose average size is around 20 bytes, happens instantly.

5.2 Simulation Scenario

The main purpose of the simulations is to show how BurstTorrent performs compared with standard BitTorrent in terms of energy consumption and download time. The simulation scenario is based on the flash-crowd setting which represents an extraordinary burden on the network because only a small set of peers can upload data to others at the beginning. However, in our setup, if a regular peer finishes downloading, it remains in the network to seed the data for an additional 10 percent of its download time, which mimics the way BitTorrent peers behave in real-life. Furthermore, if a regular peer leaves the network, a new peer joins to take its place, so that there are always a constant number of peers transferring the data, simulating a state of the torrent when it is being actively downloaded by an average number of users. The size of the shared data is 100 Mbyte. In the majority of the simulations, we had 5 seeders. The number of

downloaders was 100, and in the different simulation runs we varied the ratio of mobile and regular peers.

In each simulation a certain percentage of the peers were energy limited, and the remaining peers were regular. Both energy limited and regular peers were homogeneous, meaning that all limited peers use the same strategy and all regular peers use standard BitTorrent. The ratio of energy limited peers in the network was varied between 5% and 100%. The access link bandwidth was homogeneous: both limited and regular peers had 1 Mbit download bandwidth and 384 Kbit upload bandwidth, which corresponds to a standard 3G mobile Internet subscription. The simulations were performed five times with each setting, and the results shown here is the average of the measured values.

The model parameters used for calculating the energy consumption are $P_{active}=1.23W$ and $P_{idle}=0W$. These numbers are based on two assumptions. First, the data transfer energy is the dominant cost and we can ignore the processing cost differences of the different solutions in our comparisons. Second, the device would anyhow consume some power in its inactive state. Therefore, we use the additional power that BurstTorrent consumes and subtract the idle state power from the measured 3G cellular power consumption of Section 2.

In the simulations the peers try to request 100 subpieces (16Kbyte * 100 = 1.6 Mbyte) in a scheduled transfer. The effect of the slowdown experienced with TCP connections when data transfer is started is neglected, since according to [9], transferring at least 128 Kbit in one burst already results in close to optimal throughput using 3G/HSDPA.

5.3 Simulation Results

We analyzed how BurstTorrent compares with normal BitTorrent using two different strategies for the energy-limited peers: energy-limited peers only downloaded content (free rider cases) and energy-limited peers also uploaded to regular peers during the active periods (serving cases).

The results depicted in Figure 4 show that the simple and obvious strategy where energy-limited peers are free-riders using standard BitTorrent is bad. In all measures they experienced worse performance than the energy-limited peers which were able to contribute by uploading.

As expected, in terms of energy consumption, BurstTorrent clearly outperformed standard BitTorrent. In the 50% energy-limited case, it achieved 64% less power consumption in the serving mode and 52% in the free rider mode. The downside of the energy saving was the download time that was 54% longer for serving peers and 50% longer for free-riders in comparison to

standard BitTorrent.

With BurstTorrent the serving strategy was superior to the free-riding BurstTorrent in all attributes. This shows that the tit-for-tat mechanism works effectively with BurstTorrent as well. Even if the free-riding strategies are worse in comparison with serving strategies, we cannot ignore them since it can be that energy-limited devices are not able to upload because of the operator policies or other limitations. In those cases BurstTorrent is able to effectively reduce the energy consumption. However, the negative side-effect is that the download time of the regular peers increases dramatically.

Regarding the download time of regular peers, there was a 2% to 10% increase in the case of BurstTorrent free rider compared with standard BitTorrent free rider. However, if uploading was enabled, regular peers achieved shorter download times with BurstTorrent than with standard BitTorrent. This is mainly because energy-limited peers using BurstTorrent do not upload to each other but only to regular peers. Thus, in contrast with standard BitTorrent where all peers might compete for the bandwidth of energy-limited peers, in this case only regular peers can download from them. The higher number of download sources improves the regular peer performance.

Compared with the other cases, the most significant difference in the 100% energy-limited results is that download time of energy-limited peers was significantly higher, except when using BitTorrent in serving mode. Since BurstTorrent peers cannot serve each other, only the initial seeds serve as download sources. BurstTorrent performs similarly as BitTorrent free rider in terms of download time, but its energy consumption is only 10% of the former's. Although BurstTorrent peers could not serve any fixed peers in these simulations, we included both BurstTorrent free rider and serving modes in the figure to preserve consistency throughout the charts.

It can be observed that the energy consumption of the limited peers was less than optimal in all simulation results. This is due to the piece and peer selection strategy described in Section 4. To gain further understanding of the phenomena, we depicted the average active download speed and energy consumption of the energy limited peers in the 50% limited case in Figure 5. Since the average speed during active periods determines the total time spent in active state, this has a major effect on energy consumption. If the idle power is considered zero, the energy consumption is only affected by the average active speed. The peer and piece selection strategy outlined before ensures that most scheduled transfers occur at full speed, close to the optimum; however, as the available download sources decreases, the limited

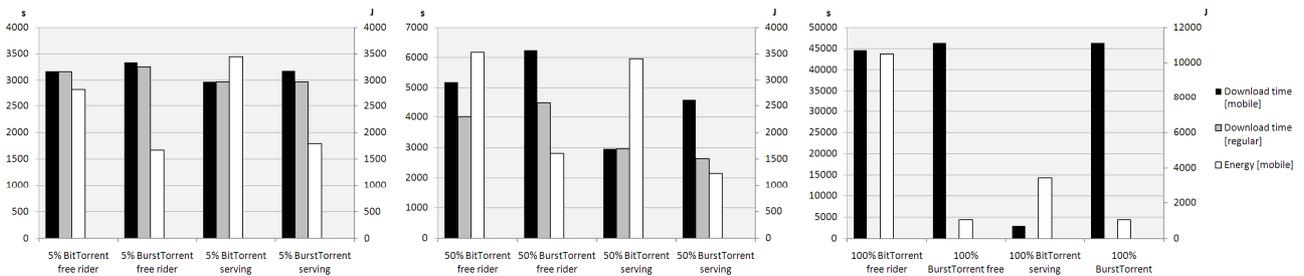


Figure 4. Simulation results with 5%, 50% and 100% of the peers being energy-limited

peer sometimes cannot schedule the desired number of transfers for a given time. This is why the energy consumption increases in the BurstTorrent free rider case. Using the optimal peer selection strategy, which ensures that all transfers occur at full speed, or increasing the β coefficient used by the current algorithm would result in closer to optimal energy consumption, but the download time would also increase. Overall, the currently applied strategy and $\beta=1.5$ value used in the simulations seems to sufficient for most cases.

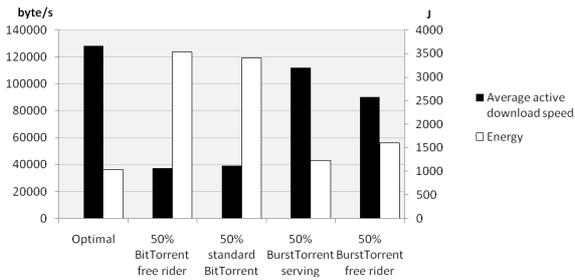


Figure 5. Average download speed and energy consumption, 50% energy limited

We also measured the maintenance overhead of BurstTorrent. The extra traffic was relatively low compared with the size of the content. In the 50% limited simulations, the maintenance overhead was 40-50 Kbyte, which is about 0.05% of the size of the total data transferred. Generally speaking, the maintenance traffic increases as the number of download sources (regular peers) decreases since limited peers receive more refused requests which they need to reissue. It should be noted here that requesting multiple subpieces together actually requires less maintenance traffic than sending out requests one by one as standard BitTorrent does.

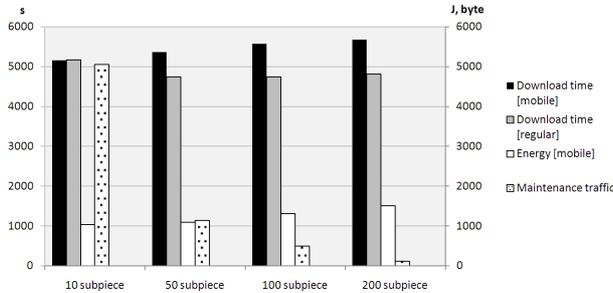


Figure 6. The effect of using different transfer sizes

Figure 6 illustrates the effect of using different scheduled transfer sizes. The chart shows the size of the request as the number of 16-Kbyte-long subpieces per request. The maintenance overhead decreases as the number of subpieces requested increases, since requesting more subpieces together decreases the total number of requests needed. However, using larger requests has a minor negative effect on the download time and the energy consumption of the energy-limited peers. This can be addressed to the decrease of the potential download sources decreases as more and more pieces are required per request.

All of the previously presented simulations were carried out with 5 seeders as the initial source of data. To give further justification to our results, we also performed the simulations with the 50%

energy-limited setup using 5 and 20 initial seeders. Figure 7 shows the differences between BitTorrent and BurstTorrent using serving and free rider strategies. BurstTorrent remained similarly effective in all cases in terms of energy efficiency, achieving around 50% energy savings. However, the download time difference of regular peers was increased as more seeders were added. Nevertheless, the maximum 15% increase in download time observed with the 20 seeder setup is still acceptable.

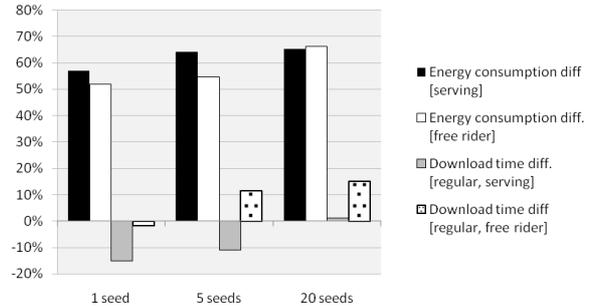


Figure 7. Differences between BurstTorrent and BitTorrent using different number of initial seeds

6. DISCUSSION

The results show that the BurstTorrent approach has a lot of potential in heterogeneous content sharing networks which consist of both energy-limited and regular peers. However, there are at least two difficulties to consider. First, although BurstTorrent is able to operate with current BitTorrent peers, they are not fully compatible. To support the energy-efficient operation both regular and energy-limited peers need to implement the new protocol. Thus, we cannot rely on existing BitTorrent users. Furthermore, we cannot necessarily assume that desktop clients will start using BurstTorrent since even if it does not affect their download times, they do not benefit from using the protocol either.

Moreover, the content that mobile devices would like to download could be different from the content available in desktop clients. The same music files probably work in all devices but handheld devices would require, for instance, different low-resolution video files.

However, a mobile community alone might be sufficient, the devices on chargers acting as regular peers. This would need further evaluations, based on the time an average user leaves their phone on the charger.

In cases dominated by energy-limited devices, long download times limit the applicability of BurstTorrent. It might be useful in cases where download time is not an issue like downloading non-critical software updates or synchronizing media files overnight. Extending BurstTorrent to handle data transfer also between energy-limited devices would require further research. It would require the ability to accept incoming connections on the devices (not possible in most mobile network), and might not improve download times to the desirable extent, since transfers can only be scheduled for active periods. Negotiating several parallel transfers for the same time would be possible only in rare cases.

The area where optimization could be applied to the protocol is the peer and piece selection strategy. The currently applied method results in close to optimal energy consumption if energy-

limited peers were serving regular peers, but in free riding mode and with higher energy-limited peer ratios, the gap between the optimal and the achieved results becomes larger. Therefore, different, more sophisticated peer selection algorithms could be defined to achieve more energy-efficient operation. Since the task of selecting the right peers can be derived from the classic set-cover problem, using the available approximation solutions can also be applied here.

The evaluations used energy values obtained using 3G networks, but the protocol can be operated over other wireless networking technologies as well, most importantly Wireless LAN. The most important difference in the case of WLAN is that the power per transfer speed rate cannot be described by a binary on/off function, since lower transfers speeds requires less power as well. However, the energy per bit ratio is still lower at higher transfer speeds, thus BurstTorrent is more energy-efficient than BitTorrent unless the latter can download data at the highest speed.

7. RELATED WORK

While there are numerous studies about BitTorrent and peer-to-peer content sharing, there is very little research on the energy aspects of content sharing.

Reference [3] investigates the battery consumption of a standard BitTorrent client on mobile phones. Reference [10] studies cooperative content downloading where pieces downloaded from the Internet are shared via proximity connections over short-range radio. Fritsche [11] investigates the possibility of using BitTorrent in sensor networks and observes that the communication is the dominant energy consumer with 63% of total energy.

Blackburn [12] has a very different angle to energy consumption in content sharing. While our work is focusing on the energy consumption of the active downloading phase, his target is the stand-by energy consumption of the seeding servers participating in the BitTorrent networks. His goal is to reduce the total energy consumption and environmental load of computers participating in content sharing networks.

Venugopal et al. [13] presents a time-slotted P2P protocol, in which the slots where the peer is sleeping are determined based on the node id. Other peers that have access to the peer id thus have a chance to know when the peer is going to be in active state. In contrast with our protocol, this mechanism does not focus on content sharing. Furthermore, they use regular sized time slots that are independent of the application or of the peers while our sleep intervals are based on negotiations with other peers.

The research presented in [5] is not P2P related, but discusses an approach similar to ours to achieve energy-efficient mobile web access. The paper states that the minimum possible energy spent for a web page download is obtained by assuming that the transfer from the access point to the mobile host is performed in a single phase. Specifically, the wireless interface is turned on, all data are transferred at the maximum throughput allowed by the wireless link, and then the wireless interface remains off until the next download. This is similar to the way our protocol handles mobile peers and switches between active and idle periods.

8. CONCLUSIONS

In this paper we have shown that significant energy conservation can be achieved by transferring data in high speed bursts.

BurstTorrent clearly outperforms standard BitTorrent in almost all cases in terms of the energy consumption of mobile peers, while the download time of regular peers is not affected significantly. This implies that other file transfer protocol could also benefit from using the concepts presented in this paper. Future work could include evaluating BurstTorrent with different wireless technologies, such as wireless LAN, further analyzing the effects causing the increase of regular peers' download time, and implementing the protocol on a mobile platform to perform measurements in a live network.

9. REFERENCES

- [1] Symella official webpage, available at: <http://symella.aut.bme.hu>
- [2] SymTorrent official webpage, available at: <http://symtorrent.aut.bme.hu>
- [3] J. K. Nurminen and J. Nöyränen, "Energy-Consumption in Mobile Peer-to-Peer – Quantitative Results from File Sharing", 5th IEEE Consumer Communications & Networking Conference (CCNC) 2008
- [4] M. Pedersen, F. Fitzek, "Implementation and Performance Evaluation of Network Coding for Cooperative Mobile Devices", IEEE International Conference on Communications (ICC) 2008, CoCoNet workshop, Beijing
- [5] Anastasi, G., Conti, M., Gregori, E., and Passarella, A. "Performance comparison of power-saving strategies for mobile web access", Perform. Eval. 53, 3-4, Aug. 2003
- [6] Wired.com's iPhone 3G Survey, available at: <http://blog.wired.com/gadgets/2008/08/global-iphone-3.html>
- [7] B. Cohen, "Incentives Build Robustness in BitTorrent," 1st Workshop on Economics of Peer-to-Peer Systems, June 2003
- [8] K. Eger, T. Hoßfeld, A. Binzenhöfer, G. Kunzmann, "Efficient Simulation of Large-Scale P2P Networks: Packet-level vs. Flow-level Simulations", 2nd Workshop on the Use of P2P, GRID and Agents for the Development of Content Networks (UPGRADE-CN'07), USA, June 2007
- [9] Jurvansuu, M.; Prokkola, J.; Hanski, M.; Perala, P., "HSDPA Performance in Live Networks," IEEE International Conference on Communications, 2007. ICC '07, June, 2007
- [10] G. P. Perrucci, F. H. P. Fitzek, M.V. Petersen, „Energy Saving Aspects for Mobile Device Exploiting Heterogeneous Wireless Networks”, Heterogeneous Wireless Access Networks: Architectures and Protocols, Springer 2008
- [11] K. H. Fritsche, "Tinytorrent: Combining bittorrent and sensornets", The University of Dublin, Trinity College, Dublin, Ireland, Tech. Rep. TCD-CS-2005-74, Dec. 2005
- [12] J. Blackburn, "Reducing the Energy Consumption of Peer-to-Peer Networks", 19th Annual Argonne Symposium for Undergraduates in Science, Engineering and Mathematics, 2008
- [13] Madhu Venugopal, Pramod Ramappa, Deep Desai, "Energy efficient P2P protocol," Project Report, UCSB, available at: http://home.umaail.ucsb.edu/~deepdesai/files/Project_Report.pdf