# Content Delivery in Hybrid Networks Using SatTorrent

Bernd Klasen*

SES ASTRA TechCom, Chateau de Betzdorf, Luxembourg
`research@berndklasen.de`

**Abstract.** The global Internet traffic is constantly increasing and consistently reaching the limits of capacity. Commonly this is challenged by upgrading the infrastructure. Alternatively this can be achieved by utilizing existing facilities more efficiently. Using a hybrid network that efficiently combines unicast and broadcast delivery can reduce Internet traffic, server loads and download durations significantly. In order to substantiate this paper introduces and evaluates the payload broadcasting facility of SatTorrent, a peer-to-peer protocol optimized for hybrid networks. The results show that it fulfills our expectations while inducing only a comparatively small amount of additional broadcast traffic.

**Keywords:** content distribution, peer-to-peer protocols, sattorrent, distributed systems, network performance.

## 1 Introduction

The degree to which the Internet and online services pervade our every day life reached a high level and is still increasing. Contemporaneously the bandwidth demand is rapidly growing. If no measures are taken, this leads to congestion and to delayed or failed transfers [11] [6]. One possibility to encounter this problem is to constantly extend the infrastructure. However, this is economically and ecologically not an optimal solution. Much better would be to use the existing infrastructure more efficiently, if possible. The most important traffic drivers in todays Internet usage are—according to studies by [12] and [2]—mainly video content and peer-to-peer (P2P) downloads. Both share common properties: The files transmitted are rather big in size and the same data is often repeatedly transferred to numerous recipients [4], [5], [9] and [8]. For delivering the same content so many recipients broadcast networks such as satellites are well suited. Additionally they provide scalability and great geographical coverage. This work will show that using a hybrid network of satellites and Internet in combination with SatTorrent, a peer-to-peer protocol optimized for hybrid networks—which is presented in section 3. Thereby knowledge of the BitTorrent protocol (see [7], [16]) is postulated.

---

* Funded by *National Research Fund* (FNR) of Luxembourg.

R. Dhaou et al. (Eds.): PSATS 2013, LNICST 123, pp. 65–76, 2013.

## 2 Content Distribution Model

The content distribution model relies on the existence of a hybrid network which is composed of a unicast and broadcast network. Precisely, we assume these two networks to be the terrestrial Internet infrastructure and a satellite network. In principle also other broadcast channels can be used with this approach. The decision for satellites has been taken because they combine good scalability, high bandwidth and outstanding geographical coverage. Most satellite reception hardware currently on the market—e.g. set-top-boxes, television sets with integrated satellite receivers—already provides Internet connectivity and thus the required infrastructure is already in place. However,currently both networks are utilized either in an mutually exclusive manner—watch e.g. either YouTube videos via Internet or HD movies via satellite—or in a way that the Internet channel is used to provide additional information to the broadcasted content. In contrast to these existing solutions, the proposed hybrid network approach uses both networks in conjunction, where delivery of arbitrary content can be dynamically assigned to either of them.

Due to the large fraction of P2P Internet traffic, a P2P protocol is a natural choice to implement our model. This selection is amplified by the ability for substitution of video streaming traffic by P2P streaming even for real time delivery [17] [13]. Among the numerous P2P protocols, BitTorrent was chosen as a basis. This decision has been taken not only because it is the most successful P2P protocol in use, it further employs a tracker as a central entity that obtains an overview of the number of active downloaders. This exactly matches SatTorrent's need for a global entity that is able to decide on what files respectively pieces should be broadcasted and which not. Further details are explained when we introduce SatTorrent in section 3. A more exhaustive description of the content distribution model and exemplary usage scenarios can be found in previous work [9].

## 3 SatTorrent Protocol

The SatTorrent protocol is an extension of the popular BitTorrent protocol. This paper first provides an overview about the basic protocol functioning and those aspects that have been previously introduced in [10] and then concentrates on the newly developed features.

SatTorrent utilizes broadcast networks—e.g. satellites—as additional distribution channel for P2P data. Thereby it supports two operation modes: Broadcasting metadata only and broadcasting payload, the latter being a superset of the former. The metadata broadcast approach is capable to reduce the number of messages in the unicast network while at the same time consuming only a minor amount of satellite bandwidth. This is achieved by broadcasting those messages that contain information that is needed by a large number of peers, such as bitfield and have messages or tracker responses to GetRequests. Not only does this save messages but further increases the information each individual peer is provided with. As a result peers gain a global knowledge about the overlay network and thus are able to choose the best exchange partners. The notable reduction of message complexity and Internet traffic that can be achieved by this approach has been analyzed and the result been presented in [10].

This paper proceeds this work and sheds light on the payload broadcast mode. Since the payload transmission—which is carried by peer wire messages of type *piece*—makes up the largest fraction of the overall traffic, shifting the corresponding piece-messages to the satellite broadcast can be assumed to lead to a much more significant traffic reduction in the unicast network (Internet) than they have already been observed for metadata broadcasts in [10]. As already mentioned in section 2, the tracker plays an important role in the SatTorrent operation. On the one hand it is the central entity where information about peers is aggregated, on the other hand it has to initiate the message broadcasts. The following paragraph introduces the procedural method that is applied in order to conform to this role.

As its ancestor—the *BitTorrent tracker*—the SatTorrent tracker is the first contact point for peers starting a download before they are able to connect to a P2P overlay network and it keeps track of all active peers, which are downloading those files it is responsible for. Let the number of files managed by the tracker be N and the the aforementioned set of files $F$ be defined as $F = \{f | f \text{ is handled by this tracker }\}$. Each file $f \in F$ consists of a specific number of pieces (or chunks) so that $f = \{c | c \text{ is a piece of } f\}$. Further for each $f \in F$ the corresponding tracker holds a list of peers which is defined as $P^f = \{p | p \text{ is currently downloading } f \in F\}$. Further, for each peer known by the tracker, it stores a vector $V^p$ containing the necessary information about peer $p$. This information includes the peer's IP address, the port it is listening on and its ID. So far, this is identical to the information that the BitTorrent tracker stores. For a SatTorrent tracker each $V^p$ contains additional information about the corresponding peer. This is stored in the fields *location*, *sat-enabled* and *bitfield* which are explained in the following.

The *location* information is used for the location awareness feature of the SatTorrent protocol as well as to allow the tracker to identify the corresponding satellite that can be used in order to reach a specific peer. The latter is needed since one satellite can not reach all locations on earth. However, to avoid unnecessary complexity which would hinder the understanding of the concept, in this paper we speak of *a broadcast* even in cases where the same content needs to be broadcasted over more than one satellite in order to reach all nodes. In fact the results presented in [14], [3] and [1] show that content popularity tends to be rather geographically localized and thus in the majority of cases one broadcast using a single satellite is supposed to be sufficient.

The *sat-enabled* entry is a boolean value reflecting the ability of a peer to receive satellite broadcasts. Even if it might be desirable to have all participating peers equipped with the satellite reception hardware, this can not be expected. Therefore the SatTorrent protocol must support heterogeneous peer configurations with peers that can not receive broadcasts. As we will see in section 4,

**Algorithm 1.** Determining piece broadcasts

```
for all $f \in F$ do
    for all $c \in f$ do
        $counter \leftarrow 0$
        for all $p \in P^f$ do
            if $b(f,p,c) = 0$ & $s(p) = 1$ then
                $counter \leftarrow counter + 1$
            end if
        end for
        if $counter \geq BT$ then
            Broadcast piece $c$
        end if
    end for
end for
```

also non sat-enabled peers take profit of the SatTorrent protocol by means of a reduced download duration even for low ratios of sat-enabled peers.

In this context we define the following function:

$$s(p) = \begin{cases} 1, & \text{if peer } p \text{ is sat-enabled} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The last additional entry stored by the tracker is the *bitfield*. This is a list of bitfield information for every file downloaded by a specific peer. We define $B_p^f$ as the bitfield of peer $p$ for file $f$ and the function

$$b(f,p,c) = \begin{cases} 1, & \text{if } p \in P \wedge p \text{ has piece } c \in f \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Having this information, a tracker will trigger a broadcast for piece $c$ of file $f$ only if

$$\sum_{i=0}^{|P^f|-1} \left(b(f,p_i,c) \cdot s(p_i)\right) \geq BT \tag{3}$$

$BT$ is a threshold which must be exceeded in order to trigger a piece broadcast. It denotes the required number of peers that potentially have a demand for the specific piece. The exact value of $BT$ depends on numerous parameters such as for example the available satellite bandwidth, the cost comparison between Internet delivery and broadcast and whether there is a competitive situation of several broadcast attempts that can not all be served within an acceptable time frame. Narrowing down this threshold for a specific scenario is not subject of this paper. Instead we evaluate the performance of SatTorrent under varying values of $BT$ and present the corresponding results in section 4. The procedure that a tracker periodically executes in order to determine the pieces that are suitable for a broadcast is illustrated as pseudo code in algorithm 1. For each piece of every file it counts the number of peers that are downloading the corresponding file but still do not possess that piece. After this has been evaluated for all files, those pieces where the number of demanding peers exceeds $BT$ are passed to the uplink station for broadcast. In current implementation there is no bound for the amount of data that the tracker is allowed to send to the uplink. However, the latter has only a limited satellite bandwidth at its disposal. Currently we just monitor the broadcast queue in order to detect congestion or under utilization. Future implementations will have a feedback mechanism that triggers an adjustment of $BT$ at the tracker according to the satellite load.

Before we examine how peers handle broadcast messages we must clarify how the information vectors $V^p$ respectively is the information that is used by algorithm 1 is maintained? Most of it is sent from a peer to the tracker with the initial GetRequest when it starts downloading a file and is seldom changed afterwards. *ID*, *IP address*, *port* and *sat-enabled* do not change frequently. Also the *location* does only change for mobile clients. However, the *bitfield* entry must be updated immediately when a peer's bitfield changes, as becomes obvious by its central role in the piece broadcast algorithm presented above. In order to achieve this, whenever a peer receives a *piece message*—besides of updating its own local bitfield—it also sends an extra *have message* to the tracker, who in turn updates the corresponding bitfield. It is worth noting that although this puts additional messages on the unicast network, the total number of *have messages* is much smaller compared to BitTorrent as we will show in section 4. This is due to the *have message suppression* feature of SatTorrent which does not only avoid *have messages* to peers that are known for already possessing this piece—as the equally named BitTorrent feature does. It further is able to eliminate *have messages* whose recipient is sat-enabled. The have messages to sat-enabled peers are substituted by metadata broadcasts (see [10]).

A peer that receives file pieces by broadcast stores them and immediately cancels all potentially pending requests for these pieces and sends updated interest status messages to the connected peers accordingly. Since each sat-enabled peer can potentially receive all piece broadcasts, no matter whether it is downloading the corresponding file or not, different modes for handling this data are provided. Either a peer can only store data for files it is downloading or it might store all data for potential future downloads. Another possibility is to apply collaborative filters, use recommendations, ratings and personal preferences in order to identify those files that might be interesting and in consequence probably will be requested in the near future. Based on this the peer can make a decision on whether the data should be kept or withdrawn in case the file is not being downloaded. However, this feature is not implemented yet and subject of future work.

## 4 Evaluation

In order to evaluate SatTorrent's performance under varying conditions in comparison to BitTorrent and how parameters should be adjusted to obtain the best attainable results for a specific purpose, the protocol has been implemented in a simulator. In order to ensure reproducibility of the achieved results, each simulation is unambiguously specified by a configuration file.

The scalability of the simulator is mostly limited by the memory consumption, which primarily depends on two factors, the number of peers and the size of the files that are to be exchanged. The former is obvious, the latter is due to the fact that every peer has to save the bitfields for all connected peers, while the size of that data structure is determined by the number of pieces a file is partitioned into and the number of files that are being downloaded. However, the piece size can not arbitrarily be increased since this causes performance degradation and thus quantity depends on the filesize. In order to increase the maximum number of peers and the size respectively the count of files that can be simulated, measures must be taken in order to reduce the memory demand as much as possible. One is the geographical limitation of the simulation. This means an assumption is made that all nodes reside within the same satellite footprint and thus when a broadcast is made, potentially all nodes can receive it (in case they are sat-enabled). Further an abstraction for the location awareness is made in a way that it determines the distance by means of autonomous system (AS) membership. In case two nodes are within the same AS they are considered being nearby, otherwise they are distant. We argue that the traffic within one AS—which in the majority of cases means that is also is within the same Internet service provider (ISP) network—comes with low cost and little delay. In contrast, traffic that is crossing AS boundaries is expensive for ISPs and has a higher probability for an increased delay. This coherence has also been highlighted in [15] who further find that the inter ISP network traffic is limiting the performance of P2P approaches since ISPs tend to throttle P2P traffic. Thus using location awareness to keep traffic within one AS as much as possible is reasonable.

Another measure to limit the simulation's complexity and its memory demand is a reduction of message granularity. In real SatTorrent traffic the peer wire messages of type *piece* do not carry a complete piece but a block of data of a piece. Thus one piece is submitted by several piece messages. However, using such a fine granularity for the simulation further increases the memory consumption and thus decreases the number of network nodes that can be simulated. Thus in the simulation a piece message always carries a complete piece. For a better distinction between real world piece messages carrying blocks and those in the simulator, we refer to the former as *block messages*. The rate in which piece messages are being sent has been adjusted accordingly. Further, since each peer wire message comes with a certain overhead, the number of blocks a piece is partitioned into must be considered in order to calculate the real bandwidth consumption of a piece message. The common block size used in most current BitTorrent client implementations is 16 kByte per block. Considering a piece size of 512 kByte, this results in 32 blocks per piece. The overhead per block message is 72 bit = 9 Byte. This means each piece message in the simulation generates a network traffic of $512kByte + 32 \cdot 9Byte = 524,576Byte$. Since the simulator counts only the number of piece messages, the results presented in this paper are adapted accordingly.

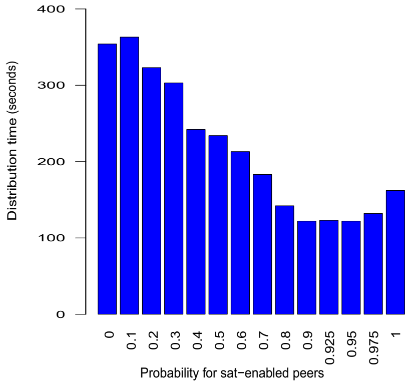

**Fig. 1.** Distribution time under different proportions of sat-enabled peers

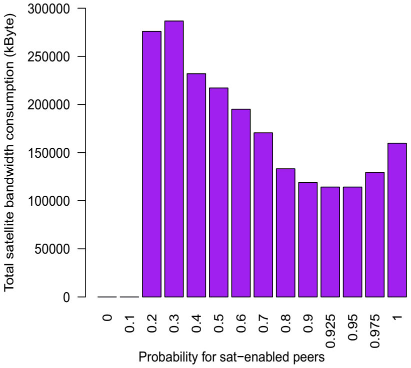

**Fig. 2.** Satellite bandwidth consumption for varying SSP

### 4.1 Specific Simulation Settings

Since it is not possible to describe each simulation parameter in detail, we will shortly discuss those settings which remain unchanged during all simulations but are important for the assessment of the results in this section. The parameters that are subject of the evaluation are described later in section 4.2. One of the parameters that are statically assigned is the satellite transponder bandwidth. Here we use a conservative value of 36Mbit. This is what even the older operational geostationary television satellites are capable of after deducting error correction (net bandwidth). Further, as already indicated above, a piece size of 512 kByte has been used. According to [16] this is a common value for BitTorrent and represents a good trade off between torrent-file size and efficiency for that protocol. In order to provide an equitable comparison between BitTorrent and SatTorrent—respectively the solely unicast delivery and a satellite broadcast aided approach—it is vitally important to apply the best possible settings for BitTorrent.

### 4.2 Results

As a first performance evaluation we compare the distribution time and the aggregated bandwidth demand for SatTorrent and BitTorrent. The latter is equivalent to SatTorrent without any sat-enabled peers. Since we do not expect to have 100% of peers being sat-enabled in a real world scenario, we analyze the results for a file of 20MB being distributed to 1000 peers with $BT = 150$ under varying number of sat-enabled peers. In the simulation settings we can provide the *SatelliteSupportProbability* (SSP) which denominates the probability of a joining peer for being sat-enabled. Figure 1 shows the distribution time for SSP ranging from 0.0 to 1.0. Since the curve progression changes between SSP=0.9 and SSP=1.0, more fine grained steps are provided within this interval. What we

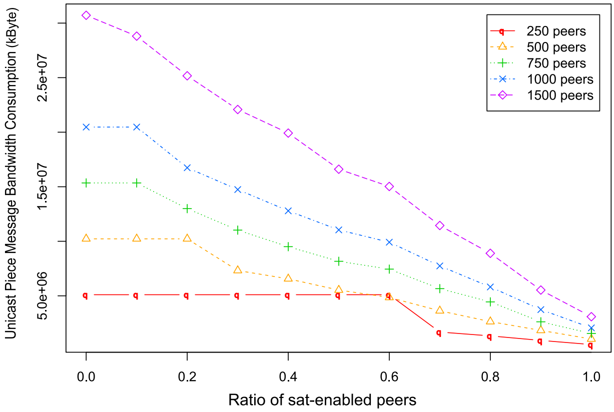

**Fig. 3.** Internet bandwidth consumption for different values of SSP

can see is a reduction of the distribution time for increasing ratios of sat-enabled peers except for settings where all or almost all peers are sat-enabled. This effect originates from the increased start delay for sat-enabled peers which have to wait longer for a tracker response on their get-requests. These metadata broadcasts are performed within fixed time intervals while non sat-enabled peers receive their answers immediately. This negative effect can be levelled out only as long as there is a reasonable number of non sat-enabled peers. However, it can probably be eliminated or at least be reduced by optimizing the tracker algorithm for handling get-requests from sat-enabled peers which is subject of future work. Another increase can be observed when the SSP changes from 0.0 to 0.1. The reason becomes clear by the results for the overall consumed satellite bandwidth shown in figure 2. For $SSP = 0.0$ there is obviously nothing broadcasted. The same applies for $SSP = 0.1$ since a sufficient number of sat-enabled peers is not reached in order to satisfy the condition given in equation 3. Thus only a small delay due to the metadata broadcasts which has been described above is introduced while the corresponding benefit—a reduced number of total unicast messages—is not reflected in these figures. At the same time, there are too little sat-enabled peers in order to take a considerable advantage of the metadata broadcasts. We will further discuss the characteristics of the satellite bandwidth demand under changing values of $BT$ later when we make a comparison for different network sizes.

The major advantages of the payload broadcasts can be observed in figure 4 where the aggregated bandwidth consumption for the broadcasts and the unicast piece messages is shown. On the one hand we observe a monotonic decrease of total bandwidth demand for increasing values of SSP, on the other it manifests the high potential for reduction of unicast network traffic while at the same time allocating only marginal bandwidth on the broadcast network.

A further important aspect that must be analyzed is SatTorrent's performance for various sizes P2P networks, particularly smaller ones. Therefore the same file distribution has been analyzed for varying numbers of peers. In figure 3 and

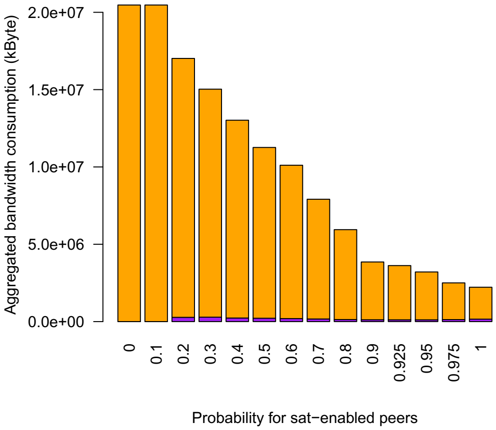

**Fig. 4.** Aggregated bandwidth consumption for varying proportions of sat-enabled peers

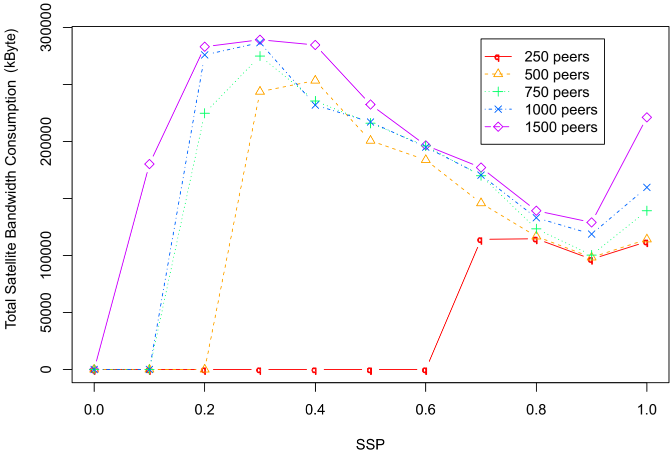

**Fig. 5.** Satellite bandwidth consumption for different values of SSP

5 we see the bandwidth consumption for different network sizes in the Internet respectively in the satellite network. The behavior with respect to the unicast messages (figure 3) is straightforward and satisfies our expectations: The bandwidth demand for payload delivery is continuously decreasing with the proportion of sat-enabled peers. The horizontal line segments in the plot reveal the only condition: The total number of sat-enabled peers must at least be equal or greater than the broadcast threshold, which is not surprising at all.

Even though the curves in figure 5 all exhibit a similar shape, their progression is by far less self-evident than the ones for Internet bandwidth consumption and

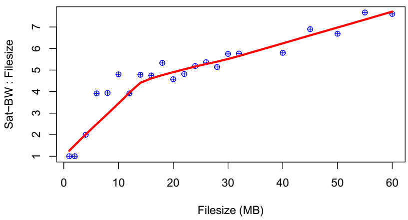

**Fig. 6.** Satellite bandwidth consumption relative to filesize

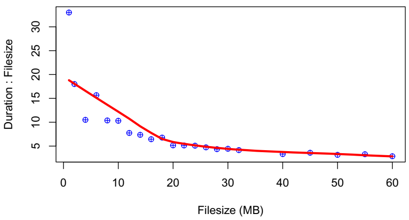

**Fig. 7.** Overall delivery duration in relation to filesize

takes a deeper investigation. First we observe a steep increase immediately after the SSP reaches a value that allows the number of sat-enabled peer to exceed the broadcast threshold $BT$. The fact that we observe a peak at $SSP = 0.3$ for network sizes of 750, 1000 and 1500 peers indicates that this high demand for satellite bandwidth is not depending on the absolute number of sat-enabled peers in relation to $BT$. The reason seems to be in the ratio between sat-enabled and non-sat-enabled peers. On the one hand peers that are able to receive pieces by broadcast don't wait idle for satellite transmissions but continuously exchange pieces with other peers in the network. On the other hand, the piece selection strategy leads to a uniform distribution of pieces at the peers. In consequence we might in many cases only scarcely fulfill the condition given in equation 3 even though the number of sat-enabled peers is twice or thrice as much as $BT$. With an increasing density of sat-enabled peers the probability that each piece broadcast is received by a larger number of peers that still need that specific part of the file grows, and thus the efficiency of the broadcast increases. The result are the decreasing curves we find after the peak.

This is true except for situations where all peers are sat-enabled. The main reason for this is the complete lost of non sat-enabled peers which are less restrictive in their behavior. Sat-enabled peers avoid connections to peers when they can not be sure about their usefulness. This leads to an increased time demand until the data has been completely delivered to all peers as we have already observed in figure 1. The positive aspect—which is not reflected in this figure—is that it also further reduces the unicast traffic. Whether this is desirable at this cost depends on the cost ratio between satellite and Internet bandwidth allocation and on the load situation on the satellite. However, there are two aspects that mitigate this undesired behavior. The first is that—as we discussed earlier—in a real life environment we will hardly ever reach a configuration where all peers are sat-enabled. Second, there is still room for optimizing the protocol parameters for $SSP = 1.0$ which will presumably reduce the bandwidth demand.

Next we investigate the impact of the file size on the performance of SatTorrent. For these measurements we simulated a distribution among 500 peers with $BT = 150$ and $SSP = 0.9$. The latter has been chosen since it delivered good results with respect to the time demand for the distribution as well as for the satellite bandwidth consumption. In general SatTorrent's performance can be expected to increase with the size of the files being distributed since that also augments the probability to have many concurrent downloads. In consequence each broadcast can be received by a potentially higher number of peers which constitutes an elevated efficiency. By applying parameters that are nearly optimal also for smaller files we reduce the influence of other factors except file sizes in this measurement. The results can be seen in figures 6 and 7. The correlation between file size and the relative satellite bandwidth demand becomes obvious in figure 6. The decreasing gradient that we find here confirms our assumption of an increasing SatTorrent efficiency for larger files. The same applies for the relative distribution time that is decreasing with growing file sizes (see figure 7). At last SatTorrent has been analyzed under varying values of $BT$. The expectation that altering this parameter would have a significant effect on the distribution time and on the satellite bandwidth demand has not been confirmed. Evaluations show that the reduction in bandwidth demand at growing SSPs is rather small ($< 10\%$). However, we see great potential to achieve benefits from larger values for $BT$—adjusted relative to the file sizes—in conjunction with an improved piece selection strategy which is subject to future work. Further this might change in a competitive situation with several concurrent attempts to allocate satellite bandwidth.

## 5 Conclusion and Future Work

In this paper the payload broadcast feature of the SatTorrent protocol has been introduced. The evaluation that followed revealed its potential to reduce the traffic in the Internet by means of utilizing a comparatively small amount of satellite bandwidth. Further several aspects of SatTorrent have been observed which bear potential for improvement. The most important among these is the optimization of the piece selection strategy for sat-enabled peers and the further measures against the increasing satellite bandwidth demand for high ratios of sat-enabled peers. Besides that, the future work includes the study on how social network structures can be utilized to further improve the delivery. This includes file demand prediction as well as enhanced file exchange options that are based on the increased trust between nodes with small distances and strong connection in the social graph.

## References

1. Backstrom, L., Sun, E., Marlow, C.: Find me if you can: improving geographical prediction with social and spatial proximity. In: WWW 2010 Proceedings of the 19th International Conference on World Wide Web, pp. 61–70. ACM, New York (2010)
2. Basher, N., Mahanti, A., Williamson, C., Arlitt, M.: A comparative analysis of web and peer-to-peer traffic. In: Proceeding of the 17th International Conference on World Wide Web, pp. 287–296. ACM (2008)

3. Brodersen, A.: YouTube Around the World: Geographic Popularity of Videos. In: WWW 2012 Proceedings of the 21st International Conference on World Wide Web, pp. 241–250. ACM, New York (2012)
4. Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., Moon, S.: I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, pp. 1–14. ACM, New York (2007)
5. Cheng, X., Liu, J., Wang, H.: Accelerating YouTube with video correlation. In: Proceedings of the First SIGMM Workshop on Social Media - WSM 2009, p. 49. ACM Press, New York (2009)
6. Cho, K., Fukuda, K., Esaki, H., Kato, A.: The impact and implications of the growth in residential user-to-user traffic. In: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications - SIGCOMM 2006, p. 207 (2006)
7. Cohen, B.: The BitTorrent Protocol Specification (2009)
8. Gill, P., Arlitt, M., Li, Z., Mahanti, A.: Youtube traffic characterization: a view from the edge. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, pp. 15–28. ACM (2007)
9. Klasen, B.: Efficient Content Distribution in Social-Aware Hybrid Networks. Journal of Computational Science (2011)
10. Klasen, B.: SatTorrent - Satellite-Aided P2P Content Distribution for Virtual Environments. In: Proceedings of the Fifth International Conference on Simulation Tools and Techniques. ACM, Desenzano (2012)
11. Leighton, T.: Improving Performance on the Internet. Queue 6(6), 20 (2008)
12. Maier, G., Feldmann, A., Paxson, V., Allman, M.: On dominant characteristics of residential broadband internet traffic. In: Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, IMC 2009, vol. 9, p. 90. ACM Press, New York (2009)
13. Ramzan, N., Quacchio, E., Zgaljic, T., Asioli, S., Celetto, L., Izquierdo, E., Rovati, F.: Peer-to-peer streaming of scalable video in future Internet applications. IEEE Communications Magazine 49(3), 128–135 (2011)
14. Scellato, S., Mascolo, C., Musolesi, M., Crowcroft, J.: Track Globally, Deliver Locally: Improving Content Delivery Networks by Tracking Geographic Social Cascades Categories and Subject Descriptors. In: WWW 2011 Proceedings of the 20th International Conference on World Wide Web, pp. 457–466. ACM, New York (2011)
15. Choffnes, D.R., Bustamante, F.E.: Taming the Torrent: A Practical Approach to Reducing Cross-ISP Traffic in P2P Systems. ACM SIGCOMM Computer Communication Review 38(4), 363–374 (2008)
16. Theory.org Wiki contributors. Bittorrent Protocol Specification v1.0 (2013), `http://wiki.theory.org/BitTorrentSpecification`
17. Yin, H., Liu, X., Zhan, T., Sekar, V., Qiu, F., Lin, C., Zhang, H., Li, B.: Design and deployment of a hybrid CDN-P2P system for live video streaming. In: Proceedings of the Seventeen ACM International Conference on Multimedia - MM 2009, ACM Press, New York (2009)