

Experimental Study of *Broadcatching* in BitTorrent

Zengbin Zhang[†], Yuan Lin[†], Yang Chen[†], Yongqiang Xiong[‡], Jacky Shen[‡], Hongqiang Liu[†], Beixing Deng[†], Xing Li[†]

[†]Tsinghua National Laboratory for Information Science & Technology,

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Email: zzb02@mails.tsinghua.edu.cn

[‡]Microsoft Research Asia

E-mail: yqx@microsoft.com

Abstract—*Broadcatching* is a promising mechanism to improve the experience of BitTorrent users by automatically downloading files advertised through RSS feeds. However, though widely used, the mechanism itself has not been well studied. In this paper, we conducted extensive experiments on PlanetLab to evaluate the performance of *Broadcatching* under different typical scenarios. The results demonstrated the effectiveness of the *Broadcatching*: it reduces the average completion time and downloading failure ratio. It also improves the overall fairness of the system: the subscribers are encouraged to share more while downloading faster, which results in the increased share ratio. Our study is the first work to systematically evaluate the benefit of *Broadcatching* and sheds lights on how to improve performance of BitTorrent by manipulating peer's behavior like *Broadcatching*.

I. INTRODUCTION

Peer-to-Peer (P2P) file sharing protocol has been widely used on Internet. According to a study of Internet traffic in five regions of the world between August and September 2007 performed by IPOQUE [4], P2P file sharing is still the application class producing the most Internet traffic. Its share varies, in their observations, between 48% in the Middle East and 80% in Eastern Europe.

BitTorrent (BT) has become by far the most popular P2P file sharing protocol worldwide thanks to its cooperative nature, but the performance of BT system is greatly affected by peers' behavior. Research has been conducted to study the performance of the BitTorrent systems in the literature. According to [2], the author analyzes the BitTorrent meta file downloading trace from a large commercial server farm hosted by a major ISP and finds that the population size of the torrent is relatively small, though there are several large torrents, 102 peers on average. In [1], the author demonstrates through simulation that the later a peer joins the torrent, the smaller its peer set size. Hence, we can conclude that the service availability of a torrent will tend to become poor quickly after the birth of the torrent, and it will become more and more difficult for a peer to locate the resources, leading to large downloading failure rate and bad downloading performance.

Broadcatching ([12], [13], [14]) is an emerging and promising technique to improve the BT performance through manipulating peer behavior with RSS. RSS (also known as "Really Simple Syndication") [6] is a way to publish information so that other computers can read it in a simple and standard way. RSS allows users to keep up-to-date with

content from sites they are interested in; thus, many BitTorrent sites have started publishing RSS feeds of their listings, and let BT clients automatically download file advertised with RSS feeds according to RSS refresh behavior, which is called *Broadcatching*, aiming to improve the overall user experience. Several broadcatching-capable BT tools have already been developed (μ Torrent [8], Azureus [7], etc), and they have become quite popular on Internet, especially in countries such as Australia and UK [15].

In literature, however, little attention has been paid on *Broadcatching* to give a comprehensive study of its performance. In a technical report [3], the author mentioned that Broadcatching can increase the seeding ratio, but he did not provide sufficient detailed analysis. Moreover, the performance of Broadcatching may vary significantly under different settings of BT protocol and different network conditions, but that case study did not cover various typical scenarios. Therefore, the existing research result is quite far from satisfying, and detailed analysis is needed.

In this paper, we conducted an extensive experimental study on the performance of the Broadcatching mechanism in BT system. Based on real Internet environment on Planetlab [11], different scenarios were carefully designed and studied. We attempt to find out answers for the following questions. a) How much gain can we get from Broadcatching? b) Does it affect or improve fairness in BT systems? c) In what configuration does Broadcatching improve the overall downloading performance most? Our results show that Broadcatching greatly improve the performance of the BT system by up to 85%, and the average share ratio has been increased by 80%, compared to a bad configuration. Moreover, under certain seeding environment, there DOES exist an optimal value for RSS update interval. Through analyzing the experimental results, we provide some guidelines on how to control Broadcatching to get an optimal performance.

The contributions of this paper are two folds. One one hand, we evaluate the effectiveness of Broadcatching mechanism in BT system with extensive experimental results. Well-designed scenarios are implemented in the evaluation experiments on planetlab with various peer behaviors as well as system settings. On the other hand, we analyze the performance as well as the fairness effect of Broadcatching mechanism. The detailed analysis offers valuable insight on how to implement

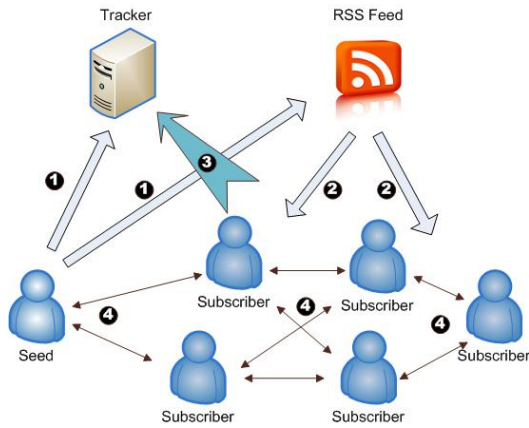


Fig. 1. System Structure Of Broadcatching

such a user-behavior-changing technique to get a better performance.

II. BITTORRENT AND BROADCASTING

In this section we will briefly introduce the mechanism of BT and Broadcatching.

In the BT system, a centralized *tracker* is implemented to enable coordination between peers. The tracker maintains the set for peers interested in the same content, which is also known as *swarm*. To share a file or a group of files in a BT system, a torrent file is generated first which contains the address of the tracker. A peer who is interested in the content must download the torrent file and then join the swarm by announcing itself to the tracker, which will return a small random subset of peers from the swarm. Peers use this subset to connect to other peers to exchange pieces of the file. These pieces of the file are called *trunks* in a BT system. Peers with the whole file are called *seeds* while others are called *leechers*.

In the Broadcatching scheme, there are mainly four parts to be considered, namely, *Tracker*, *Website*, *Seed*, *Downloaders*. *Downloaders* are assumed to be interested in some particular contents, such as periodically updated TV programs or popular movies. The *Website* contains a RSS feed for each kind of content, called a *channel*, accordingly we divide *Downloaders* into two groups: those downloaders who have subscribed to their interested channel are called *Subscribers* of this channel, while others called *Normal Leechers*.

The subscribers need to use broadcatching-capable BT client to subscribe to their interested channels. The built-in RSS reader will periodically check the feeds and start downloading once there're new updates. Different *RSS Update Interval* settings are used in various broadcatching BT clients, including fixed ones like 1 hour, 12 hours, 24 hours, and self-adaptable ones according to the update interval of the feeds.

Fig.1 shows the structure and operation steps of Broadcatching. We briefly describe the procedure of the system as follows.

Step 1. The seed publishes a content. It firstly generates the torrent file and then notifies the tracker. At the same time, the Broadcatching BT client of the seed connects to the Website and creates a new term in the corresponding RSS feed.

Step 2. The subscribers check the feeds. During a *RSS update interval*, the subscribers will come to check if there is an update. However, due to the asynchronism of the subscribers, they will not likely come at the same time. In our experiments, we assume that the arrival rate is an uniform distribution in the RSS update interval.

Step 3. The subscribers get the peer set. They connect to the tracker immediately after finding the update. The tracker returns a peer list to each subscriber, in which the peers are randomly chosen from the swarm. Usually the normal leechers connect to the tracker much later and more diversely just like in a traditional BT system.

Step 4. The peers start downloading. The whole swarm forms a random connected overlay. Trunks from the original seed will be distributed to the subscribers and normal leechers.

Unlike traditional BT system, Broadcatching will aggregate the subscribers together immediately after the content is published during a RSS update interval. Intuitively, this mechanism will bring large amount of online users so that everyone can get enough neighbors from the swarm and download faster than normal. However, the overall benefits and quantitative analysis of this peer-behavior-manipulation technique haven't been well studied. What else can we get from Broadcatching and what is the limitation of Broadcatching remain unknown. In the following section, we present well-designed scenarios to make a comprehensive evaluation of Broadcatching.

III. PERFORMANCE EVALUATION

A. Experiment Setup

We conduct real experiments on planetlab to evaluate the performance of Broadcatching from April.2008 to June.2008. About 200 Planetlab nodes are chosen in our experiments, among which planetlab1.csail.mit.edu is used as the tracker, thu1.6planetlab.edu.cn as the initial seed, and thu2.6planetlab.edu.cn as the web server. We use BitTornado [9] for both tracker and clients. The size of the file for downloading is 157MB and the upload rate of the initial seed is limited to 500KB/s as a default value. We installed Apache on thu2.6planetlab.edu.cn and generated a RSS feed on it.

To achieve the functionality of Broadcatching, we modified the BitTornado client and combined it with a RSS reader, which will check our RSS feed periodically. Due to the asynchronism between planetlab nodes, all the subscribers tend to come up randomly during the RSS update interval. For normal leechers, we assume that they will come randomly in a long period after the birth of the torrent, which is a common assumption in literature.

We also activate the Super-Seeding mode in the seeds, which is firstly introduced in BitTornado and has been implemented in most BitTorrent versions. Super-seeding improves the efficiency in producing multiple seeds while reducing the uploading cost of the initial seeders.

As to the peer behavior, in the real world, most peers tend to close the client soon after completing their download. In the experiments, we consider the users to be selfish.

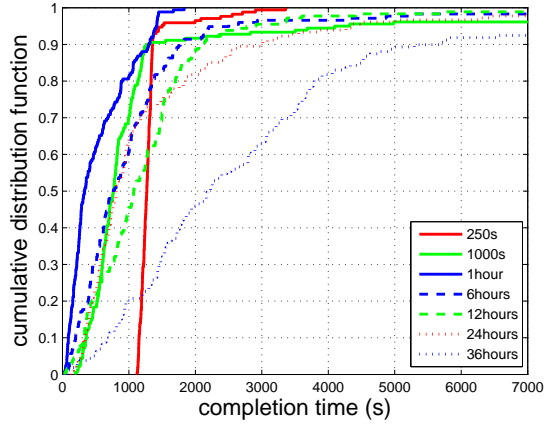


Fig. 2. Completion Time Distribution Varying RSS Update Interval

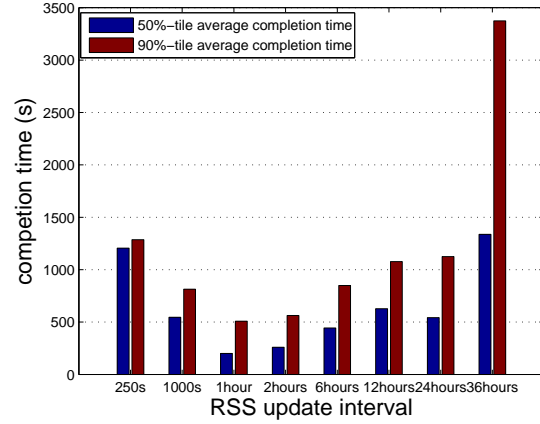


Fig. 3. 90-percentile Completion Time Average Varying RSS Update Interval

B. Performance Metrics

Based on the key concern of both the system as well as the seeders, we use three metrics to evaluate Broadcatching scheme under different settings.

Completion Time: The time for a client to download the entire file. Both the average completion time of the whole system and the cumulative distribution of completion time of all clients are considered. This metric represents the basic performance of the system. In most of our experiments, due to various problems, there are no more than 10% of the peers that can not finish their downloading. To eliminate the influence of these peers, we use the average completion time of 90-percentile peer as one of our metrics.

Share Ratio: We define the Share Ratio as follows:

$$\text{Share_Ratio} = \frac{\text{size_of_total_uploading}}{\text{size_of_the_downloaded_file}}$$

This metric evaluates the fairness of the system.

Seed Upload Ratio: We define the Seed Upload Ratio as follows:

$$\text{Seed_Upload_Ratio} = \frac{\text{size_of_total_uploading}}{\text{size_of_the_original_file}}$$

This metric measures the upload cost of the initial seed.

C. Experiment Results

To evaluate the performance of under different settings and user behaviors, we proposed several scenarios, i.e. varying RSS update interval, subscriber ratio, initial seed number, initial seed upload bandwidth.

1) *Varying RSS Update Interval:* In this experiment, we vary the RSS update interval to see its influence on the performance of the system. Series of values of RSS update interval are used in the experiment: 250s, 1000s, 1 hour, 2 hours, 6 hours, 12 hours, 24 hours, 36 hours.

As the fig.2 shows, the completion time tends to be much smaller under a relatively short RSS update interval, so is the average download time as the fig.3 shows. This result is quite straightforward. In this case, when a subscriber gets the

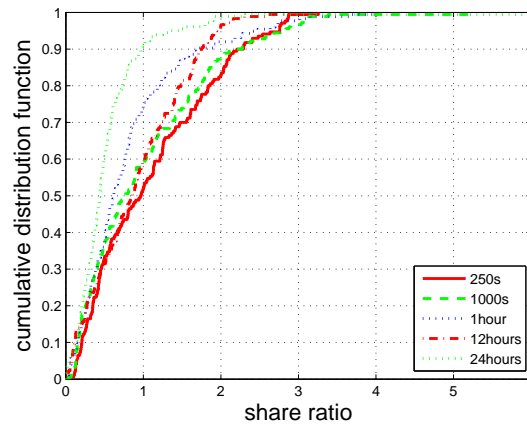


Fig. 4. Leacher Share Ratio Varying RSS Update Interval

RSS notification and connects to the tracker, there are still many peers downloading, so it can get more resources and download much faster. However, we can also get some interesting observations from the result. At a RSS update interval of 1 hour, the download performance of the whole system reaches the optimum. At a period of 250s, the completion time variation is quite small but the average completion time is not the optimum. That is because of the flash crowd effect: when all the peers come in a very short time after the birth of the torrent, the limited seeding capacity is not large enough so no one can get the whole file at first. However, once the seed has uploaded the whole file, all the peers will complete their downloading very quickly, and roughly at the same time.

From the fig.2 we can also conclude that the downloading failure ratio has been improved in a Broadcatching mode. That is because it's much easier for a peer to find a seed under this mechanism.

Fig.4 shows the cumulative distribution of the share ratio of all peers. We can see that in a shorter RSS update interval, everyone contributes more to others. As a result, the whole system is much healthier, which embodies the spirit of Peer-

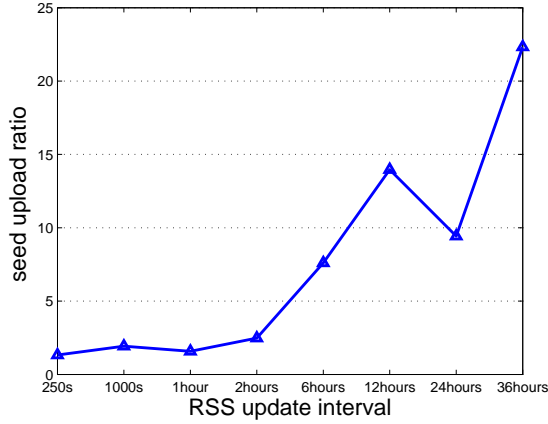


Fig. 5. Seed Upload Ratio Varying RSS Update Interval

to-Peer mechanism. Remember that here we have assumed that all the peers are selfish. Even in this case, the average upload ratio of all the peers is up to 1.0118 at a RSS update interval of 1000s. The reason it can be more than 1 is that we didn't count in the peers that hadn't finished downloading, which was a very small portion as we have explained above.

We can see from fig.5 that the initial seed upload ratio is rather small in Broadcatching mechanism(only 1.6 at a interval of 1 hour), which is a direct result of the share ratio improvement of all peers. A small upload ratio will guarantee the cost to be a seed is low, and thus can encourage the content publisher to stay longer. We have observed that at a period of 24 hours, the seed upload ratio is smaller than that at a period of 12 hours. However, through later analysis we found that it was because one peer didn't kill itself after downloading and served as a seed.

To be simple, we only use two RSS update intervals in the following experiments: 1000s and 24 hours.

2) *Varying Subscriber Ratios*: In this experiment, we show the influence of different Subscriber Ratio among all peers. We use three different subscriber ratios: 75%, 50%, 25%. All subscribers will randomly come during the 1000s-period after the publication of the content while the normal peers will randomly come during the 24-hour-period after the content is published.

As the fig.6 shows,the completion time decreases significantly as the ratio of the subscribers increases. And not surprisingly, the average completion time is smaller with a larger subscriber ratio.

TABLE I
SEED UPLOAD RATIO VARYING SUBSCRIBER RATIO

Update Interval	75%	50%	25%
Seed Upload Ratio	4.17	14.10	16.09

Table.I shows the seed upload ratio of the three conditions. With a higher subscriber ratio, everyone shares more and as a consequence the seed uploads much less.

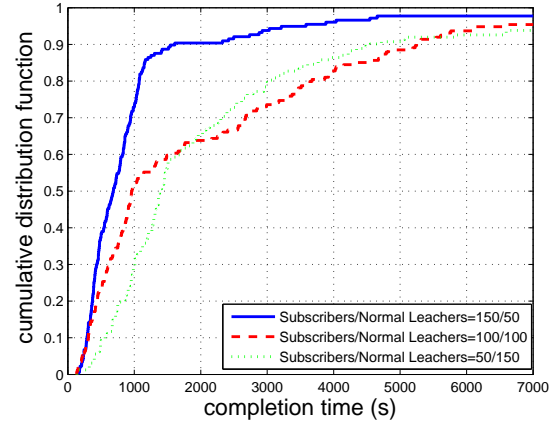


Fig. 6. Completion Time Distribution Varying Subscriber Ratio

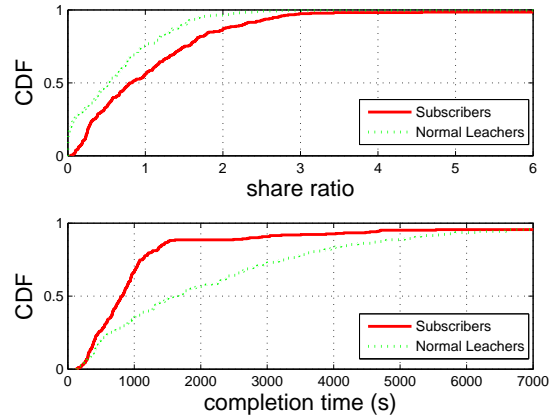


Fig. 7. Completion Time VS Upload Ratio

In order to explore the benefits of the subscribers, we collected the completion time and share ratio of each peer and then separated them apart as *subscribers* and *normal leachers*. Then we plotted the cumulative distribution of two metrics for each group in the fig.7. The top half part of fig.7 plots the upload ratio distribution of subscribers and normal leachers while the bottom one shows the completion time distribution of each group. The results show that, a subscriber can get a much faster download speed while contributing more to the system. We believe that this benefit will attract a single peer to use the Broadcatching mechanism, hence can improve the popularity of the Broadcatching as well as the robustness of the BT system.

3) *Varying Capacity of Initial Seeds*: In this section, scenarios are designed to see the impacts of changing capacity of the initial seed. Firstly we varied the initial seed upload bandwidth its influence on the performance of Broadcatching mechanism. Only one seed was used with three bandwidth settings: 50KB/s, 100KB/s and 500KB/s.

The completion time distributions of the six experiments were plotted in fig.8.

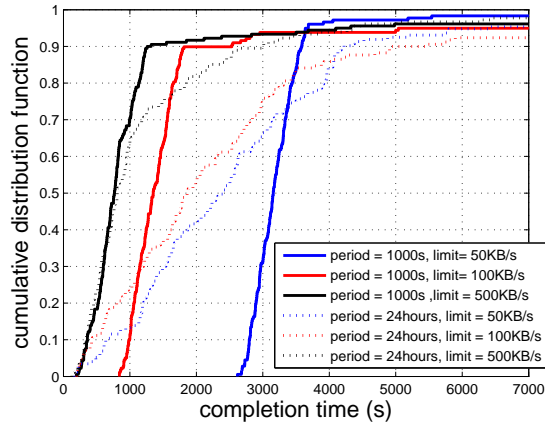


Fig. 8. Completion Time Distribution Varying Seed Upload Speed Limit

TABLE II
90%-TILE AVERAGE COMPLETION TIME VARYING SEED UPLOAD BANDWIDTH LIMIT

Update Interval	50KB/s	100KB/s	500KB/s
1000s	3157s	1399s	814s
24hours	2522s	2221s	1125s

As the fig.8 shows, the initial seed upload limitation of bandwidth significantly impacts on both two RSS update settings while for 24 hours it's much larger. However, as table.II shows, the 90-percentile average completion time varies little with a period of 24 hours. This means that an appropriate Broadcatching setting can make the whole system benefit more from the seeds.

We also varied the number of the initial seeds. The upload bandwidth of each seed was limited to 500KB/s, and three modes were compared: 1 seed, 2 seeds and 4 seeds.

TABLE III
90%-TILE AVERAGE COMPLETION TIME VARYING NUMBER OF SEEDS

Update Interval	1 seed	2 seeds	4 seeds
1000s	814s	968s	991s
24hours	1126s	1266s	1133s

Table.III gives out the 90-percentile average completion time of the six experiments. We can see that the changing of the number of the initial seeds has some influence on the downloading performance of the system with a period of 1000s while having little impact on that of 24 hours. That is because the capacity of the initial seeds is more critical when with a large population of online downloaders. Overall, seeding capacity influences significantly on Broadcatching mechanism.

IV. CONCLUSION AND FUTURE WORK

In this paper we conducted experimental research to evaluate the performance of Broadcatching mechanism in BT system. Based on the results of our experiments, we can draw the following conclusions.

1) Broadcatching can greatly improve the performance of the BT system. Through this mechanism, everyone tends to get the file much faster. The 90%-tile average completion time has been reduced by as much as 85%. In addition, the downloading failure ratio has also been slightly improved.

2) The bandwidth cost of the initial seed is reduced significantly. Under a rational RSS update interval like 1 hour, the seed upload ratio is rather small, mainly due to the increased share ratio of peers. According to our experiments, the average share ratio has been increased notably.

3) To get an optimum performance, the capacity of the seeds as well as RSS update interval are the most important factors to consider. The capacity of the seeds has more influence on a Broadcatching BT system than the normal one. Further more, under certain seeding environment, there exists an optimal value for RSS update interval.

Our work provides some insights for exploring the potential of such a peer-behavior-changing technique as Broadcatching. However, the analytical relationship between Broadcatching and the resulting performance is still an open problem. Our future work is to combine Broadcatching with BT models to perform an analytical study in order to guide the design of such systems.

ACKNOWLEDGMENT

This work is supported by the National Basic Research Program of China (No.2007CB310806), the National Science Foundation of China (No.60473087, No.60703052) and the National High Technology Development Program of China (No.2007AA010306). We also wish to thank Xiaohui Shi and Ting Zhang for their constant support and sound advice.

REFERENCES

- [1] M. Izal, G. Urvoy-Keller, E.W. Biersack, P.A. Felber, A. Al Hamra, and L. Garcés-Erice. Dissecting BitTorrent: Five Months in a Torrent's Lifetime In Passive and Active Measurements, Antibes Juan-les-Pins, France, April 2004.
- [2] Lei Guo, Songqing Chen, Zhen Xiao, Enhua Tan, Xiaoning Ding, and Xiaodong Zhang. Measurements, Analysis, and Modeling of BitTorrent-like Systems. In Internet Measurement Conference (IMC), Berkeley, CA, October 2005.
- [3] Matei Ripeanu, Miranda Mowbray, Nazareno Andrade, Aliandro Lima. Gifting Technologies: A BitTorrent Case Study. Technical Report HPL-2007-26, Enterprise Systems and Software Laboratory, HP Laboratories Bristol February 19, 2007.
- [4] Internet Study 2007. <http://www.ipoque.com>.
- [5] Giovanni Neglia, Giuseppe Reina, Honggang Zhang, Don Towsley, Arun Venkataramani, John Danaher. Availability in BitTorrent Systems. In INFOCOM, 2007.
- [6] The application/rss+xml Media Type: <http://www.rssboard.org/rss-mime-type-application.txt>. Network Working Group, May 22, 2006
- [7] Azureus homepage. <http://azureus.sourceforge.net>.
- [8] μ Torrent homepage. <http://www.utorrent.com>.
- [9] BitTornado homepage. <http://www.bittornado.com>.
- [10] Official BitTorrent homepage. <http://www.bittorrent.com>.
- [11] PlanetLab homepage. <http://www.planet-lab.org>.
- [12] Broadcatch Technologies. <http://www.broadcatch.com>.
- [13] Gillmor, Steve. BitTorrent and RSS Create Disruptive Revolution EWeek.com, December 13, 2003.
- [14] Raymond, Scott. Broadcatching with BitTorrent. Scotttraymond.net, December 16, 2003.
- [15] <http://en.wikipedia.org/wiki/broadcatching>.