

# Robustness of BitTorrent-like VoD protocols

Tamás Vinkó\*

## Abstract

Besides server supported solutions for Video-on-demand, approaches based on distributed systems such as BitTorrent are being used due to their efficiency and high scalability. There are several protocol variants proposed in the literature, which are mainly concerned with providing mechanisms for piece selection and peer selection. In this paper, using the concept of Design Space Analysis, we give comparisons of the performances of several BitTorrent-like Video-on-demand protocols under the assumption that other protocol variants may also enter the system.

**Keywords:** Design Space Analysis BitTorrent Video-on-demand Piece selection Peer selection

## 1 Introduction

Among the peer-to-peer (P2P) content sharing applications BitTorrent [6] is inevitably the most popular protocol. One of its key ideas is that a file, which is subject to be shared, is divided into small pieces and the participating peers are exchanging these pieces between each other. For traditional file sharing it is a good strategy to request the rarest first piece to be downloaded. That is indeed the default option in BitTorrent. However, in case of video-on-demand (VoD) systems, where users would like to watch or listen the media content (video or audio), the pieces of the file should be received in an in-order fashion and the system also needs some quality-of-service (QoS) requirements to be fulfilled, like smooth playback and quick startup. Contrasting this service with live streaming, different users in VoD systems are usually interested in different parts of the media content, which appears to be similar to the dynamics of users in traditional P2P file-sharing. Thus, recently, much attention has been paid to the VoD extensions of the BitTorrent protocol [5, 11, 13, 18, 19].

Two of the most important aspects of BitTorrent-like VoD approaches are piece selection and peer selection. As it has been shown by D'Acunto *et al.* [7], the proposed solutions to these two policies can lead to different performance and thus they should be chosen according to the given scenario. On the other hand, performance evaluations of the VoD protocols in a mixed swarm, where peers can use

---

\*University of Szeged, Hungary, E-mail: tvinko@inf.u-szeged.hu

different protocol variants have not yet been considered in the literature. To this end, in this paper we apply the approach of Design Space Analysis [16] and investigate the ability of VoD protocol variants outperforming other variants in a mixed population with respect to the two aforementioned QoS requirement measures. In this way, besides the Performance, which indicates the average efficiency of a given protocol in a homogeneous population, the Robustness and Aggressiveness of the considered protocols can also be given.

In the following we first give the necessary definitions of the Design Space Analysis and the BitTorrent-like VoD systems, see Section 2. Then we set up the design space of the VoD protocols by giving all the details of its dimensions and measures, see Section 3. In Section 4 we describe the simulation model and methodology applied in the paper. Finally, in Section 5 the simulation results are presented.

## 2 Background

In this section all the important definitions used later in the paper are introduced.

**Design Space Analysis.** In order to comprehensively model incentives in distributed protocols, a method called Design Space Analysis (DSA) was proposed by Rahman *et al.* [16]. This simulation based approach was inspired by Axelrod [4] modeling specific interactions in repeated games. The DSA framework combines the specification of a design space of protocols together with their analysis by means of extensive simulations. The specification part consists two steps: parametrization and actualization. Parametrization is the determination of the salient dimensions of the design space. In actualization one specifies the actual values of every individual dimension. Having done with these two steps, a solution concept can be used, in which any protocol of the design space can be characterized using different measures. For a given protocol Performance, Robustness and Aggressiveness measures (called PRA quantification) were proposed [16]. Using a predefined utility, a protocol's Performance is the average performance of the system when all the participating peers apply the same protocol variant. This predefined utility, which quantifies individual performance, is always domain specific. For example, in traditional P2P based content distribution it can be the peers download speed. Robustness and Aggressiveness indicates the ability of outperforming other protocol variants depending on the composition of the system, e.g. the protocol is used by a majority, respectively a minority, of the population. Using these three measures, provided that they are normalized into the interval  $[0, 1]$ , the properties of a given protocol can be characterized as a point in the three dimensional PRA space  $[0, 1]^3$ . System designers might want to introduce protocols which maximize on all dimensions. However, usually one needs to compromise between the three. As it was shown in [16], this is indeed the case regarding a large space of P2P file swarming systems.

**BitTorrent.** The most popular P2P based content distribution protocol is BitTorrent. In this system, the files are divided into small pieces, which allows the participating peers to share the parts of the file which they already obtained while still downloading the other pieces. Peer discovery is usually done with the help of a central component (tracker) that, upon requests, sends a list of nodes participating in the downloading (leeching) and uploading (seeding) of the same content. Each peer maintains connections to its neighbours and exchanges the pieces with the subset of them. Technically, each peer equally divides its upload capacity into upload slots of two types: regular unchoke slots and optimistic unchoke slots. Regular unchoke slots are assigned to peers that have recently provided data directly to the peer at their highest speeds. The assignments of these slots are re-evaluated in every unchoke interval  $\delta$ , usually  $\delta = 10$  seconds. Optimistic unchoke slots, on the other hand, are assigned to randomly selected peers. This allocation is also re-evaluated in every  $3\delta$  interval. This scheme can be seen as a hill-climbing type optimization method of finding out the peers from which the file can be obtained at the highest possible speed. At the same time it also helps to newcomers starting their download despite the fact that they have nothing to share at the beginning.

Each peer informs (and gets informed by) its neighbours about the pieces it owns (they own). Using this information, the request of pieces are done by local rarest first policy, e.g. each peer requests pieces which are the rarest among its neighbours. The outcome of this policy is that less available pieces of the file get replicated among the peers.

### 3 Setting up the design space

In this section, the application of DSA for BitTorrent-like VoD systems is described. First, we Parameterize the generic design space giving its dimensions. Next, based on this space, we Actualize a specific BitTorrent-like VoD design space. This is then followed by the application of the PRA quantification on this space.

#### 3.1 Parametrization

In our design space we specify two dimensions relevant to the VoD protocols:

- *Piece Selection* determines which piece of the video file should be selected for downloading by the peer. The role of this policy is to find a trade-off between in-order piece download (for QoS) and high enough bartering chances among peers (to guarantee efficient bandwidth utilization). An important parameter of this dimension is called *sequentiality parameter* to be tuned to favour one of these two aspects.
- *Peer selection* dictates how peers selects each other to upload data to. The role of this policy is to incentivize cooperation among peers. In BitTorrent-like systems it is usually designed to favour good uploaders.

### 3.2 Actualization

Having defined the two dimensions of our design space, the next step is to specify the actual values for every individual dimensions.

Three different policies were investigated for *Piece selection*:

- *Window-based* piece selection (WIN) employs a sliding window of fixed size within which pieces are chosen to download [15]. Depending on the sequential download progress of the peer the window advances from the start to the end of the media file. Within this sliding window BitTorrent’s standard rarest-first piece selection is applied. The size of the window, denoted by  $w$ , is the sequentiality parameter and naturally has effect on the piece diversity among the peers.
- *Probabilistic* piece selection (PROB) chooses pieces in relation to a fixed probability distribution, usually giving higher chances to first pieces not yet downloaded. In our simulations we employed the Zipf probability distribution as proposed in [12], which has the form  $P(k) \sim (k + 1 - k_0)^{-\theta}$ , where  $k_0$  is the index of the first piece that the peer has not yet downloaded. In this case the parameter  $\theta$  represents the sequentiality parameter.
- *Priority-based* piece selection (PRIO) gives priority to pieces which are close to be played. This policy, as it was defined in [9], uses a parameter  $h$  which defines the size of the priority set (and thus it is the sequentiality parameter of this policy) in the following way. Assume that the peer’s current playback position is  $t$ . The peer requests the piece  $k$  on the first match in the following list of sets of pieces (called priority sets):
  - high priority:  $t \leq k < t + h$  in-order piece selection if the local peer has already started playback, rarest first otherwise;
  - mid priority:  $t + h \leq k < t + 5h$ : with rarest first piece selection;
  - low priority:  $t + 5h \leq k$ : with rarest first piece selection.

In the dimension of *Peer selection* we studied the following three policies:

- *Direct reciprocity* (D), which is the standard peer selection policy of BitTorrent. Using this policy a peer uploads to other peers that have recently uploaded to it at the highest rate. This decision is taken based on local information only, that is, the peers themselves are measuring the upload speed of the other peers.
- *Indirect reciprocity* (I) prescribes that a peer uploads to other nodes that have recently forwarded data to others at the highest rate. We are using the Give-to-Get protocol introduced in [9], in which a peer  $p$  discovers the forwarding rate of a child node  $q$  by periodically asking its grandchildren about the pieces received from  $q$ . In order to apply this selection mechanism, each peer needs to gather information from other nodes, which is more costly than the direct reciprocity.

- *Random (R)*, in case the peers do not follow any incentive scheme, they select peers to upload to in a random fashion.

### 3.3 Quantification

In order to apply the DSA approach we need to define the quantification measures for Performance, Robustness and Aggressiveness. In traditional file sharing scenario this measure can simply be the downloading rate. However, in the VoD context one should use much more informative measures. We are thus using the following two measures:

- *Continuity index*, defined as the ratio of pieces received before their deadline over the total number of pieces [17], and
- *Startup delay*, defined as the time a user has to wait until the playback can be started.

As it was argued in [7] the smooth playback continuity is of higher importance in a VoD system regarding QoS. Especially, when the bandwidth is insufficient to meet a demand, it is more important to serve those peers that have already started playing. Only when there is enough bandwidth available, the startup delay should be the goal to be minimized. Habib *et al.* [8] demonstrated that when the continuity index of a stream is 95%, a users overall perceived video quality is very good.

## 4 Simulation Model and Methodology

Our simulation model is based on the MSR BitTorrent simulator [3], which was extended by Lucia D'Acunto with the goal of supporting VoD simulations including the piece selection policies and peer selection policies listed in Section 3. For further details regarding the extension see [7]. The original simulator includes most of the elements of a BitTorrent swarm from the overlay level down to the piece exchange of the peers. The original version of the simulator was used, and extended accordingly, in other P2P VoD studies, e.g. [19, 20].

We simulated a VoD swarm in which the service provider consists of one original seeder who is always online and has upload capacity  $U_s$ . Peers are entering the swarm based on a prescribed arrival rate  $\lambda = 0.05$  peers/s, and leave the system as soon as they finish with the downloading. The media file, which is divided into  $n$  pieces of identical size, has playback rate  $R$  and size  $F$ . The average upload capacity of the participating peers is  $\mu$ , and as usual in analysis of BitTorrent-like systems, we assume that the download capacity of peers is infinite. As it was laid out in [7], the required server bandwidth  $U_s$  can be calculated as  $U_s = (1 - \frac{\mu}{\gamma R})\lambda F$ , hence, given the parameters of the system, one only needs to set up the parameter  $\gamma$  that suits the needs of the participating peers. For better comparisons all the parameters used in our subsequent simulations are the same as it was in [7]. These are listed in Table 1.

Table 1: Simulation Settings

| Parameter               | Value                                      |
|-------------------------|--|
| Video playback rate $R$ | 800 kb/s                                   |
| Video length $L$        | 50 min                                     |
| Simulation time         | between 250 and 750 min ( $5L$ and $15L$ ) |
| Piece size $P_S$        | 256 kB                                     |
| Priority set size $h$   | 25 (direct) / 20 (indirect)                |
| $\theta$                | 2 (both direct and indirect)               |
| Window size $w$         | 40 (direct) / 30 (indirect)                |
| Initial buffer $B$      | 20 pieces (PROB) / $w$ (WIN) / $h$ (PRIO)  |
| Upload rate $\mu$       | 1000 kb/s ( $1.25R$ )                      |
| $\gamma$                | 1.3  |

In order to conduct the Robustness and Aggressiveness tests of DSA, further extensions of the simulator were needed. Specifically, the possibility of using two or more different strategies in a swarm was not possible in general. In our current extension the peer- and piece selection policies are private to the peers. Thus it is possible to set up experiments in which a swarm is composed by peers using different protocol variants at the same time.

Technically, the Robustness test was done in the following way. Each protocol variant is pitted against every other variant. The competition in which two different protocol variants (say  $p_1$  and  $p_2$ ) are playing against each other is called *encounter*. In each encounter peers are arriving to the swarm with a prescribed arriving rate (which was 0.05 peers/s in all the simulations). Upon arrival a peer is associated to protocol  $p_1$  or  $p_2$  with equal probability. For each encounter we made ten runs. For each run, we compare the average performance (measured either by Continuity index or by Startup delay) of protocol  $p_1$  with the average performance of the other protocol. If the performance of  $p_1$  is better than the performance of the other protocol, we mark it as a *win* for  $p_1$ , otherwise we mark it as a *loss* for  $p_1$ . The Robustness value for a protocol is calculated by the number of encounters it wins against all opponents in all encounters divided by the total number of encounters that it plays, which is a constant for all protocol variants. Concerning Aggressiveness the same kind of scheme is applied with the difference that upon arrival into the swarm, a peer is associated to protocol  $p_1$  with probability equal to 0.1 and to protocol  $p_2$  with probability equal to 0.9.

## 5 Results

**Robustness.** The results of the robustness test of the different VoD protocol variants are shown in Figure 1. It is important to note that the axes represent the applied measures in a normalized manner. While in the original definition of Startup delay, the lower is the better, here higher values mean more robust variant.

By its original definition the value of the Continuity index lies in the interval  $[0, 1]$ . This also holds here, however, the meaning of the actual value is different.

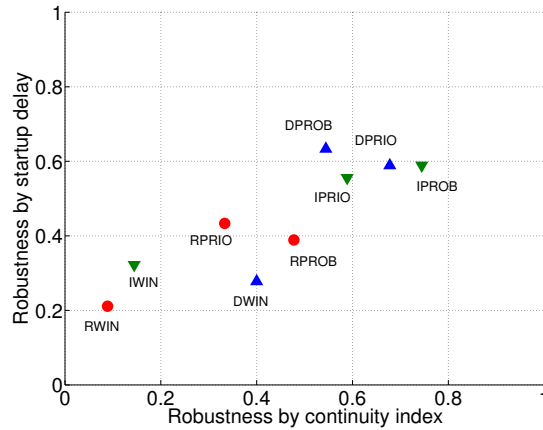


Figure 1: Robustness scatter plot

It can be clearly seen that the robustness values of the different variants are highly correlated in the two applied measures. The Pearson's linear correlation coefficient is 0.8568. In protocol-homogeneous swarms, where each peer applies the same protocol it was shown that shorter startup delay usually leads to lower continuity index [7]. What we can see here is that this does not hold in a mixed environment. The most robust variants are using PROB piece selection together with direct (for better startup) or indirect (for better continuity) reciprocity. At the lower end, random peer selection together with WIN is the worst choice. In general we can also conclude that the window-based piece selection is not robust at all. Applying random peer selection policy is also proven to be a bad choice.

In order to see more details on the actual results of the encounters using the two different measures heatmaps are shown on Figures 2 and 3. Here, lighter square means more frequent winnings against the other protocol. Regarding continuity index, it can be clearly seen that the protocol variants using Direct or Indirect reciprocity are dominating the others. This is indicated by the light upper right  $5 \times 5$  corner. From here, we can see that D and I type protocols outperform all the R types and IWIN (with the exception of DWIN being outperformed by RPROB). Especially, as we have already seen, the row of IPROB is the lightest one, that is the most robust variant. This finding emphasizes the advantage of the usage of non-random incentive mechanism in P2P VoD systems. Regarding IWIN, it is interesting to remark that, according to [7] it performs well in protocol-homogeneous swarm as well as in bandwidth-heterogeneous swarm.

Using startup delay as measure the corresponding heatmap is shown on Figure 3. Due to the already noticed correlation with the other measure, we have similar picture here, although the squares are bit darker in general (according to Figure 1

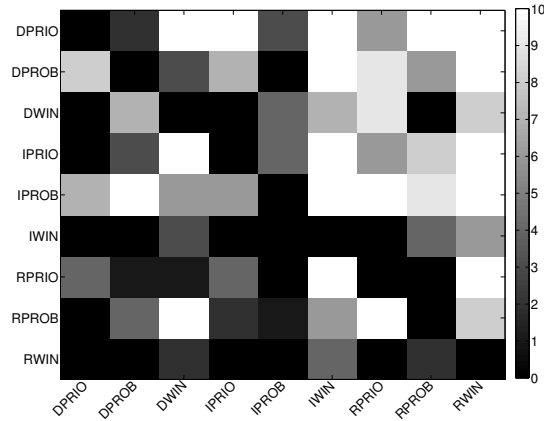


Figure 2: Heatmap of encounters with continuity index measure

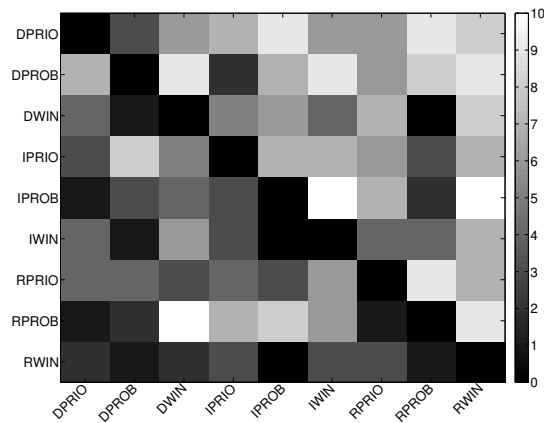


Figure 3: Heatmap of encounters with startup delay measure

the most robust protocol w.r.t. startup delay has robustness value 0.62).

**Robustness versus Performance.** As it was already mentioned the Performance test of the considered VoD protocol variants has already been done in [7]. On Figures 4 and 5 we can see comparisons of Robustness against Performance using the two different measures. We have to emphasize again that the values in these figures are normalized. This normalization was already explained about Robustness. As for Performance, the protocol having the highest continuity index (or



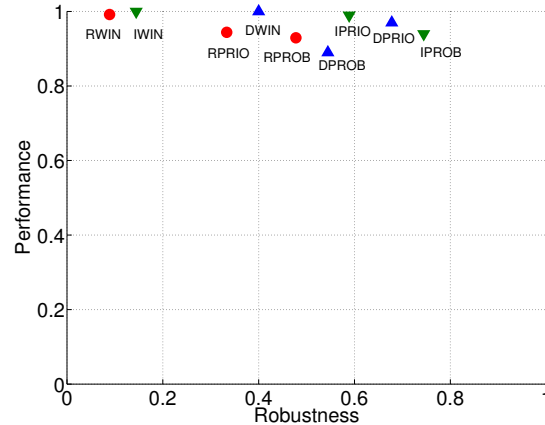


Figure 4: Robustness against Performance for continuity index

the lowest startup delay) has value 1 and all the other protocols values are relative to this.

Regarding continuity index we obtained no correlations between Performance and Robustness (Figure 4) due to the simple fact that all the tested protocols have very high continuity index. This is not the case with the startup delay measure, see Figure 5. Although we did not get significant correlation, the Performance values show greater variety in this case. From these two figures we can conclude that DPRIOR is the best choice to achieve the best Performance and Robustness together. This is in line with the experiments done in [7], where the same protocols were pitted against peers using the traditional BitTorrent (e.g. file transfer instead of media streaming) and resulted in the same conclusion.

**Robustness versus Aggressiveness.** Now we turn our attention to the results of the Aggressiveness tests. For the continuity index the results are shown on Figure 6. The Pearson’s linear correlation coefficient is 0.4553, thus, as it can be noticed also from the figure, the two measures are not correlated in general. The best choice in this case is the DPRIOR protocol variant. In case of PROB piece selection there is a trade-off between Robustness (combined with indirect reciprocity) and Aggressiveness (combined with direct reciprocity). Note that protocols applying Indirect reciprocity resulted in very low Aggressiveness values. In this situation, being in the minority, the peers using indirect reciprocity are struggling with finding honest peers, about which they cannot obtain information from peers using direct reciprocity (who are in the majority). We can also see that Random peer selection is not a good choice.

Regarding startup delay we get slightly different results, see Figure 7. In general, there is no correlation (correlation coefficient is 0.3376) between the two measures. In this case the DPROB variant is the best choice for maximizing the two measures.

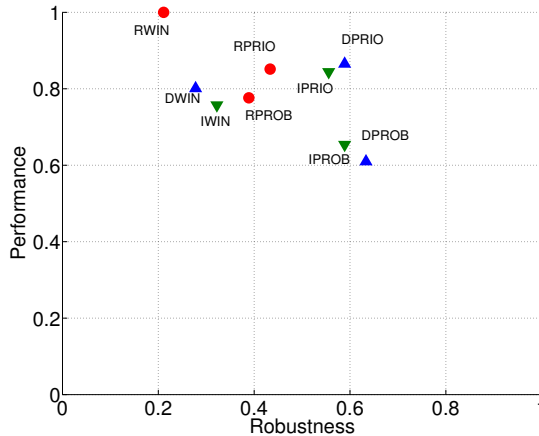


Figure 5: Robustness against Performance for startup delay

## 6 Related work

The theoretical and practical (both simulation-based and measurement-based) investigation of BitTorrent-based VoD approaches have attracted the attention of many researchers in the recent years. In particular, considerable effort has been invested in designing and evaluating the piece selection (e.g. [1, 2, 5, 14]) and peer selection (e.g. [10, 15, 20]) policies of these systems. Perhaps the most related works to ours are the following two. Xu *et al.* [18] investigated the performance of different piece- and peer selection policies, as well as solutions to replica management. However, they did not take mixed swarms into consideration. Ma *et al.* [11] studied the performance of some piece selection policies for VoD in the presence of traditional BitTorrent peers (independently, similar simulation results can also be found in [7]). Our present work provides insights on the performance of some of the VoD approaches in mixed environment, which situation is very typical in open distributed systems.

## 7 Conclusion

We have demonstrated, by means of simulations, that performance of BitTorrent-like VoD approaches can show high variance depending on the protocol composition of the swarm they participate in. Following the terminologies of Design Space Analysis [16], robustness of a given protocol indicates the ability of outperforming the other protocols also existing in the same system. Our results show the importance of built-in incentive mechanisms (contrasted with random peer selection), and that probabilistic piece selection policy provides good robustness value coupled either with direct or indirect reciprocity. However, when a protocol is in the minority,

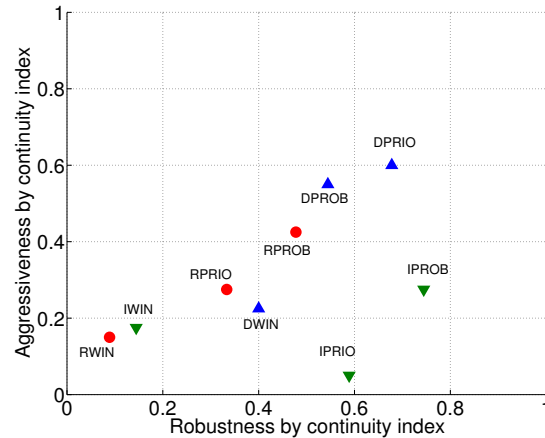


Figure 6: Robustness against Aggressiveness measured by continuity index

direct reciprocity with priority based piece selection (for smooth continuity) or with probabilistic piece selection (for shorter startup delay) should be used.

## Acknowledgements

This work was partially supported by the European Union and the European Social Fund through project FuturICT.hu (grant no.: TAMOP-4.2.2.C-11/1/KONV-2012-0013) and by the Bolyai Scholarship of the Hungarian Academy of Sciences.

## References

- [1] A. Vlavianos, M. Iliofotou and M. Faloutsos: BiToS: Enhancing BitTorrent for Supporting Streaming Applications. In: Proceedings of IEEE Global Internet Symposium, pp. 1–6, 2006.
- [2] Annapureddy, S., Guha, S., Gkantsidis, C., Gunawardena, D., Rodriguez, P.R.: Is high-quality vod feasible using p2p swarming? In: Proceedings of the 16th international conference on World Wide Web, pp. 903–912, 2007.
- [3] A.R. Bharambe, C. Herley and V.N. Padmanabhan: Analyzing and Improving a BitTorrent Network’s Performance Mechanisms. In: Proceedings of IEEE INFOCOM, pp. 1–12, 2006.
- [4] Axelrod, R.: The Evolution of Cooperation. Basic Books, New York, 1984.

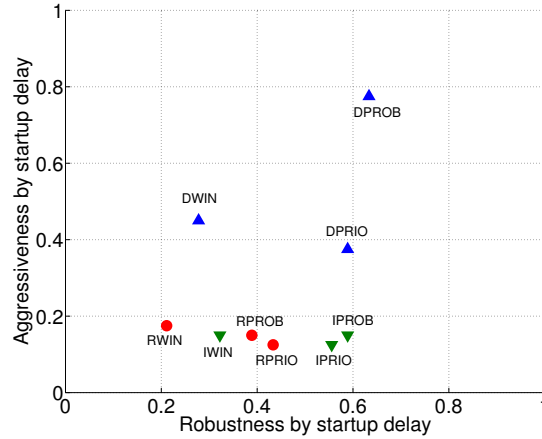


Figure 7: Robustness against Aggressiveness measured by startup delay

- [5] Choe, Y.R., Schuff, D.L., Dyaberi, J.M., Pai, V.S.: Improving vod server efficiency with bittorrent. In: Proceedings of the 15th International Conference on Multimedia, pp. 117–126, 2007.
- [6] Cohen, B.: Incentives Build Robustness in BitTorrent. In: Workshop on Economics of Peer-to-Peer Systems. pp. 68–72, 2003.
- [7] D’Acunto, L., Chiluka, N., Vinkó, T., Sips, H.: Bittorrent-like P2P approaches for VoD: A comparative study. *Computer Networks* **57**(5), 1253 – 1276, 2013.
- [8] Habib, A., Chuang, J.: Service Differentiated Peer Selection: An Incentive Mechanism for Peer-to-Peer Media streaming. *IEEE Transactions on Multimedia* **8**, 610–621, 2006.
- [9] J.J.D. Mol, J. A. Pouwelse, M. Meulpolder, D.H.J. Epema and H.J. Sips: Give-to-Get: Free-riding-resilient Video-on-Demand in P2P Systems. In: Proceedings of SPIE MMCN, 2008.
- [10] L. D’Acunto, N. Andrade, J.A. Pouwelse and H.J. Sips: Peer Selection Strategies for Improved QoS in Heterogeneous BitTorrent-like VoD Systems. In: Proceedings of the IEEE International Symposium on Multimedia, 89–96, 2010.
- [11] Ma, Z., Xu, K., Liu, J., Wang, H.: Measurement, modeling and enhancement of bittorrent-based vod system. *Computer Networks* **56**(3), 1103 – 1117, 2012.
- [12] N. Carlsson, D. L. Eager and A. Mahanti: Peer-assisted on-demand Video Streaming with Selfish Peers. In: Proceedings of IFIP NETWORKING, 586–599, 2009.

- [13] N. Parvez, and C. Williamson and A. Mahanti and R. Carlsson: Analysis of BitTorrent-like Protocols for On-Demand Stored Media Streaming. In: Proceedings of ACM SIGMETRICS, 301–312, 2008.
- [14] P. Savolainen, N. Raatikainen and S. Tarkoma: Windowing BitTorrent for Video-on-Demand: Not All is Lost with Tit-for-Tat. In: Proceedings of IEEE GLOBECOM, pp. 1–6, 2008.
- [15] P. Shah and J. F. Pris: Peer-to-Peer Multimedia Streaming using BitTorrent. In: Proceedings of IEEE IPCCC, pp. 340 - 347, 2007.
- [16] Rahman, R., Vinkó, T., Hales, D., Pouwelse, J., Sips, H.: Design space analysis for modeling incentives in distributed systems. In: Proceedings of ACM SIGCOMM, pp. 182–193, 2011.
- [17] X. Zhang, J. Liu, B. Li, and T.S. P. Yum: DONet/CoolStreaming: A Data-driven Overlay Network for Live Media Streaming. In: Proceedings of IEEE INFOCOM, pp. 2102–2111, 2005.
- [18] Xu, K., Liu, X., Ma, Z., Zhong, Y., Chen, W.: Exploring the policy selection of the p2p vod system: A simulation-based research. *Peer-to-Peer Networking and Applications* 8(3), 459–473, 2014.
- [19] Y. Borghol, S. Ardon, N. Carlsson and A. Mahanti: Toward Efficient On-Demand Streaming with BitTorrent. In: Proceedings of IFIP Networking, pp. 53–66, 2010.
- [20] Yang, Y., Chow, A., Golubchik, L., Bragg, D.: Improving QoS in BitTorrent-like VoD systems. In: Proceedings of IEEE INFOCOM, pp. 1–9, 2010.

*Received ...*