

Improving Content Availability for BitTorrent-like File Sharing employing User-driven Virtualized Nodes

SuHyun An

School of Information &
Communications, GIST, Gwangju,
Korea
shan@nm.gist.ac.kr

HyunYong Lee

National Institute of Information and
Communications Technology (NICT),
Tokyo, Japan
ifjesus7@gmail.com

JongWon Kim

School of Information &
Communications, GIST, Gwangju,
Korea
jongwon@nm.gist.ac.kr

Abstract—Even though BitTorrent-like file sharing is successful for P2P (peer-to-peer) content-sharing over Internet, it suffers from the content unavailability problem when the contents are not popular (i.e., the absence of seed). This content unavailability can be mitigated by using multi-swarm cooperation and/or by bundling popular/unpopular contents. In this paper, by employing user-driven virtualized nodes, we propose a content availability improvement method in a BitTorrent-like file-sharing system. The virtualized node is assigned to (and driven by) the user node so that it can help the user node to accelerate the P2P download by taking advantage of the swarm it belongs to. We evaluate the performance of proposed method by simulating a practical BitTorrent file sharing scenario and by comparing the download completion time.

Keywords: *User-driven virtualized node, network and system virtualization, BitTorrent file-sharing, and network research testbeds.*

I. INTRODUCTION

In the Peer-To-Peer (P2P) systems nodes rely on connections with other nodes instead of a server. P2P file sharing is useful for various reasons such as scalability and server's overload. We do not worry about the resource deficiency of the server because the P2P system takes advantage of resources of participating nodes. P2P file sharing based on Kulbak, Bickson, Kazza, and BitTorrent (in short BT) is popular [1, 2, 3]. Among them, BT has been broadly used for its efficiency and fairness [1, 2]. It originates with both tit-for-tat (TFT) node selection and rarest-first chunk selection. Give-and-take of TFT node selection leads to active participation of nodes and reduction of free riders. Also, by taking rarest-first criterion, nodes select the rarest chunk(s) as next downloading chunk(s) so that we can speed up replications of rare chunks.

In the BT-like file sharing, a swarm is a set of nodes sharing the same content. If a node becomes a *seed* (i.e., a node with entire content), it does not need to stay any longer and quite often it leaves the swarm. Thus, if the content is not popular, the entire chunk set may not exist in the swarm (i.e.,

content unavailability problem) and *leechers* (i.e., normal nodes) cannot complete the content download [5, 11, 13].

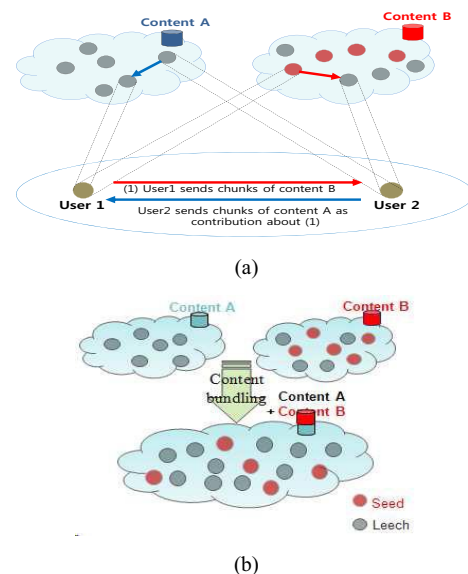


Figure 1. Content availability improvements: (a) multi-swarm cooperation and (b) bundling of related content chunks.

In order to solve this content unavailability problem, several ideas such as multi-swarm cooperation and/or bundling popular/unpopular contents are proposed so far [4, 6]. In Figure 1(a), the multi-swarm cooperation is illustrated, where the unavailable chunks of a swarm is obtained from other cooperating swarm(s). Note that the participating nodes are still subject to give-and-take content exchange [4]. Also, in Figure 1(b), we illustrate the bundling of related content chunks. The content provider bundles a bunch of related content chunks including the chunks of popular content. The downloading nodes then utilizes the relatively long (due to its popularity) seeding time of seed(s) [6]. However, all the above ideas are

suffering from the dependency on the volunteering cooperation of participating nodes.

In this paper, by employing user-driven virtualized nodes, we propose a content availability improvement method in BT-like file-sharing systems. We are assuming a deployment scenario where the content provider contributes physical nodes that can support computing/network virtualization [7]. The contributed nodes are placed at selected geographical locations for efficient content distribution¹. The virtualized node is assigned to (and driven by) the user node and it can help the user node to accelerate the P2P download by taking advantage of the swarm it belongs to. After detecting the content unavailability, a user node obtains help from the virtualized node. Then the user and virtualized nodes together download chunks within their swarms so that the downloaded chunks are not overlapped. We evaluate the performance of proposed method by simulating a practical BT file sharing scenario and by comparing the download completion time.

The rest of this paper is organized as follows. In Section II, we introduce the computing/network virtualization and the content unavailability problem, and then the content availability improvement method using the user-driven virtualized node. In Section III, we introduce the simulation environment setup incorporating virtualization and discuss simulation results. Finally, Section IV concludes this paper.

II. CONTENT AVAILABILITY EFFICIENCY IMPROVEMENT THROUGH USER-DRIVEN VIRTUALIZED NODE

A. Utilization of Virtualization Techniques

Virtualization techniques make a user or service feels like to use the full resource alone, even though physical node(s) are actually used by many users or services simultaneously [7]. Network virtualization provides the abstraction of networking resources and computing virtualization provides the abstraction of computing resources. Network virtualization has been propounded as the future inter-networking paradigm because network virtualization provides flexibility, promotes diversity, and promises security, and increased manageability [7, 12].

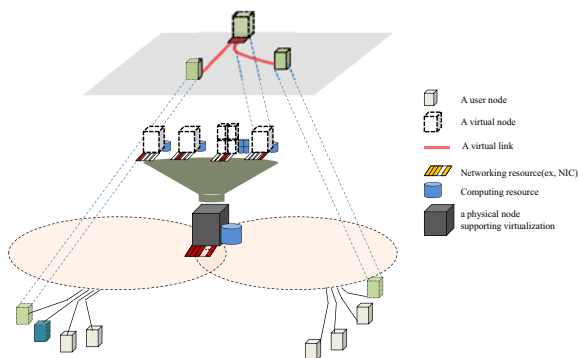


Figure 2. The concept of user-driven virtualized node.

¹ We also assume that, by taking advantage of network virtualization, virtual links are created among the user node and user-driven virtualized node to support the guaranteed networking performance.

In Figure 2, a physical node supporting virtualization is considered as a resource pool and the virtual nodes are created from the physical node. User nodes within the coverage of the physical node supporting virtualization can utilize the virtualized physical node. The user nodes have direct connections with the user-driven virtualized node which are created from the physical node supporting virtualization. Like this, if a user can use the virtualized resources for one's own performance improvement, the range of utilization is very wide. For example, if a user or service has a lack of resources, it is possible to overcome the performance problem by utilizing the virtualized resource.

B. Content Unavailability Problem

In the previous section, we already introduced the content unavailability problem of BT-like P2P file sharing. Because the popularity of the content means the number of nodes sharing content in a swarm, the low popularity leads to low chunk replication. When there is no seed, the occurring possibility of content unavailability problem is increased. That is, content unavailability happens when other nodes (i.e., leechers) cannot download the entire content and wait continuously because a set of chunks does not exist in the swarm.

In the BT system, a user node access to a website and knows the address of a tracker through .torrent file received from the website. The user node accesses the tracker by using the tracker information in .torrent file and gets the node list sharing the same content. The user node randomly selects the neighbor (default 50) nodes and makes a two-way TCP connection with the selected nodes through handshake messages. They exchange BMs (BitMaps) that indicate chunks each node has.

- Initialization: Contact the tracker and obtain the node list. Also, establish connections with neighbor node(s).
- Perform choking algorithm to select nodes.
- Perform the rarest-first chunk selection and download chunks.

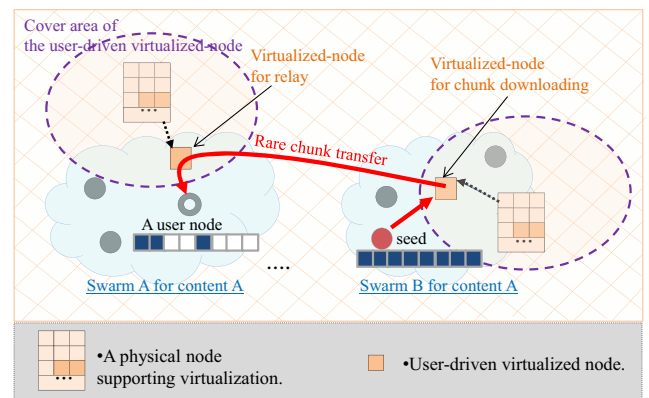


Figure 3. Improving the content availability through the user-driven virtualized node.

We suggest a content availability improvement method for a user node by using user-driven virtualized node. The user

node periodically checks its own content availability in order to justify whether it may receive the entire content in the swarm. If the user node detects the content unavailability problem, it lets the virtualized node join another swarm sharing the same content (through another tracker) and then the virtualized node downloads the unavailable chunks for the user node.

C. Improving the Content Availability with User-driven Virtualized Node

In the following, in three major steps, we list the basic procedures for our proposed approach.

Step 1: Check the content availability periodically: Periodically (e.g., 10 sec), by exchanging BMs with neighbor nodes, the user node attempts to estimate the content availability BMs of itself from the swarm it belongs. The estimated content availability BMs measure indicates how much safe for the user node to download all the content chunks. During the content availability measurement period, if the user node does not get any new chunk from neighbor nodes and the value of the content availability (i.e., the ratio of received chunks to the entire content) is less than '1', the user node decide that the available chunks are not sufficient in the swarm. However, this periodical measurement has a difficulty in finding the accurate time when the content unavailability problem arises due to the absence of seed(s). Note that the chunks of seed are distributed and shared among the nodes even after the seed node leaves the swarm. We therefore suffer from some delay in detecting the content unavailability problem. If we have sufficient amount of affordable resources, one practical best choice to use the virtualized node immediately after the user node joins in the swarm. Like this, the best time when the virtualized node is put into use may be decided according to the resource allowance.

Step 2: Activate the user-driven virtualized node: After gaining information of other swarms sharing the same content from other tracker(s), the user node activates the virtualized node and then let it join in another selected swarm. The virtualized node also establishes a communication connection with the user node by leveraging (and partially modifying) the handshake messages of the BT file-download system. The virtualized node prepares downloading those chunks indicated by (or communicated with) the user node.

Step 3: Downloading rare chunks and transferring the received chunks: By exchanging messages, the virtualized node gives priority to the rare chunks that the user node does not have, and downloads them. Immediately after downloading them, the virtualized node transfers the chunks to the user node. In this step, we should prevent the unnecessary duplicated downloads.

Step 3-1: Messaging between the virtualized and user nodes for chunk selection: The virtualized node initiates the chunk list by getting messages about current bitfield and selected chunks from the user node. The selected chunk message indicates chunks which the user node at the moment starts to download. To avoid duplicated downloads, as soon as the user and virtualized nodes select chunk(s), they immediately send selected chunk messages to each other. Figure 4 shows the related messaging sequence

between the user node and virtualized nodes. The bold arrows are general BT messages. The dotted arrows shows when selected chunk messages are exchanged.

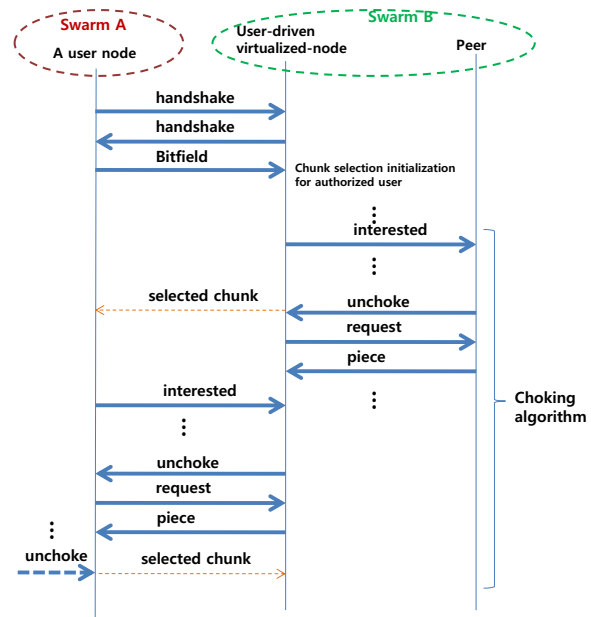


Figure 4. message sequence between the user and virtualized nodes.

Step 3-2: Downloading and transferring the selected chunks: Following the detailed procedures depicted in Figure 5, the virtualized node decides to download the selected rarest chunk(s) of the user node. Also, as mentioned earlier, the download chunks are immediately transferred from the virtualized node to the user node.

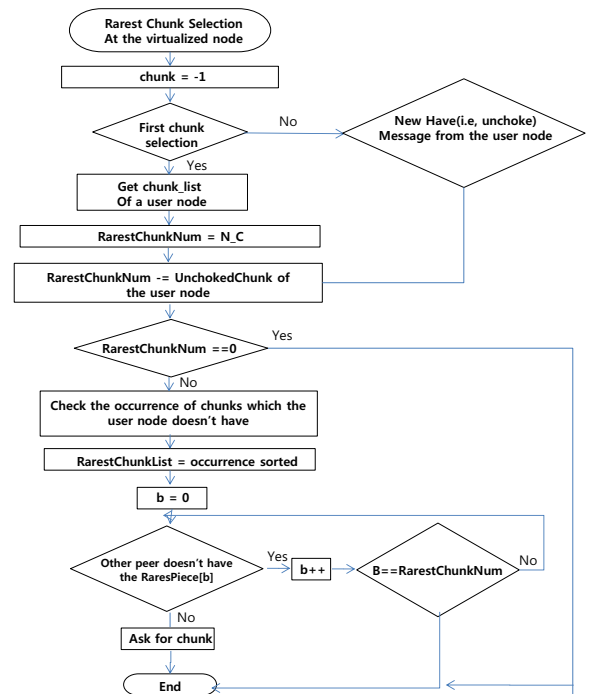


Figure 5. Rarest-first chunk selection of the virtualized node.

D. Details of the Chunk Selection Procedure

Let's look into the details of chunk selection at the virtualized node, illustrated in Figure 5. When the chunk is first selected, the virtualized node gets the chunk list from the user node, and then the *RarestPieceNum* parameter is initialized to the total number of chunks, N_C . The unchoked chunk number (*UnchokedChunk* in Figure 5) of the user node is subtracted from *RarestChunkNum* to extract chunks already in the possession of the user node. At each chunk selection, after checking new 'selected chunk message' from the user node, the number of selected chunk(s) is also subtracted from *RarestChunkNum*. Next, *RarestChunkNum*=0 means the completion of entire chunk download. If it is not 0, we need to check the rarity of every chunk. We then sort the occurrence of chunk list to select the rarest chunk (i.e., index 'b' of sorted array set to '0'). Next, if other nodes do not have *RarestChunk[b]*, the next rarest chunk ('b++') is selected. This process is repeated until we find the available chunk(s) in the swarm.

In the BT-like file-sharing system, the user node usually has a single 'optimistic unchoke' to which it uploads regardless of the current download rate from it. This optimistic unchoking discovers currently unused connections better than the ones being used and to provide minimal service to new nodes. However, the virtualized node uses 'authorized unchoking' instead of optimistic unchoking (default every 10 seconds). The authorized unchoking interval can be changed according to the chosen BT-like file-sharing system. The authorized unchoking is that the virtualized node certainly selects the user node as an interested node every choking time. It provides the user node with a steady contribution. It means that the virtualized node is completely committed to the user node by joining another swarm sharing the same content.

III. SIMULATION SETUP AND RESULTS

We evaluate the performance of proposed method by simulating a practical BT-like file sharing scenario and by comparing the download completion time.

A. Simulation Environment incorporating Virtualization

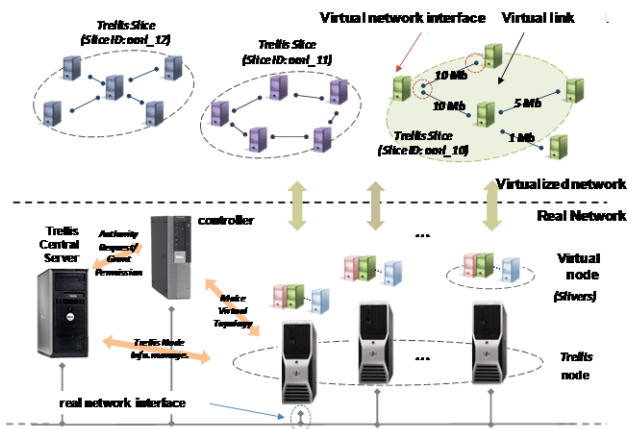


Figure 6. Conceptual illustration about network virtualization testbed.

The simulation should reflect the virtualization level of current network research testbeds. In this paper, we consider the computing/networking virtualization based on 'VINI testbeds supporting Trellis', as shown in Figure 6 [9, 10]. VINI virtualization can support simultaneous experiments with arbitrary network topologies on a shared physical infrastructure.

TABLE I. SIMULATION PARAMETERS.

Field	Value/unit
Number of nodes	100
File size	100 segments
Segment size	262Kbyte
Delay[min/max]	1 msec : 50 msec
Upload capacity in bytes	500 Kbps
Arrival time	[0-10] sec

To configure the virtualized node for simulation, we assume the virtualized node having low delay and high bandwidth connection to the user node. We also simulate the single 26Mbyte content sharing, split into 100 chunks. One key variable, arrival time, is the time that a node joins in the swarm. If the arrival time of all nodes is close, it could cause the flash crowd problem. Simulation parameters are depicted in Table I.

B. An Improvement Scenario for Content Availability

We consider the situation depicted in Figure 3. A user node joins in Swarm #1 for downloading content. In Swarm #1, there are one seed and ninety-nine leeches. After seeding time during 350 seconds, the seed node leaves Swarm #1. The user node checks the content unavailability periodically, and then if the user node decides to generate the content unavailability problem, the user node finds Swarm #2 of sharing the same content of Swarm #1. Then the user node lets the virtualized node to join the Swarm #2. The simulation setting and field setting of Swarm #2 is also the same as the TABLE I of Swarm #1. The difference is only in the long seeding time of a seed in Swarm #2.

C. Simulation Results

We show the efficiency of suggested method by comparing three cases such as 'original BT system', 'the virtualized node using the BT system' (Scheme #1) and 'the virtualized node using chunk selection and message sequence' in Section II (Scheme #2). In the simulation, the user node makes a decision to generate the content unavailability problem at about 420 seconds and at that time the virtualized node joins Swarm #2. In the case of original BT system, the user node and Swarm #1 finishes downloading approximately 77% of the entire content and then the user node does not get any chunk because there is no seed. Because the case of original BT system is obvious, we only show the results of the virtualized-node schemes. As a performance indicator, we use the download completion (%) at the user node and all nodes in Swarm #1 which the user node is joining in. We can check the suggested method efficiency from Figure 7 and Figure 8. The virtualized node joins in Swarm #2 at about 420 seconds. The dotted line means Scheme #1 and the straight line means Scheme #2. In Figure 7, even though the case of Scheme #1 downloads the content a little longer

than the case of Scheme #2 at the time which the virtualized node joins, Scheme #2 shows better download completion than Scheme #1. As a result, Swarm #1 finishes downloading the entire content ahead of about 20 seconds as shown in Figure 8.

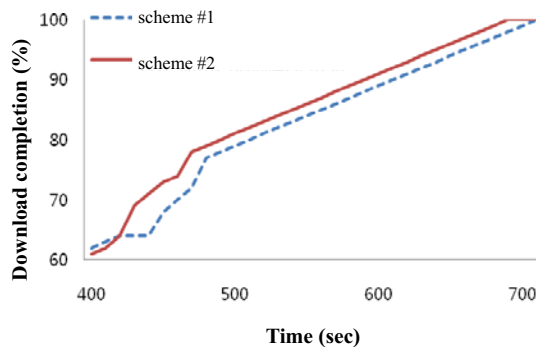


Figure 7. Download completion of the user node: Scheme #1 (the virtualized node using chunk selection for the user node) and Scheme #2 (the virtualized node using chunk selection of the BT system).

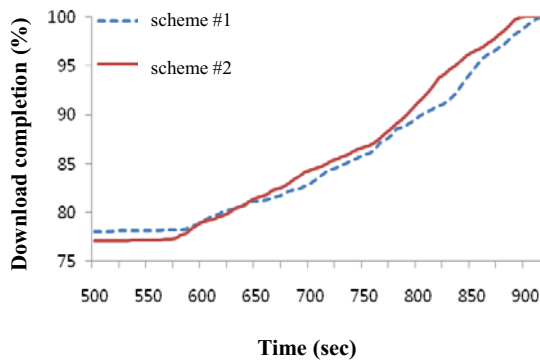


Figure 8. Download completion of the Swarm #1 which the user node is joining in: Scheme #1 (the virtualized node using chunk selection for the user node) and Scheme #2 (the virtualized node using chunk selection of the BT system).

IV. CONCLUSION

In this paper, we introduce the content unavailability problem of P2P and propose a content availability improvement method through the user-driven virtualized node. We increment the upload capacity which the user node is received from the other nodes by using the user-driven virtualized node. In future, we will research other performance improvements in the BT-like file-sharing systems through the user-driven virtualized node.

ACKNOWLEDGMENT

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2010-(C1090-1011-0004)).

REFERENCES

- [1] BitTorrent Specifications. <http://wiki.theory.org/BitTorrentSpecification>.
- [2] B. Cohen, "Incentives build robustness in BitTorrent," in *Proc. of Workshop on Economics of Peer-to-Peer Systems*, 2003.
- [3] E. Costa-Montenegro, et. al., "Implementation and analysis of the BitTorrent protocol with a multi-agent model," *Journal of Network Computing Applications*, 2010.
- [4] Y. Yang, et. al., "Multi-torrent: A performance study," in *Proc. of MASCOTS*, 2008.
- [5] S. Kaune, et. al., "Unraveling BitTorrent's file unavailability: Measurements, analysis and solution exploration," in *Proc. of ACM*, 2009.
- [6] D.S. Menasche, et. al., "Content availability and bundling in swarming systems," in *Proc. of ACM CoNEXT*, 2009.
- [7] N.M. Chowdhury and R. Boutaba, "A survey of network virtualization," *University of Waterloo TR CS-2008-25*, 2008.
- [8] A. Legout, et. al., "Rarest first and choke algorithms are enough," in *Proc. of IMC*, 2006.
- [9] A. Bavier, et. al., "In VINI veritas: Realistic and controlled network experimentation," in *Proc. ACM SIGCOMM*, 2006.
- [10] S. Bhatia, et. al., "Trellis: A platform for building flexible, fast virtual networks on commodity hardware," in *Proc. of ROADS Workshop*, 2008.
- [11] K.P. Gummadi, et. al., "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proc. of SOSP*, 2003.
- [12] J. Carapinha and J. Jiménez, "Network virtualization: A view from the bottom," in *Proc. of the 1st ACM Workshop on Virtualized infrastructure systems and architectures*, 2009.
- [13] N. Christin, et. al., "Content availability, pollution and poisoning in file-sharing peer-to-peer networks," In *Proc. of the 6th ACM Conference on Electronic Commerce*, 2005.