平成 23 年度情報処理学会関西支部　支部大会

F-10

A Study on Extended Implementation of Multi-Swarm BitTorrent System with Support Nodes: Support Node and its Manager

Kei Ito†　　Yosuke Tanigawa†　　Hideki Tode†

Abstract

The aim of our research is to improve capacity of BitTorrent download, because it has some disadvantages like unavailability of some pieces of a file in network or difficulties for newcomer to download pieces for instance.

To perform this, we include in BitTorrent network a Support Node that enables to assist node in difficulties, and Support Node Manager, to strategically place Support Node in different swarms to assist several swarms at the same time.

１． Introduction

Due to the remarkable progress in network bandwidth and the storage capacity, larger and various files are sent and received on the current network.

Among the different methods to exchange data, the P2P is the one that is getting noticed, for its innovative idea of not relying on a few servers as well as its speed for delivering data that satisfies the consumers who are always looking for faster transfers.

Among many P2P protocols, one in particular has known a huge surge in what concerns file exchange: BitTorrent. This particular software even created a protocol at its name (the bittorrent protocol) that consists in fragmenting the files into pieces so that the users that detain the file, that were clients may send pieces to another user. This protocol that exists only for the user's participation trusts each client with the process of tit-for-tat: the users that help out the most benefit the most from the network's services.

This, however, has restraints and lets weak points appear in the process.

That's why we produce and install a service that enables to accelerate the connections establishment and the download speed for the users by taking in consideration the possible inaccessibility of certain files due to the users' instability. We shall introduce in the process two elements that assist the downloading: the Support Node and the Support Node Manager.

2． BitTorrent's system initially put in place

BitTorrent is file download software that uses the P2P protocol to enable to gather files for users.

It uses several parts to appoint specific terms:

- **Peer** that is user (node) contributing to BitTorrent network
- **Tracker** generally launched to a dedicated server, that is one mean to get information to all peers, and to send to each requesting peer a list of other peers that possess some piece of the requested file to allow download
- **Swarm** that is a set of peers around the same tracker
- **Leecher** that is a peer downloading file
- **Origin** that is a peer that possesses the original file
- **Seeder** that is a peer that finishes its download, and is able to send file to others

A swarm is mainly composed of one tracker and many peers which can be seeders or leechers.

Each download for one peer is performed based on the following procedure:

1. Acquisition of a torrent files on the web or by other means. Torrent files are usually set up on the web site. This torrent file stocks various information especially the address to connect to tracker and some insurance to increase security.
2. Request to the tracker for a list of peers, possessing some pieces of the file
3. Request to peers for pieces of the file
4. When a file is completely downloaded, it becomes a seeder, and helps leechers

This protocol controls the connection between peers thanks to several algorithms:

- **Tit-for-tat**: people who upload a lot are favored in the file download. This strategy is against users who download only, without upload.
- **Optimistic Unchoking**: Each downloader initially chooses 4 peers with the highest downloading rate to provide uploading according to tit-for-tat, but randomly one peer enables to contribute to other peers too. Fairness is conserved, without being favored
- **Anti-Snubbing** : Peers that do not have other peers to download the file, are able to force one peer to get the content

To succeed the best download, a specific algorithm is used by peers:

- **Rarest-piece** : It requests a rare piece to spread it in the swarm

This protocol highlights files sharing, and is most useful for the most favored files: the coveted file is downloaded by many peers in the swarm, and each applicant can easily find the file in

†Osaka Prefecture University
Department of Computer Science and Intelligent Systems

the swarm. But this aspect has inconveniences too : the files that have no success become increasingly difficult to be downloaded as time passes. Downloadable pieces entirely depend on peers, so we do not have insurance to get a requested file within a reasonable time. Furthermore, this protocol does not promote newcomers in the swarm: new downloaders that are not able to upload, are often faced with downloading with a low flow.

In order to go against the problem of file availability and restriction to newcomers or low flow users, we propose some additional nodes in the swarm to increase download speed: The Support Node.

3． One way to help: The Support Node

The Support Node (SN) is a node that is able to upload files and distribute them with different strategies from common peer.

There may be several in a swarm.

Once connected to a tracker, Support Node locates peers with low flow or finds a few newcomers, and directly assists them by providing the download of desired pieces.

The Support Node is therefore not governed by "Optimistic Unchocking" but by "Weak Peer Unchoking".

Rate is calculated with these values:

- n_i : number of nodes from which node i can download pieces
- d_i : total amount of downloaded data of node i
- u_i : total amount of uploaded data of node i
- $r_i = d_i / u_i$
- N : number of nodes
- D : sum of all download rate
- U : sum of all upload rate
- $R = D / U$

And we calculate two indexes:

$$A_1(i) = \frac{n_i}{N} \cdot \frac{d_i}{D}$$

$A_1(i)$checks if node i has good download rate (Judgment of newcomers)

$$A_2(i) = \frac{n_i}{N} \cdot \frac{r_i}{R}$$

$A_2(i)$ checks if node i has node to download (Judgment of snubbed peers)

After this calculation, Support Node takes peer that has smallest $min(A_1,A_2)$ and unchokes them.

Thus, if a peer looks for a file not possessed by any peers but by Support Node, Support Node will detect peers with low download speed and it will distribute pieces to this peers. With this strategy, it is possible to assist the newcomer and maintain a file that has no peers, by Support Node.

This will help to support an emerging small swarm.

But if the swarm extends, the Support Node would be more benefiting to help other smaller swarms.

To do this, it is necessary that a supervisor contains the information on every swarm so that it can indicate how many Support Nodes should be placed in a given swarm: It is Support Node Manager.

4． A supervisor to assist on several swarms: The Support Node Manager

The Support Node Manager is a unique node that orders Support Node to help an assigned swarm. The behavior of the Support Node Manager is described in Fig 1. It gets information on the swarm from trackers, and takes the strategy based on the number of peers and leechers in each swarm. It will prefer to assist a swarm with more leechers than seeders.

Support Node Manager calculates *seeder_rate* to check whether the number of seeder in swarm is appropriate or not, and then adds or removes Support Node to swarm.

In this way, the Support Node Manager can assign Support Nodes to a swarm which requires the support.

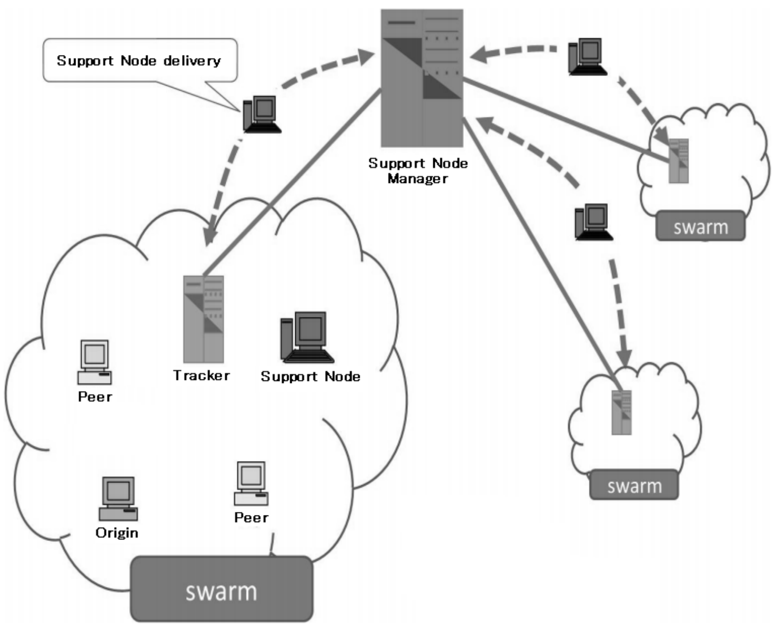
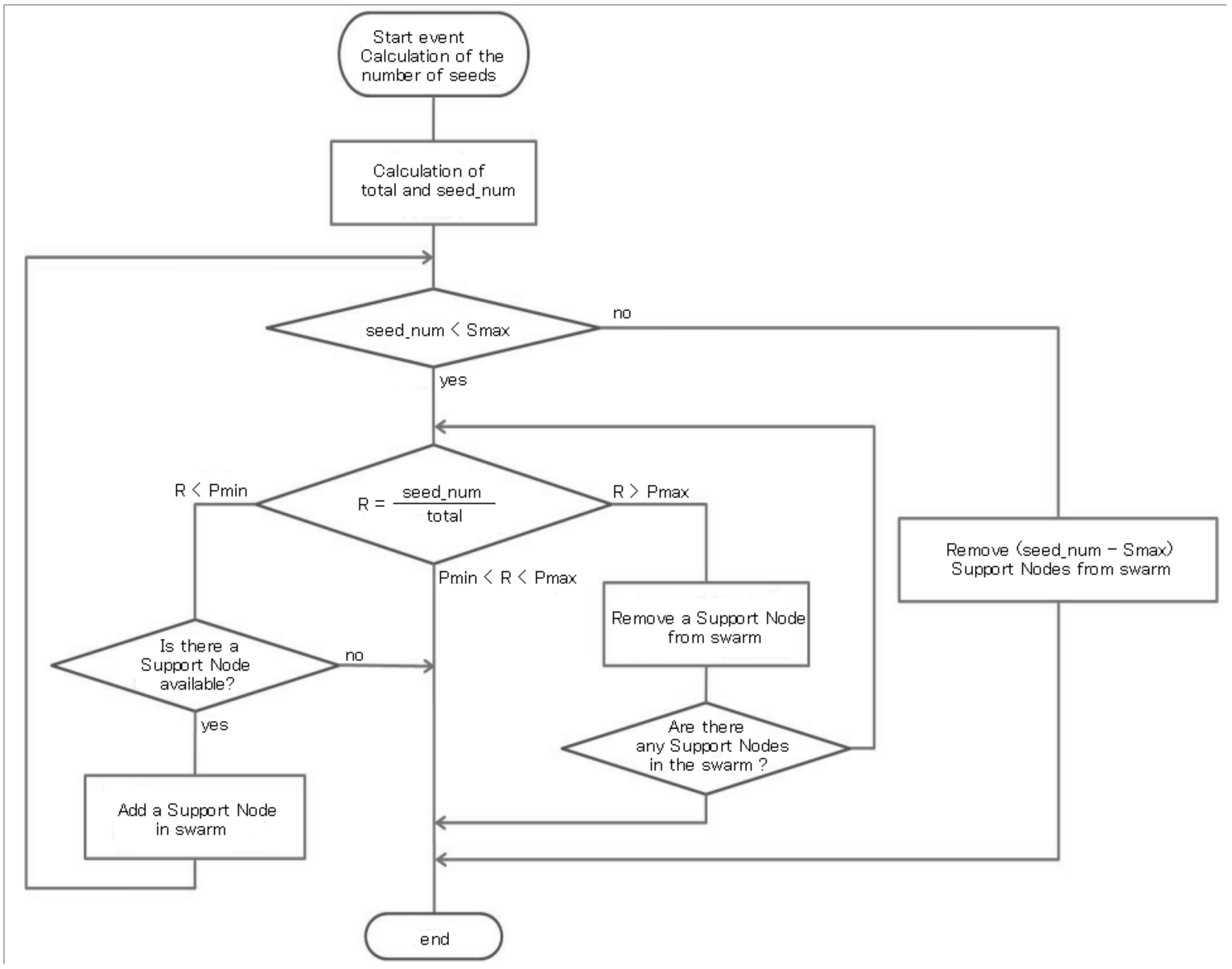


Figure 2: Support Node system schematic



Start event
Calculation of the number of seeds
Calculation of total and seed_num
seed_num < Smax
no
yes
R < Pmin
R = seed_num / total
R > Pmax
Pmin < R < Pmax
Remove (seed_num − Smax) Support Nodes from swarm
Remove a Support Node from swarm
Is there a Support Node available?
no
yes
Are there any Support Nodes in the swarm ?
Add a Support Node in swarm
end

Figure 1: Support Node Manager Algorithm

5 . Simulation

The simulation is done in a swarm with one origin, and 100 peers. The schematic view of Support Node system is illustrated in Fig 2. The simulations calculate the time it takes to have the peer's entire file.

The theoretical simulations show that the presence of a Support Node naturally helps the spread of a file in a swarm, since it is an additional upload in the swarm.

6 . Implementation

To help to get a better idea of this project, an early implementation was begun.

Implementation is based on official bittorrent-3.2.4 sources, realized in python.
Two other files are added in bittorrent source: “SupportNode.py” and “SupportNodeManager.py”.

All the developments are realized in python, and launchable for example with “python SupportNode.py” followed by their arguments. Other modified files are “track.py”, “Chocker.py”, “Rerequest.py”, but their utilization does not change anyway.

The data communication between swarms was established by a URL based method: These are HTTP requests with GET arguments that contain the information: This is a method used by BitTorrent for some operations (for example for the communication with the tracker).

To allow more opportunity in the swarm, the tracker has added some additional URL path to analyze, to gather information from Support Node and downloaders.
Thanks to this, it is able to send information to the Support Node and Support Node Manager.

Support Node only communicates with the tracker, and awaits the orders of the Support Node Manager. In the case of the tracker, it will recalculate each received information about downloads, and then make a new strategy based on new values.

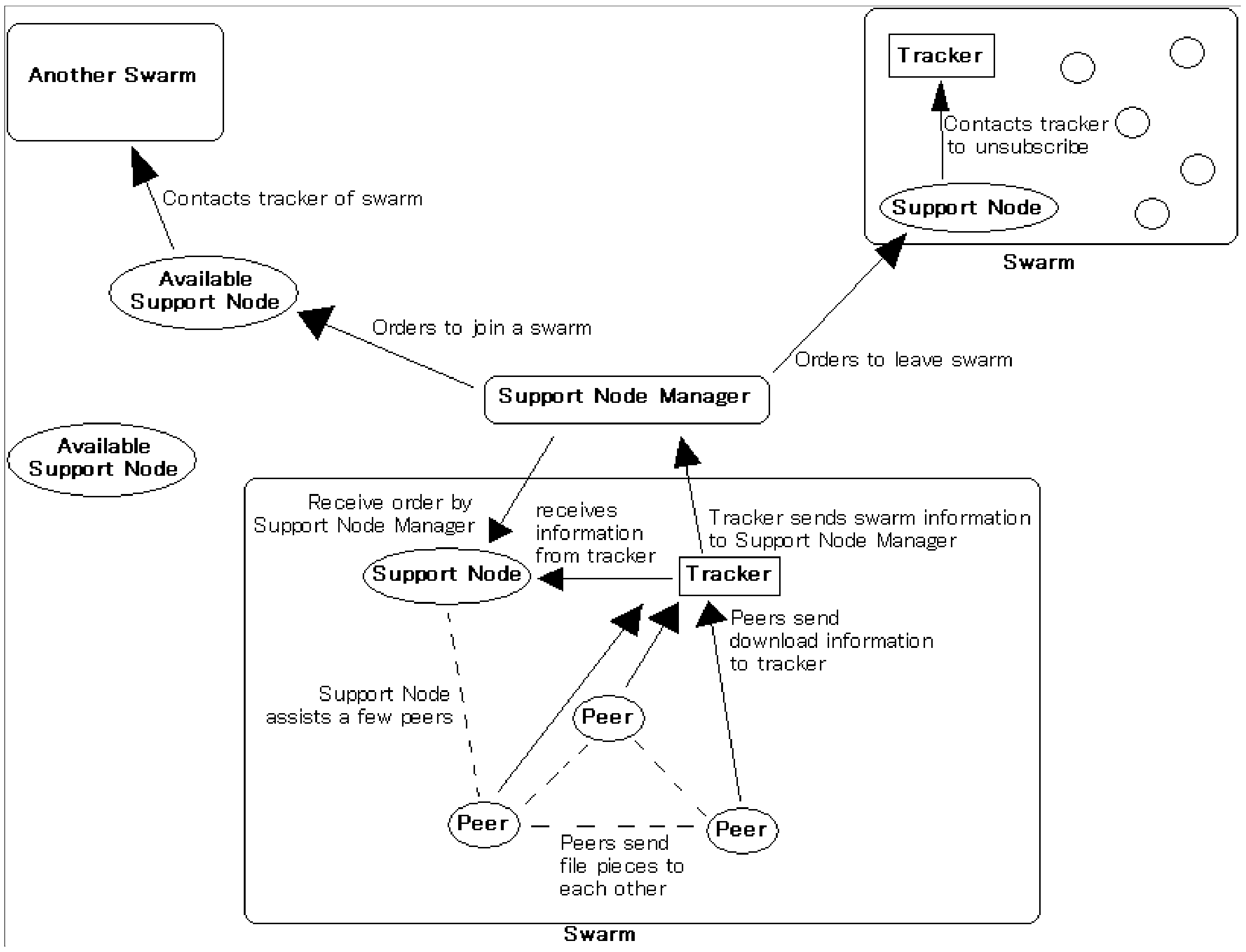

Another Swarm
Contacts tracker of swarm
Available Support Node
Orders to join a swarm
Support Node Manager
Orders to leave swarm
Tracker
Contacts tracker to unsubscribe
Support Node
Swarm
Available Support Node
Receive order by Support Node Manager
receives information from tracker
Tracker sends swarm information to Support Node Manager
Support Node
Tracker
Peers send download information to tracker
Support Node assists a few peers
Peer
Peer
Peer
Peers send file pieces to each other
Swarm

Figure 3: Support Node possibilities

On receiving an order to change swarm from a Support Node Manager, Support Node quits any work in progress, and supports new indicated swarm.

You can configure various settings:

- "*--auto_subscribe*" (or "*-s*") : Subscribe automatically to trackers
- "*--tracker*" (or "*-t*") : address of the tracker to assist on start-up (in case you would not need to multi-swarm dimension but only running in one swarm)
- "*--auto_upload*" (or "*-u*") : Upload automatically files
- "*--port*" (or "*-p*") : Port to open to receive all requests
- "*--assist*" (or "*-a*") : Number of peers to assist
- "*--support_node_manager*" (or "*-m*") : Address of Support Node Manager
- "*--debug*" (or "*-d*") : to be more verbose in output messages

The Support Node upload system is realized with the "download" function in "download.py". To perform this, it is necessary to create a torrent file to download and upload a desired file. Support Node creates automatically two folders: "./torrent", to store torrent files, and "./upload/", to store files.

Currently, it is necessary to have one file to store a list of files to download (initially "./upload/list.txt"), but this function will be removed to upload automatically all files in "./upload" or "./torrent/" folders.

Support Node Manager is a simple server that receives HTTP requests from the trackers, and performs calculations based on the information received. Then it sends orders to the available Support Node or orders to be detached from a swarm.

It reads in a file list of tracker addresses to receive information.

All the information of Support Node Manager is stored in the "./supportnodemanager/" folder, for example in "./supportnodemanager/trackers.txt" where the list of trackers can be found.

Implementation of this system is designed not to disturb the current system.

1. Launch Tracker. At this stage, the peer can connect when they want.
2. Launch Support Node Manager, calls on all their trackers to registration to be assisted by Support Nodes
3. Launch Support Nodes

This strategy was implemented to be an additional malleable option for the current swarm. All the possibilities of Support Node system is described in Fig 3.Unfortunately, it cannot assist trackers and peers without the option to send information to the

Support Node Manager and Support Node. Sending additional information is needed to operate the system of Support Node.

In the present state, it is difficult to assess the possible connection, greatly dependent on connections used by Internet users. That is why the implementation is not yet complete in its current state.

7. Some findings in the test phases

During the test phases, we have been able to notice:

- The support node can be potentially confused at first, because if peers are too consistent to help, it will help each one of them in turn, which could potentially delay the spread of a file.
- The support node cannot help if a Support Node does not have the file that the peer wants. This is a major problem, and all the tests were carried out with a Support Node holding all files, but that does not reflect reality.

8. Some possible improvements

To remedy the situation, other possible extensions of the Support Node are possible to add in.

- The ability to download a file, to increase the number of possible files to upload
- The opportunity to request a list of files available through the Support Node, directly to the tracker, to allow a better expansion of files, depending on user needs (BitTorrent suffers the lack of research of torrent file, this would help a little more with the expansion of files)
- The ability to change the settings of the Support Node, depending perhaps on the Support Node Manager.
- Download only a few pieces of files to help and save some memory
- The opportunity for a user to launch a small support node, which helps other users and enables to share his bandwidth

The Support Node has many possibilities to help the bittorrent protocol users at its best.

9. Conclusion

The Association of Support Node and Support Node Manager is a combination to help a bittorrent network and fill the gaps in the original protocol. Setting up an additional server is potentially a major drawback of this system, but the benefits are quite substantial.

Support Node wants scalable and possible changes to increase the capacity advantages of the internet support.

References

[1] “ BitTorrent Protocol Specification, ” http://wiki.theory.org/BitTorrentSpecification.

[2] Masaya YOKOHATA, Yosuke TANIGAWA, and Hideki TODE “ Implementation and Evaluation of Multi-Swarm BitTorrent System with Support Nodes ” *IEICE Technical Repport. NS2010-22,* pp.37-42, May 2010.