Performance Analysis of Controlled Scalability in Unstructured Peer-to-Peer Networks

P. M. Rubesh Anand
Research Scholar
Department of Electronics and Communication Engineering, SRM University,
Kattankulathur – 603203, Tamil Nadu, India.

Vidhyacharan Bhaskar
Professor
Department of Electronics and Communication Engineering, SRM University,
Kattankulathur – 603203, Tamil Nadu, India.

ABSTRACT
In unstructured peer-to-peer networks, the analysis of scalability is a challenging task due to the unpredictable nature of churn rate. Specifically, in the P2P file sharing applications, peers join and leave the overlay network in a dynamic fashion increase the complexity of the network and leads to a huge wastage of bandwidth during the search for a particular file. In this paper, we proposed a controlled scalability model in unstructured P2P networks for achieving efficient bandwidth utilization and high scalability. Performance measures such as peer availability and scalability are analyzed and compared with the BitTorrent system show that the proposed model overcomes the problems of bandwidth wastage and initial time flash crowd while maintaining the scalability of the network. Mathematical modeling and simulation results illustrates that the controlled scalability performs well in utilizing bandwidth during the event of polluted or infected files sharing than the uncontrolled scalability in the existing unstructured P2P networks.

Keywords
Bandwidth; File sharing; Peer-to-Peer computing; Scalability; Unstructured P2P network.

1. INTRODUCTION
In file sharing Peer-to-Peer (P2P) networks, all the peers form an overlay network and contribute their resources such as storage space, processing power and bandwidth for sharing a particular file have gained much interest recently. Unstructured P2P networks have a high resilience and tolerance to the continuous and random arrival or departure of nodes. The unstructured P2P file sharing systems started with Napster [1] had a quick journey in a decade by reaching millions of users. Napster leads to many different types of unstructured P2P file sharing systems like Gnutella, Gnutella2, KaZaA, eDonkey, LimeWire and BitTorrent. Gnutella2 and Kazaa use a hierarchical, two-tier architecture in which most of the nodes are leaf peers but some are elected as superpeers depending on their bandwidth capacity [2], [3]. The two-tiered architecture improves query efficiency and allows unstructured overlays to scale but places most of the load in a small set of nodes whose failure has a high impact on the network. BitTorrent [4] is one of the most successful P2P file sharing network used by majority of internet users. BitTorrent file-sharing protocol has a centralized server, called tracker, whose role is strictly limited in helping the peers find each other [5]. The files shared in the P2P network are hard to be checked for their genuineness well before the download process. This provides a chance for malicious nodes to share vast number of polluted (corruption of files due to malicious codes or transmission errors) or infected (any file intentionally hiding the malicious codes to affect the integrity of the peer's system that downloads it) files in the network for their personal benefits [6]. Moreover, the peers with bandwidth limitations are affected severely by these polluted and infected files as their bandwidth is wasted in downloading such fake files. Scalability is an important factor in the unstructured Peer-to-Peer network expansion [7]. In the event of polluted or infected file sharing, high scalability leads to more wastage of bandwidth. However, in the unstructured P2P network, scalability is uncontrolled as the overlay links are established arbitrarily [8], [9]. Any new peer that wants to join the network can copy and form its own links from the existing links of another node. As the links between the peers are uncontrolled and unpredictable, the analysis of scalability is considered tedious in unstructured P2P networks [10], [11]. In our paper, we propose and analyze a model with controlled scalability to detect and prevent polluted or infected file from spreading in the network. In the model, scalability is restricted in the initial time of sharing process for solving problems of bandwidth wastage, flash crowd, file pollution and poisoning. Once any polluted or infected file is detected by the tracker, that particular file is removed from the network. In other cases of unsure detection of polluted or infected file, the tracker imposes restricted scalability till any concrete decision is met.

This paper is further organized into six sections. Following an introduction to unstructured P2P system in section 1, section 2 covers the model description and controlled scalability, section 3 explains the mathematical model and analysis, section 4 shows the theoretical and simulation results along with the discussions, and section 5 presents the conclusion.

2. MODEL DESCRIPTION
In the proposed model for Peer-to-Peer file sharing networks, controlled scalability is considered in the initializing and stabilizing phase for the peers in the overlay network [12]. The decaying phase in which the peers leave the network indefinitely works as in existing P2P file sharing systems.

2.1 Initialization Phase
In the existing BitTorrent-like systems, a provider (peer) shares a particular file by generating a .torrent for the file to be shared [13]. The generated .torrent contains information about file name, size, and hash values. The provider needs to authenticate with its login ID and password to the centralized server in order to publish its .torrent in the web server or an anonymous upload of file is also allowed without revealing any identity. In the BitTorrent-like systems, any peer can download the .torrent file without going through the authentication process. In order to ease the search for a

genuine and unpolluted file by the anonymous peers, the proposed model has two different webpages. When a peer authenticates itself with the authentication server, it gets access in the unrestricted web page containing all the existing .torrent including old and new ones. If any peer fails to get successful authentication or wants to be in an anonymous state, it will be directed to the restricted web page containing only the verified and genuine .torrent. The .torrent file can be downloaded from both the webpages depends on the visibility and searchability of the .torrent that is controlled by the webserver. Once the .torrent is obtained, the peer contacts the tracker to get the list of peers for bootstrapping into the overlay network as shown in Fig. 1.

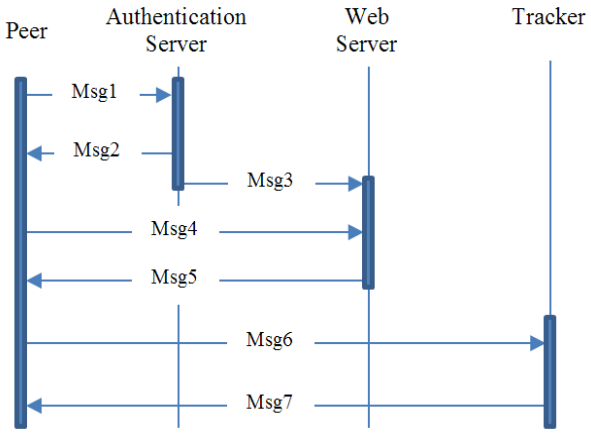

Fig 1: Sequence diagram of a peer establishing connection with the tracker through an authentication process [12].

The tracker serves as a managing entity to detect the polluted or infected files that are shared in the P2P overlay network. In this phase, the tracker restricts the scalability of the network by allowing only a limited number of authenticated peers to join the network in a first-come-first-served basis for any new file available for sharing. This indeed solves the problem of flash crowd initially. Once a set of limited number of peers are bootstrapped into the overlay, the excess number of peers are queued until the next phase. The messages specified in the process of peer establishing connection with the tracker are given as follows:

Msg1: Request for authentication using login ID and password;
Msg2: Reply from Authentication Server if the process is successful;
Msg3: Authentication Server intimates to the Web Server with peer details;
Msg4: Request for the meta-data of a selected file from the list of downloadable files;
Msg5: Reply with the meta-data to the client version of the P2P program;
Msg6: Request for the list of peers to get bootstrapped in the overlay;
Msg7: Reply with the list of peers connected to the particular overlay network.

2.2 Controlled Scalability

During the initialization phase, the file sharing process is initiated with restriction in scalability. In the stabilization phase, the tracker decides to stabilize the P2P network while a genuine file is shared. When the shared file is found to be polluted or infected, the tracker terminates the overlay network from further expansion.

Table 1. Notations and Definitions

Notation	Definition
N_s	Network size
N_p	Permitted Network size
S	Number of Seeds
P	Number of Peers
D	Detection of Polluted / Infected file
ON_i	Overlay Network number
$StatusON_i$	Overlay Network number existence
R	Restrictions to scalability

The decision process of the tracker is supported by the inputs like overlay network identification number, network size, number of peers or leeches and number of seeds as given in Algorithm 1. In our model, when the first set of peers download the file, they verify it with their resources like anti-virus software or sand-box technique for the authenticity and originality of the file. If they are convinced with the description of the content and the file downloaded, they stay in the overlay network for further sharing. Otherwise, they discard the file by deleting it and exiting from the overlay network. This prevents spreading of polluted or infected file in the network. When the number of seeds for any particular overlay network increases during the initial restrictions, the tracker allows for unrestricted scalability of the overlay network. In case of unsure detection of polluted or infected file, the tracker continuously imposes restricted scalability of the overlay network till any concrete decision is met. In the model, two controlled scalability approaches are considered depending upon the network size and permitted network size. In the first approach, network size is not allowed to exceed the permitted network size and in the second approach, permitted network size is varied to accommodate multiple numbers of peers in the network. After the restrictions on scalability are removed, the system works as in the existing BitTorrent-like systems.

Algorithm 1. Controlled Scalability for detection of Polluted file
Input: ON_i, N_p, N_s, S, P
Output: Status of D, R, ON_i

```
        / / Tracker's side at every time interval / /
01: repeat
02:     if (S < P and S < Ns/2) then
03:         R ← True;
04:         D ← False;
05:         repeat
06:             if (Ns < Np/2) then
07:                 add (Np – Ns) peers to the network;
08:             else
09:                 add (S*S) peers to the network;
10:             end
11:         until R = False;
12:     else if (S > P and S < Ns/4) then
13:         D ← True;
14:         StatusONi ← False;
15:     else
16:         R ← False;
17:         D ← False;
18:     end
19: end
20: until StatusONi = True;
```

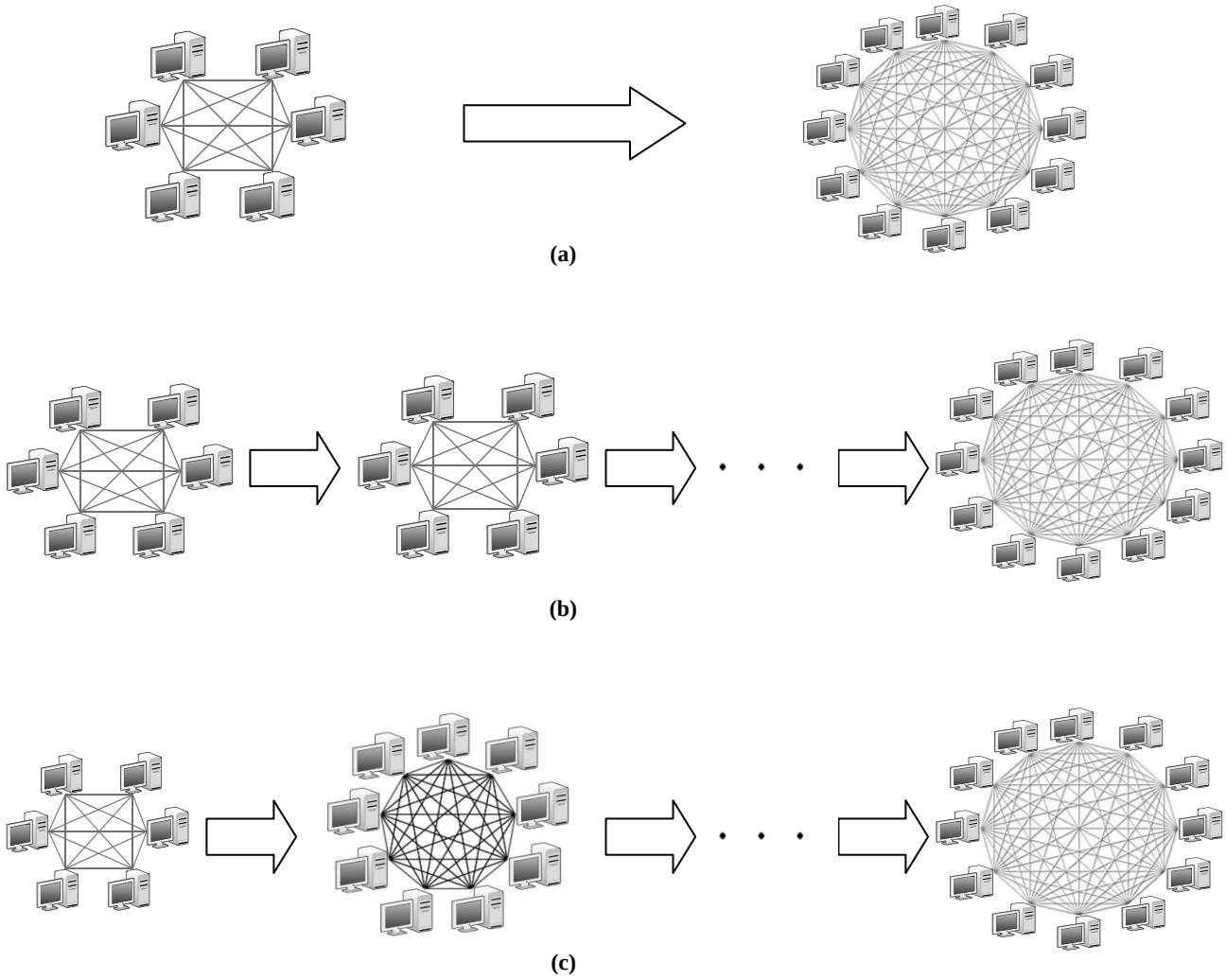

Fig 2: Controlled Scalability in (a) unpolluted or uninfected file sharing, (b) unsure detection of polluted or infected file with constant network size, and (c) unsure detection of polluted or infected file with variable network size.

3. MATHEMATICAL ANALYSIS

The P2P networks can be considered as homogeneous or heterogeneous depending upon the peers' resource diversity. In homogeneous P2P network, the peers are considered to have identical resource capability, whereas in heterogeneous P2P network, the peers have varying resource capacities including their contribution level. In both cases, unpredictable churn rate is common for the unstructured P2P networks. The mathematical analysis of controlled scalability considers two methods for P2P overlay network expansion by considering the number of seeds, namely, the n^{th} stage seeds only and combined number of seeds from all the stages. In the situation of unsure detection of polluted files, controlled scalability is considered in the following three cases until the file is found to be a genuine one.

3.1 Case 1: Controlled scalability with all cooperating seeds

As a theoretical ideal case, the first case considers all seeds to be cooperating in sharing files. Let, '*S*' number of peers constantly bootstrapped in the network at each stage by every seed, with the assumption that all seeds cooperate for the overlay expansion, is shown in Table 3. In the initial stage, a single provider starts the sharing process with '*S*' peers, which leads to '*S*' number of seeds available at the end of the first stage.

Table 2. Notations and their interpretations

Notation	Interpretation
n	Overlay stage number
N_1	Total number of seeds in the n^{th} stage only
N_2	Total combined number of seeds in *n* stages
1	Initial number of seeds
S	Number of peers added for each seed
u	Constant number of non-cooperating seeds
u_i	Variable number of non-cooperating seeds

Table 3. Unstructured P2P Network expansion with all cooperating seeds

Overlay Stage (n)	Mathematical Representation	Mathematical Equivalent
0	1	1
1	$1*S$	S
2	$1*S*S$	S^2
3	$1*S*S*S$	S^3
4	$1*S*S*S*S$	S^4
⋮	⋮	⋮
n	$1*S*S*S*S*\cdots*S$	S^n

When the network expands through each stage, multiples of '*S*' number of seeds are attained in every stage leading to (1) in n^{th} stage, given by

$$N_1 = S^n. \tag{1}$$

The contribution from all seeds is obtained by the summation of the number of seeds from the initial stage till the n^{th} stage, given by

$$N_2 = \sum_{i=0}^{n} S^i. \tag{2}$$

3.2 Case 2: Controlled scalability with constant non-cooperating seeds

In the P2P network, non-cooperating peers are formed due to various reasons, like their malicious behavior, free-riding nature or unavoidable circumstances. Those non-cooperating peers or seeds are excluded from the overlay network at every stage for monitoring the exact number of seeds and their contributions as specified in the second case, as shown in Table 4. A constant number of non-cooperating peers are removed from the overlay network during the addition of new peers. The number of seeds in the n^{th} stage after removing '*u*' non-cooperating seeds from each stage is given by

$$N_1 = S^n - u\left[S^{n-1} - S^{n-2} - S^{n-3} - \cdots - S^2 - S^1 - S^0\right]. \tag{3}$$

Equation (3) can be rewritten in the general form as,

$$N_1 = S^n - u\sum_{i=0}^{n-1} S^i. \tag{4}$$

The combined number of seeds from the initial stage is given as,

$$N_2 = 1 + \sum_{j=1}^{n}\left[S^j - u\sum_{i=0}^{j-1} S^i\right]. \tag{5}$$

3.3 Case 3: Controlled scalability with variable non-cooperating seeds

In the third case of the unstructured P2P network, the number of non-cooperating seeds '*u*' in each stage is considered as a variable whose value varies from 0 to *S*. The overlay network expansion through each stage is shown in Table 5.

For a constant '*S*' peers joining the network and variable (u_1, u_2, u_3, ...u_n) non-cooperating peers removed from the network, the total number of seeds in (4) and (5) are rewritten as (6) and (7), respectively, while considering u_i non-cooperating peers at each stage (*i*), as given by,

$$N_1 = S^n - \sum_{i=0}^{n-1} S^i u_{n-i}, \tag{6}$$

$$N_2 = 1 + \sum_{j=1}^{n}\left[S^j - \sum_{i=0}^{j-1} S^i u_{j-i}\right]. \tag{7}$$

Table 4. Unstructured P2P Network expansion with 'u' non-cooperating seeds

Overlay Stage (n)	Mathematical Representation	Mathematical Equivalent
0	1	1
1	$(1*S)-u$	$S-u$
2	$S[(1*S)-u]-u$	S^2-Su-u
3	$S[S[(1*S)-u]-u]-u$	$S^3-S^2u-Su-u$
4	$S[S[S[(1*S)-u]-u]-u]-u$	$S^4-S^3u-S^2u-Su-u$
⋮	⋮	⋮
n	$S\cdots[S[S[(1*S)-u]-u]-u]-\cdots-u$	$S^n-S^{n-1}u-S^{n-2}u-\cdots-S^{n-n}u$

Table 5. Unstructured P2P Network expansion with 'u_i' non-cooperating seeds

Overlay Stage (n)	Mathematical Representation	Mathematical Equivalent
0	1	1
1	$(1*S)-u_1$	$S-u_1$
2	$S[(1*S)-u_1]-u_2$	$S^2-Su_1-u_2$
3	$S[S[(1*S)-u_1]-u_2]-u_3$	$S^3-S^2u_1-Su_2-u_3$
4	$S[S[S[(1*S)-u_1]-u_2]-u_3]-u_4$	$S^4-S^3u_1-S^2u_2-Su_3-u_4$
⋮	⋮	⋮
n	$S\cdots[S[S[(1*S)-u_1]-u_2]-u_3]-\cdots-u_n$	$S^n-S^{n-1}u_i-S^{n-2}u_{i+1}-\cdots-S^{n-n}u_n$

4. RESULTS AND DISCUSSIONS

The mathematical model of case 2 and case 3 are plotted for each overlay network stage seeds and combined stage seeds in Figure 3 and 4. Here, each stage is assumed when controlled scalability is applied for bootstrapping of the peers according to Algorithm 1. The controlled scalability allows only permitted network size to evolve for the network expansion until the number of seeds are sufficiently high compared to the network size.

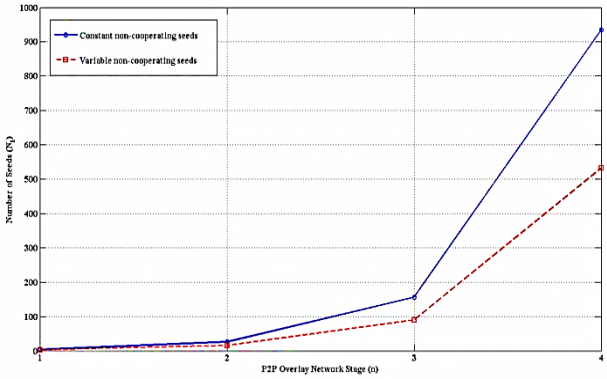

Fig 3: Comparison of the number of seeds in each stage for case 2 and case 3.

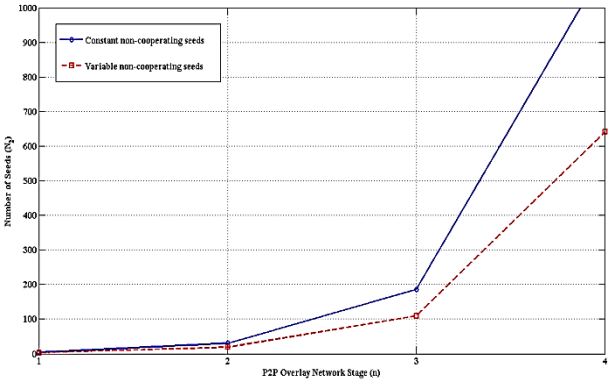

Fig 4: Comparison of the number of seeds in combined stages for case 2 and case 3.

From the Figures 3 and 4, it is observed that controlled scalability still provides sufficient number of seeds during the unsure detection of polluted or infected files. Comparing N_1 and N_2, variable number of non-cooperating seeds show slow improvement in achieving scalability with that of the constant number of non-cooperating seeds. Though the variable number of non-cooperating seeds suits for the practical situation, the slow scalability helps in identifying and removing the polluted or infected files from being shared in the network.

We simulate our proposed model in PeerSim [14] upto the scalability of 5000 nodes in event driven mode. The simulation uses existing BitTorrent model and the proposed model. In the simulation, the file size considered is 100 MB, which is split into pieces of 256 KB each and further each piece is split in 16 blocks with each block size of 16 KB. In BitTorrent model, each node is configured as in the real world with the existing BitTorrent protocol setup by allowing uncontrolled churn rate. Choking algorithm using tit-for-tat mechanism, rarest first, endgame mode, tracker down time, flash crowd and peer down time are considered for both BitTorrent and the proposed model simulations.

The controlled scalability in the simulated environment shows efficient utilization of bandwidth in the initial stages as in Figure 5. Comparing with the BitTorrent simulation, our model also achieves better scalability in the latter stages. The constant and variable removal of the non-cooperating seeds helps in better search efficiency and uninterrupted download of the files. This aids the proposed model in performing better than the BitTorrent in the due course of time.

Trend analysis is a statistical means of predicting or extracting the pattern of behavior in the time series. We analyzed the data on the number of seeds for different time using exponential trend line. As in the practical case, the number of peers joining in the network for downloading files increases exponentially, the comparison shown in Figure 6 highlights the scalability of our proposed model in par with the BitTorrent system.

In the initial time of file sharing, a sudden increase in the number of peers happens to connect with the provider to download a particular file. This leads to a situation like DoS (Denial of Service) attack on the provider. When restrictions are applied on the initialization phase of the P2P network, DoS attack-like situation is prohibited as shown in Fig. 7. The comparison of the proposed model and the BitTorrent system shows that in the event of flash crowd, the proposed model efficiently protects the initial provider from the problem of DoS attack, simultaneously allowing the network to expand after acquiring certain number of seeds.

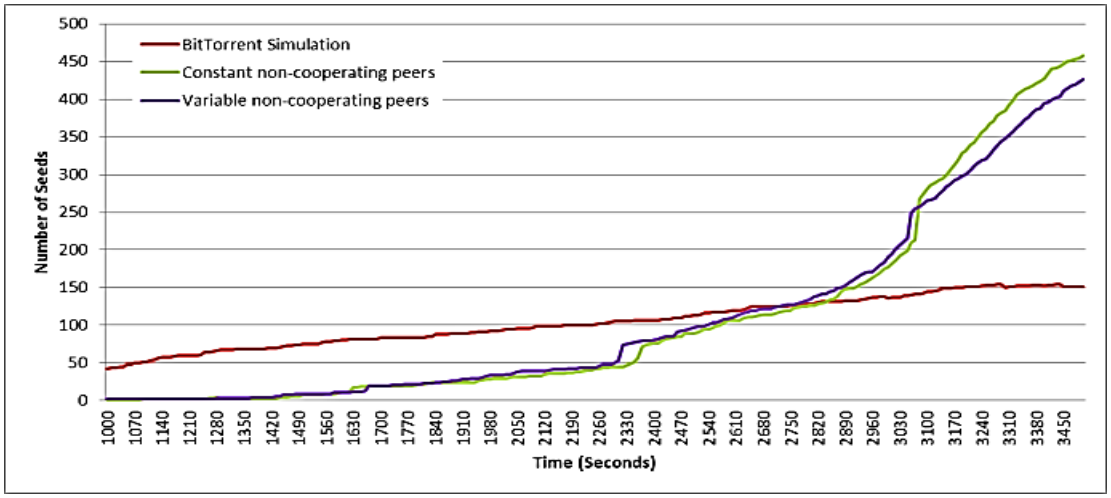

Fig 5: Comparison of the number of seeds in BitTorrent simulation and controlled scalability.

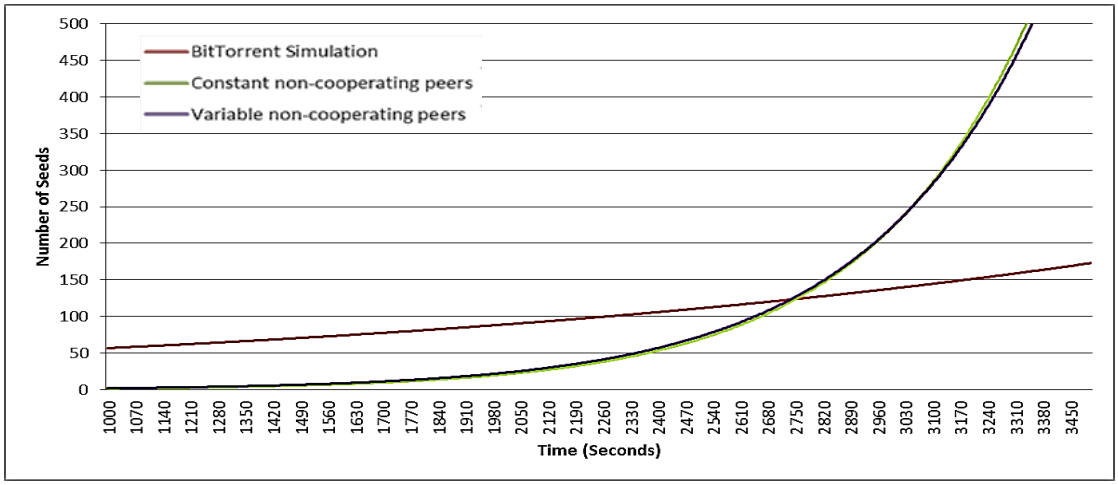

Fig 6: Comparison of the BitTorrent simulation and controlled scalability through trend analysis.

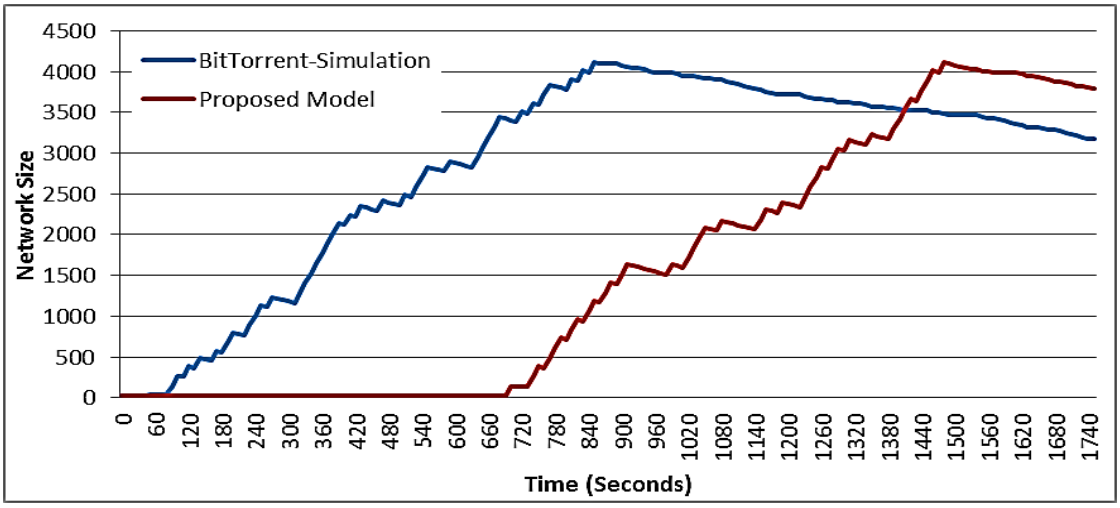

Fig 7: Comparison of the effect of network size between BitTorrent and proposed model in the presence of flash crowd.

5. CONCLUSION

We proposed a controlled scalability model to handle the polluted or infected files shared in the unstructured P2P file sharing networks. The restriction in scalability during the initialization phase has efficiently solved problems of flash crowd, infected or polluted file, and bandwidth wastage than the existing models of BitTorrent-like systems. The performance analysis of the mathematical and simulated model helps in predicting the scalability in the occurrence of unsure detection of polluted or infected files. The trend analysis and simulation results show that the model performs well compared to BitTorrent-like unstructured P2P systems in the presence of polluted or infected files by saving large amount of bandwidth and time of individual peers.

6. REFERENCES

[1] Giesler, M. and Pohlmann, M. 2003. The Anthropology of File Sharing: Consuming Napster as a gift. Advances in Consumer Research. vol. 30, 273-279.

[2] Gnutella2. [Online]. Available: http://en.wikipedia.org/wiki/Gnutella2

[3] Liang, J., Kumar, R., and Ross, K. 2005. The Kazaa Overlay: A Measurement Study. Computer Networks (Special Issue on Overlays) (2005).

[4] BitTorrent. Available: http://www.bittorrent.com

[5] Cohen, B., 2003. Incentives Build Robustness in BitTorrent. In Proceedings of the Workshop on Economics of Peer-to-Peer Systems, Berkeley, California, USA, (May 2003).

[6] Christin, N., Weigend, A.S., and Chuang, J. 2005. Content Availability, Pollution and Poisoning in File Sharing Peer-to-Peer Networks. In Proceedings of the 6th ACM conference on Electronic commerce (EC'05), Vancouver, Canada, (June 2005), 68-77.

[7] Rubenstein, D. and Sahu, S. 2005. Can unstructured P2P protocols survive flash crowds? IEEE Transactions on Networking, vol. 13, no. 3, 501-512.

[8] Chiu, Y.-M. and Eun, D.Y. 2010. On the Performance of Content Delivery under Competition in a Stochastic Unstructured Peer-to-Peer Network. IEEE Transactions on Parallel and Distributed Systems, vol. 21, no.10, 1487-1500.

[9] Yang, M. and Yang, Y. 2010. An Efficient Hybrid Peer-to-Peer System for Distributed Data Sharing. IEEE Transactions on Computers, vol. 59, no. 9, 1158-1171.

[10] Marchetto, G., Manzillo, M.P., Torrero, L., Ciminiera, L., and Risso, F. 2011. Robustness analysis of an unstructured overlay for media communication. IET Communications, vol. 5, no. 4, 409-417.

[11] Smaragdakis, G. et al., 2011. Selfish Overlay Network Creation and Maintenance. IEEE/ACM Transactions on Networking, vol. 19, no. 6, 1624-1637.

[12] Anand, P.M.R., and Bhaskar, V. 2011. Polluted Content Prevention in Peer-to-Peer File Sharing Networks. In Proceedings of the IEEE Conference on Engineering Sustainable Solutions, (Dec. 2011), 1-4.

[13] Izal, M., Biersack, E. W., Felber, P. A., and Hamra, A. A. 2004. Dissecting BitTorrent : Five Months in a Torrent's Lifetime. Passive and Active Network Measurement, Lecture Notes in Computer Science, Springer, vol. 3015, 1-11.

[14] Montresor and Jelasity, M. 2009. PeerSim: A scalable P2P simulator. In Proceedings of the IEEE Ninth International Conference on Peer-to-Peer Computing, Seattle, Washington, USA, (Sep. 2009), 99-100.