International Journal of Computers and Applications, Vol. 30, No. 4, 2008

ANALYSIS ON A SIMULATED MODEL FOR GNUTELLA TOPOLOGY: CONNECTEDNESS AND EXTENSION

J. Sum,* ,*** J. Wu,** C.-S. Leung,*** and K.I.-J. Ho****

Abstract

This paper presents a model to generate and study Gnutella topology from an original point of view. Instead of using characteristics of the final topology, the network is constructively created from scratch and its connectedness is studied by simulation. As the resultant topology has the same node degree distribution as what has been measured from the true Gnutella and a virus outbreak simulation has shown that the network is not connected, it is argued that the true Gnutella might not be a connected network. To improve the connected ability of the model, a modification on the connection mechanism is proposed and the topological change of the network is studied by simulation. Although the node degree distribution of the resultant topology is deviated from the measurement results, this new connection mechanism can indeed improve the connectedness of the network that is confirmed by the virus outbreak simulation.

Key Words

Gnutella, network connectedness, peer-to-peer, simulation

1. Introduction

Understanding the underlying structure of the Gnutella peer-to-peer (P2P) network is not a new problem. As a number of natural networks have been uncovered to have small-world phenomena [1], i.e., the node degree distribution follows the *power law*, a number of studies on complex networks have been conducted in the last decade, including the Internet [2], World Wide Web [3] and the Gnutella P2P network [4].

As the Gnutella is a popular P2P network for file sharing, many measurement studies have been conducted in the recent years to investigate the underlying topology and other properties of such a network, and eventually to design better algorithm to improve the efficient of the network. After Jovanovic's preliminary study [4], Saroiu, Gummadi and Gribble gave a comprehensive measurement study on Gnutella in [5]. Not only the power law property has been re-discovered, they have also figured out a significant group of peers called free riders. This group of users only download files. But they do not share any files to other peers. As Gnutella is a dynamic and growing peer-to-peer network, Ripeanu *et al.* did two measurements in 2000 and 2001 [6] and compared their node degree distributions. It was found that the distribution seems to have changed from a power law distribution to a mixture of power law and Poisson.

Although a lot of measurement results have been reported in the subsequent years [7–9], not much work has been done on the *connectedness* property of the network. *Connectedness* is a very important property that the Gnutella should have. In the past few years, it is simply made as an implicit assumption while a search algorithm is evaluated [10–12]. If Gnutella is a connected network, the performance of a search algorithm could solely be determined by the computational resources needed for a search and the time spent. If Gnutella is not a connected network, the coverage of a search will become a critical factor that every search algorithm needs to take into account. Besides, one might need to design an alternative peer node connection mechanism like Phenix [13] to alleviate such limitation.

In regard to the importance of *connectedness*, the topology of Gnutella will be the primary focus that we will elucidate in the rest of the paper. As mentioned before, reconstructing the network topology simply by connecting the neighbourhood IPs measured as was done in [4] is not appropriate. We take the following methodology to investigate the *connectedness* issue:

- The node degree distribution and the lifetime distribution for the ultrapeer nodes amongst the ultrapeer–ultrapeer subnetwork are measured.
- Then we proposed a peer connection mechanism that was inspired by an S.N. Dorogovtsev and J.F.F. Mendes evolution network model (p. 121 in [14]).

* Institute of Electronic Commerce, National Chung Hsing University, Taichung 402, ROC; e-mail: pfsum@yahoo.com.hk
** Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, USA; e-mail: jie@cse.fau.edu
*** Department of Electronic Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong; e-mail: eeleungc@cityu.edu.hk
**** Department of Computer Science and Communication Engineering, Providence University, Sha-Lu, Taichung, ROC; e-mail: ho@pu.edu.tw
Recommended by Dr. Tiziana Calamoneri
(paper no. 202-1958)

- A sequence of simulated networks are thus generated and their node degree distributions are recorded.
- Once the network has reached its stationary condition (i.e., once (i) the number of online nodes, the total number of edges and the node degree distribution do not change much), the node degree distribution generated by this simulated Gnutella is compared with measurement result to validate the viability of the simulated model.
- The connectivity of this simulated Gnutella is then studied by (i) using the criteria derived by M. Molloy and B.A. Reed in [15] and (ii) a virus outbreak simulation.

The other contribution of this paper is an improved connection mechanism for Gnutella and demonstrate by simulation how a connected network can be obtained.

In the next section, we will describe the peer connection mechanism of Gnutella. Section 3 presents the measurement results on the node degree distribution and the lifetime distribution. The behaviour and the connectedness of the simulated Gnutella will be elucidated in Section 4. Section 5 presents an improved connection mechanism we proposed. Its behaviour in node degree distribution and network connectedness will be described. Section 6 discusses a simulation result of a variant of the model presented in Section 4. Section 7 gives a conclusion of the paper and introduces possible future work.

2. Peer Connection Mechanism

In accordance with Gnutella protocol [16], while a new peer (say Peer A) is trying to make connection to the existing Gnutella network, Peer A can either send a XTRY or PING message to an online Gnutella peer (say Peer B) asking for the IP addresses of its neighbour. If XTRY is used, Peer B will return with a list of its direct neighbours. If PING is used, Peer B will broadcast the request to its direct neighbours it is connecting to and its direct neighbours will then broadcast the request to their direct neighbours, so on and so forth until the message has been propagated a TTL (Time-To-Live) hops away. All their IP addresses will then propagate backwards to Peer B based on the PONG protocol. After receiving these PONG messages, Peer B will send a PONG message to Peer A. No matter if the XTRY or PING-PONG protocol is used, Peer A can select k IP addresses from the list to connect to.

Several remarks should be noted for this connection mechanism. The first remark is about the *seed IP* – where a peer can get the online peers' IP. Once a peer, say Peer A, wants to connect to Gnutella, it needs to send a message to an online Gnutella peer. Basically, a software developer will set two default locations for a peer to get online peers' IP addresses. The first location is the local cache. It stores all the IP addresses a peer connected to before. The second location is a well-known global server, limewire.com for instance. It stores the IP addresses of all the current online peer nodes. Whenever a peer wants to connect, it can get the IP addresses from these locations and select one IP as seed for XTRY or PING.

The second concerns *the number of seed nodes*. Suppose there are two available seed nodes, say X and Y, having been searched from the local cache. In the current setting Peer A is unlikely to ask Peer Y for further information if Peer X has been asked and replied with a list of IP addresses. Therefore, we can assume only one seed node will be XTRYed whenever a peer node wants to make a connection to Gnutella.

The third remark is about *the connectedness of Gnutella*. In the past few years, researches on the file searching mechanism usually assumed that the Gnutella is a connected network. However, no solid evidence has been reported to confirm this claim. Fig. 1 shows the situation when only one seed node will be XTRYed and the maximum number of new connections a new node can make is 3. Suppose a new node A has been loggon, it makes new connections to node B and two of its neighbours (Fig. 1(a)). In that moment of time, the network is connected and a search initiated from node A can reach the subgraph G_2. But when node C is out, it is clear that the network will be disconnected and then a search initiated from node A can only reach the subgraph G_1 (Fig. 1(b)). The results replied from a search will definitely be limited to the size of the component (G_1) in which the search initiated.

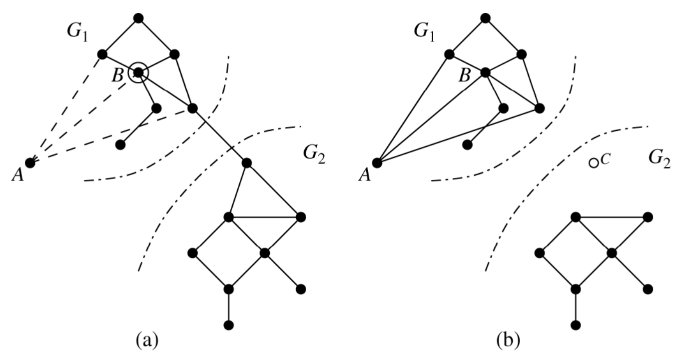

Figure 1. Network disconnection due to a node out.

3. Measurement Results

The measurement was carried out in August 2003. The measurements of the node degree and the node lifetime were carried out in two consecutive phases.

In the first phase, we started with an active Gnutella Ultrapeer whose IP address was obtained from limewire.com. Then, other active peers were crawled, and a master list of 100,000 IP addresses was maintained. This master list of IPs constituted the basis set of nodes we would use to capture node IDs.

In the second phase, we repeatedly sent out PING messages to these 100,000 IP addresses every 15 min. An IP is assumed to be an active Gnutella peer if it replies. If an IP does not reply for two consecutive accesses, we assume the Gnutella peer has been logged out. As some of these IPs are possibly from Internet Service Providers (ISPs) and these IPs might be assigned to other Gnutella users as well, PING messages are continuously sent to them. But any new replies afterward will be treated as if they are from a new Gnutella user.

3.1 Node Degree Distribution

Instead of considering all the Gnutella nodes, we focus on the ultrapeer–ultrapeer subnetwork. As the Gnutella Protocol v0.6 allows a peer node to be shielded as a leaf, this will over count the node degrees being measured. To maintain the nodes uniformity as the when studying the topology of the Internet, only the subnetwork constituted by ultrapeer nodes will be investigated. The total number of ultrapeers that we have identified from the PONG messages is 5,067. The node degree of an ultrapeer is then defined as the number of ultrapeers within this subnetwork it is connected to.

It should be noted that the actual figure can hardly be obtained. Gnutella peer can deny a reply to a PING message in a number of different ways. One might be that the port buffer is full. Another might be that the quota limit for the connections a peer can make has been reached. Therefore, in the rest of the paper we assume that the actual node degree is the number tallied from the PONG message. The node degree distribution for the ultrapeer–ultrapeer subnetwork is obtained and shown in Fig. 2. Observed from the plot, the distribution looks like a mixture of two power-law distributions:

$$P_n(k) \sim \begin{cases} k^{-0.8} & k \leq 5 \\ k^{-4} & k > 5 \end{cases} \tag{1}$$

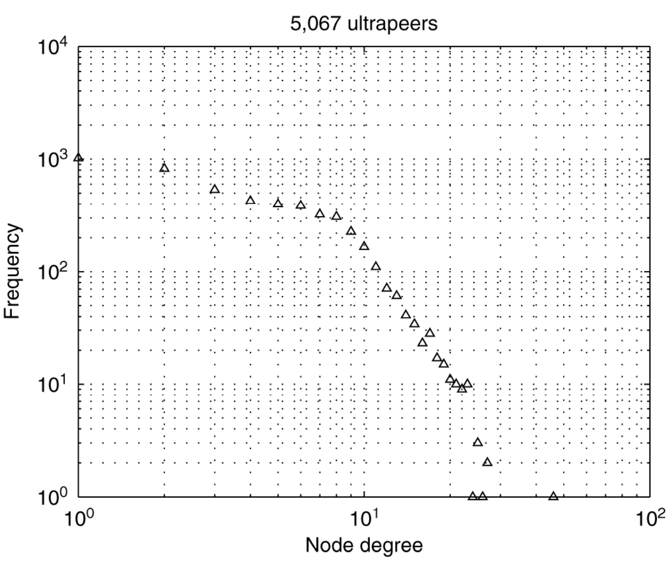

Figure 2. Node degree distribution of the ultrapeer–ultrapeer subnet of Gnutella: for $k \in [10, 20]$, $P_n(k) \propto k^{-4}$.

3.2 Lifetime Distribution

For the lifetime distribution, measurements were carried out together with the measurement of node degree. Once a node was found to be active, its neighbour peers were traced using the Gnutella connection protocol. Via the connection protocol, a long list of IP addresses were obtained. Then, connection request messages were sent to these IP addresses. By analyzing the return message from these IPs, we are also able to identify whether the peer node is a shielded leaf node or an ultrapeer node.

Measuring the lifetime of a Gnutella peer is more difficult. To measure the lifetime distribution of the ultrapeers, a PING message is repeatedly sent to the 100,000 IP addresses that have replied within every 15 min interval. As long as a reply has been received, the peer node is assumed to be active. That is to say, the node was still connected to the Gnutella P2P network. On the other hand, if no reply has been received, the peer node is assumed inactive. To avoid miscounting due to a network fault, a peer node is assumed inactive only if no reply has been received for more than 30 min (i.e., two consecutive trials). It should be noted that a Gnutella peer might not have a permanent IP address. Most of these IP addresses are dynamic IPs that are provided by the ISP. Once a peer node has disconnected from Gnutella, it might also be disconnected from its ISP. In such case, the ISP might assign the IP address to another user. Hence, we need to assume that one IP address is occupied by only one user at one session. The lifetime distribution is shown in Fig. 3. It is found that the lifetime distribution follows Gamma distribution [17]:

$$P_l(t) \sim t^{-1.12} \exp(-t/780) \tag{2}$$

where t is in the scale of minutes. It should be noted that Daniel Stutzbach and Reza Rejaie [7] have recently measured and reported a result on the lifetime distribution of Gnutella ultrapeers. Assuming that the shape of the lifetime distribution follows the Power-Law, they have found that $P_l(t) \sim t^{-1.75609}$. As their measurement is from September 2004, which is 1 year after the time of our own measurement, it would be an interesting issue to investigate if the session lifetime distribution is dynamically changing over time.

4. Simulated Gnutella

As the measurement of the node degree distribution and lifetime distribution will take a period of a couple of days, the actual network topology in any particular instant is still unknown. To observe the transient behaviour and investigate the topological evolution, the following Gnutella-like network connection model extended from a Dorogovtsev-Mendes (p. 121 in [14]) evolution network model is used. Studies on the topological evaluation will be accomplished by simulation.

1. s nodes are fully connected initially.
2. At time t,
   - The lifetime of each existing node is incremented by 1.
   - n new nodes are added to the network with the lifetime being set to 1.

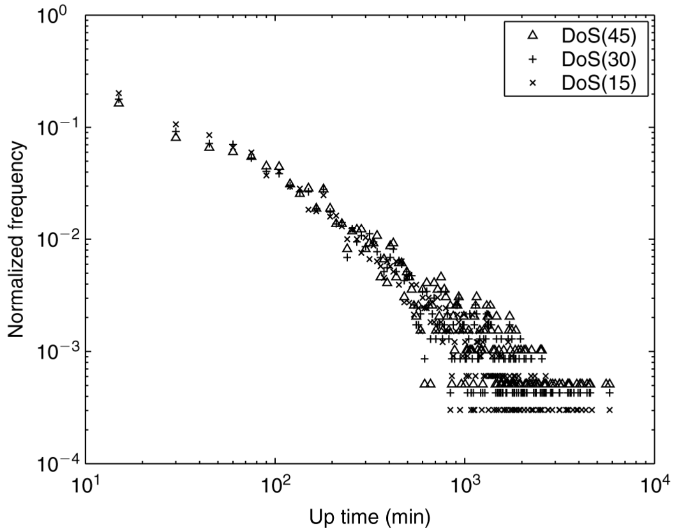

Figure 3. Lifetime distribution directly measured from Gnutella for 100 h. DoS(15): A node is assumed departed if no reply has been received for two consecutive trials. DoS(30): A node is assumed departed if no reply has been received for three consecutive trials. DoS(45): A node is assumed departed if no reply has been received for four consecutive trials.

- A new node (say X) will randomly select a node (say Y) from the network. One connection will be made to Y. Let Ω_Y be the number of neighbour nodes to which Y connects. Another $(\max\{\Omega_Y, m-1\})$ connections will be made between X and the neighbour nodes of Y.
- Each online node (including old nodes and new nodes) will be assigned with a random number generated in accordance with the lifetime distribution. The random number will determine whether the node has to be offline or online.

It should be noted that this Gnutella-like model is essentially an approximated model for Gnutella. It resembles the case that each new node makes a connection only to TTL 2 at most, i.e., by sending PING TTL 2.

4.1 Simulation

The simulation starts with 100 fully connected nodes. In every time step, 25 new nodes will be added. Each new node will make at most 20 connections (i.e., $m = 20$). The lifetime of an online node is recorded by a counter. After all the new connections have been made, all online nodes will be assigned with a random number that is uniformly selected from $[0, 1]$. Let r_i be the random number assigned to node i, and its corresponding lifetime is t_i. The ith node will be alive only if the following conditions hold.

$$\begin{aligned} 1 - F_l(t_i) &> r_i \quad \text{for } t_i = 1 \\ \frac{1 - F_l(t_i + 1)}{1 - F_l(t_i)} &> r_i \quad \text{for } t_i > 1 \end{aligned} \tag{3}$$

Here[1]

$$F_l(t) = \int_t^\infty P_l(t)dt$$

Once a node is "out", all its connections will be removed. The lifetime of each survival node is then incremented by 1.

Figs. 4 and 5 show the change of the number of online nodes and the node degree subtotal over time. These figures indicate when the network is in the transient state and when is in the stationary state. From observing the figures, it is found that the network is in the stationary state when after 5,000 steps. There are about 1,900 online peer nodes with 14,000 edges.

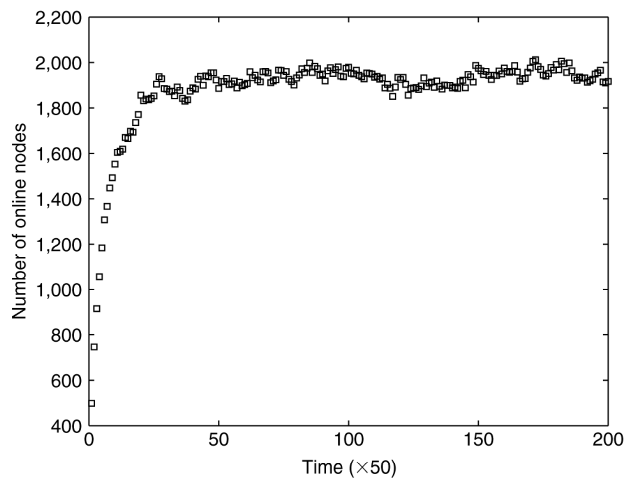

Figure 4. Total number of online nodes against time for the simulated Gnutella.

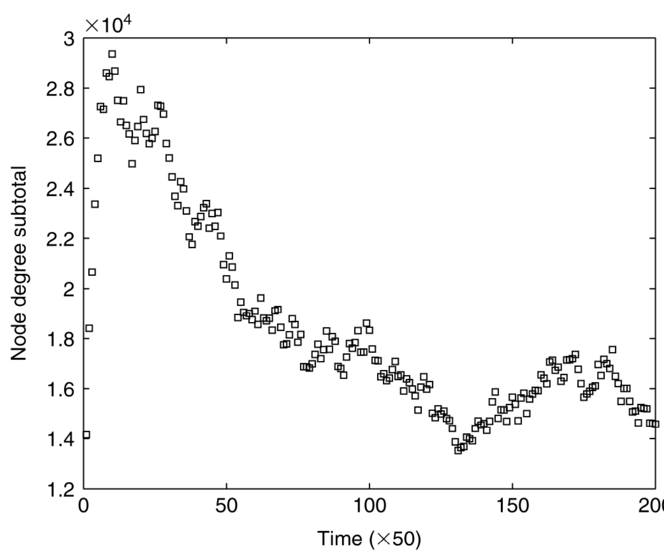

Figure 5. Node degree subtotal against time for the simulated Gnutella.

During the course of simulation, node degree distributions were also measured at times $t = 1{,}500$, 1,000, 2,500,

[1] In our simulation, we calculate this value by $F_l(t) = \sum_{t' \geq t} P_l(t')$.

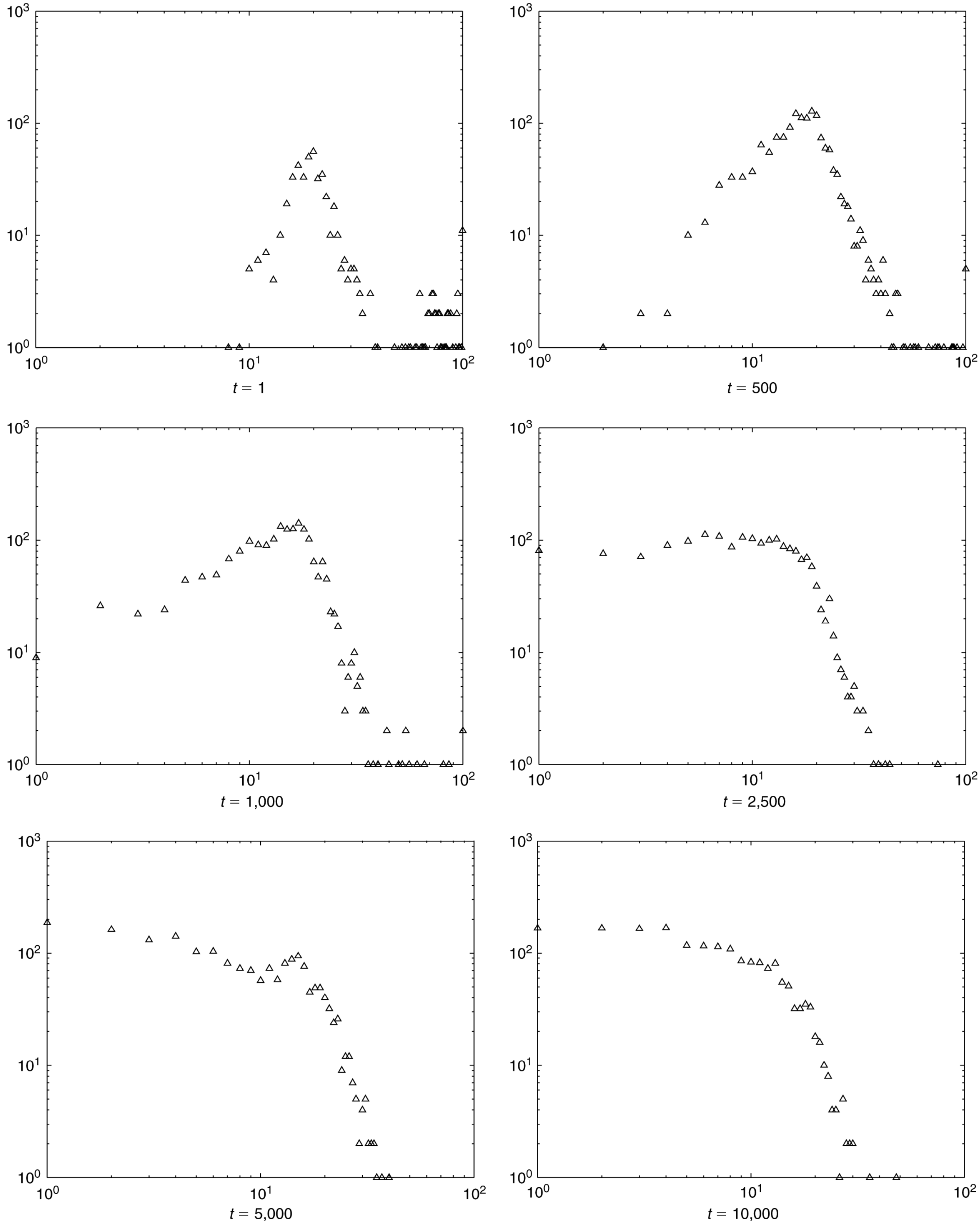

Figure 6. Node degree distributions measured at different times for the simulated Gnutella.

5,000 and 10,000, as shown in Fig. 6. From observing the figures, one can note that the distribution formation can be divided into two phases. In the early phase, the shape of the tail (i.e., node degree larger than 20) appears but the shape of the head is still evolving. Until time t around 5,000, the shape of the head appears and the shape of the distribution is maintained.

4.2 Connectedness

To investigate the connectedness behaviour of the simulated model, we take two approaches. The first approach is based on the utilization of Molloy–Reed criteria. The other is based on a virus outbreak simulation.

4.2.1 Molloy–Reed Criteria

In their paper, Molloy and Reed [15] have shown that the size of the giant component of a graph can be determined by the following factor:

$$Q = \sum_{k \geq 0} k(k-2)\lambda_k \tag{4}$$

where λ_k is the probability that a node is of k node degree. Let n be the total number of nodes. Molloy and Reed have shown the following theorem.

Theorem 1. (Molloy–Reed [15]) *If $Q > 0$, the random graph with high probability has a giant component and the size of the giant component is of order $\mathcal{O}(n)$. If $Q < 0$, the random graph with high probability has no giant component, and the graph consists of only small components of size $\mathcal{O}(\log n)$.*

In our simulated result, we have substituted the node degree distribution obtained as time $t = 5{,}000$ into (4) and the value is negative. Thus it implies that the network generated is, with high probability, a disconnected network.

4.2.2 Virus Outbreak Simulation

To confirm our anticipation, we repeat the simulation described in the last section by adding one more substep in Step 2 – a node will label itself as a virus corrupted node if any of its neighbour nodes is a virus corrupted node. To ensure that the network has been in its stationary state, the first virus corrupted node is introduced at time $t = 3{,}000$. The time setting is with reference to the simulation results depicted in Figs. 4–6.

Therefore, virus outbreak simulation is conducted in the following manner. In the very beginning, there is no virus node. The simulation is carried out in the same way as before. At $t = 3{,}000$, a node is randomly selected from the online node set and labelled as a virus-contaminated node. Then in every step after $t = 3{,}000$, each survival node will do one more substep in the Step 2 – it labels itself as a virus corrupted node if any of its neighbour nodes is a virus corrupted node. The change of the total number of online corrupted nodes are shown in Fig. 7. Once a node has been corrupted at time $t = 3{,}000$, the virus quickly spreads throughout the network. However, about 50 online nodes have been corrupted till the end of the simulation. The number of far below than the total number of online nodes. Only one thing can explain this phenomena: the network is disconnected. Therefore, the virus can only spread throughout the component in which it is located.

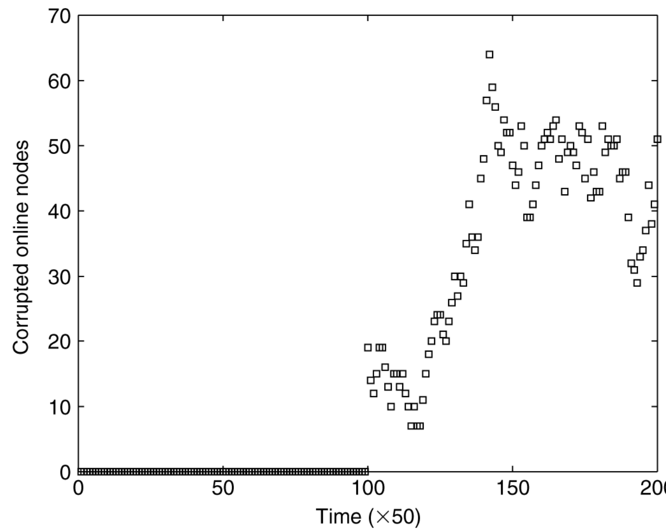

Figure 7. The number of corrupted nodes over time obtained for the simulated Gnutella.

5. Improved Connection Mechanism

The generic idea of this novel connection mechanism is very simple. Here, the number of seeds is 2. Whenever a new node has been added to the network, it will randomly select 2 online nodes.

The network generation algorithm can be described as follows. Suppose that each new node needs to make at most $2m$ new connections.

1. s nodes are fully connected initially.
2. At time t,
   - The lifetime of each existing node is incremented by 1.
   - n new nodes are added to the network with a lifetime set to 1.
   - A new node (say A) will randomly select two nodes (say B and C) from the network. Let Ω_B and Ω_C be the number of neighbour nodes for B and C, respectively.
     - One connection will be made to B. Another connection will be made to C.
     - $\max\{\Omega_B, m-1\}$ new connections will be made between A and the neighbour nodes of B.
     - $\max\{\Omega_C, m-1\}$ new connections will be made between A and the neighbour nodes of C.
   - Each online node (including old nodes and new nodes) will be assigned with a random number generated in accordance with the lifetime distribution. The random number will determine whether the node has to be offline or online.
3. Repeat Step 2.

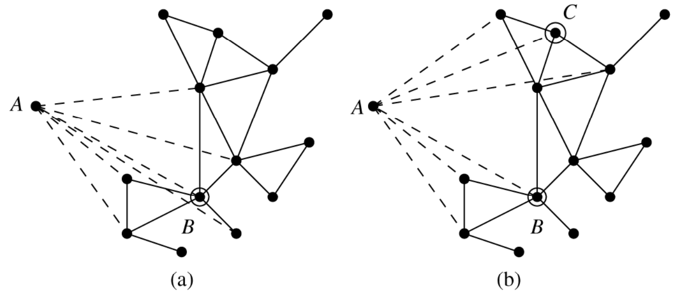

Figure 8. Connection mechanisms for the original model (a) and the improved model (b) when a new node A has been logged on. The maximum number of connections m that a new node can make is 6.

The bottom panel of Fig. 8 shows the idea of the Step 2. Every time a new peer wants to make a connection, it simply searches for online peers' IPs as the conventional Gnutella peer does. Again, the information about online IPs can be searched from a local cache or from a global server. Once a list of online peers' IPs have been collected, this new peer can randomly select two IPs from the list as seeds and then $XTRY$ these two IPs for their neighbors' IPs. The implementation is indeed quite simple.

5.1 Simulation

The simulation of the improved connection mechanism is almost the identical to the one presented in the previous section. Except that the number of seed nodes is 2. In every time step, 25 new nodes will be added. Each new node will randomly select 2 seed nodes and make connections.

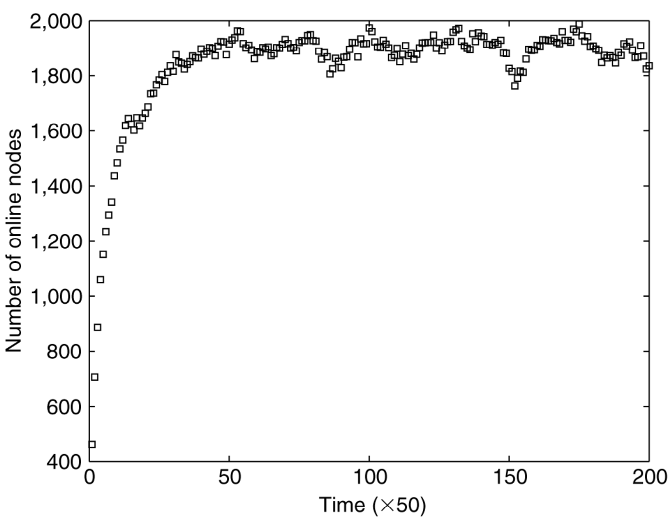

Figure 9. Total number of online nodes against time for the improved connection mechanism.

Figs. 9 and 10 show the number of online nodes and the node degree subtotal against time. From observing the figures, it is found that the network size is saturated after 5,000 steps. There are about 1,900 online peer nodes with 35,000 edges.

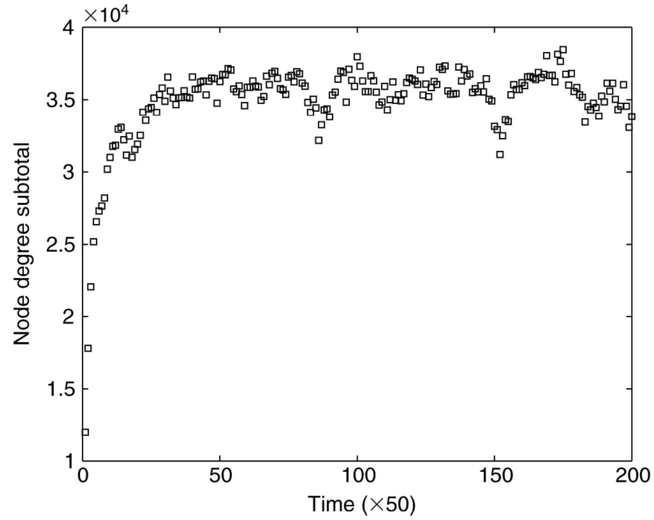

Figure 10. Node degree subtotal against time for the improved connection mechanism.

The node degree distributions measured at times $t = 1$, 500, 1,000, 2,500, 5,000 and 10,000 are shown in Fig. 11. For $t < 2{,}000$, the shapes of the distributions are similar to the one seed case. Until at t is larger than 2,500, the shape of the head appears and the shape of the distribution is maintained. The shape of the tail looks similar to that of Gnutella [11]. But the shape of the head is totally different. The peak of the distribution is around degree 9.

5.2 Connectedness

To investigate the connectedness behavior of the simulated model, we take two approaches. The first approach is based on the utilization of Molloy–Reed criteria. The other is based on a virus outbreak simulation.

5.2.1 Molloy–Reed Criteria

Substituting the node degree distribution obtained at time $t = 5{,}000$ into (4), it is found that the value of Q is positive. Thus it implies that the network being generated, is with high probability, a connected network.

5.2.2 Virus Outbreak Simulation

To confirm our anticipation, we repeat the simulation as described in Section 4.2. Except that the improved connection mechanism is used. The total number of online corrupted nodes is shown in Fig. 12. It is observed that total number of corrupted online nodes increases to the total number of online nodes, just a few time steps after a virus has been introduced. This shows that the network is connected. We have repeated the experiment several times and the same results obtained.

6. Discussion

To strengthen our argument, another control models have also been simulated and studied. In the control model, each new node *PINGs one seed node with TTL 3* instead

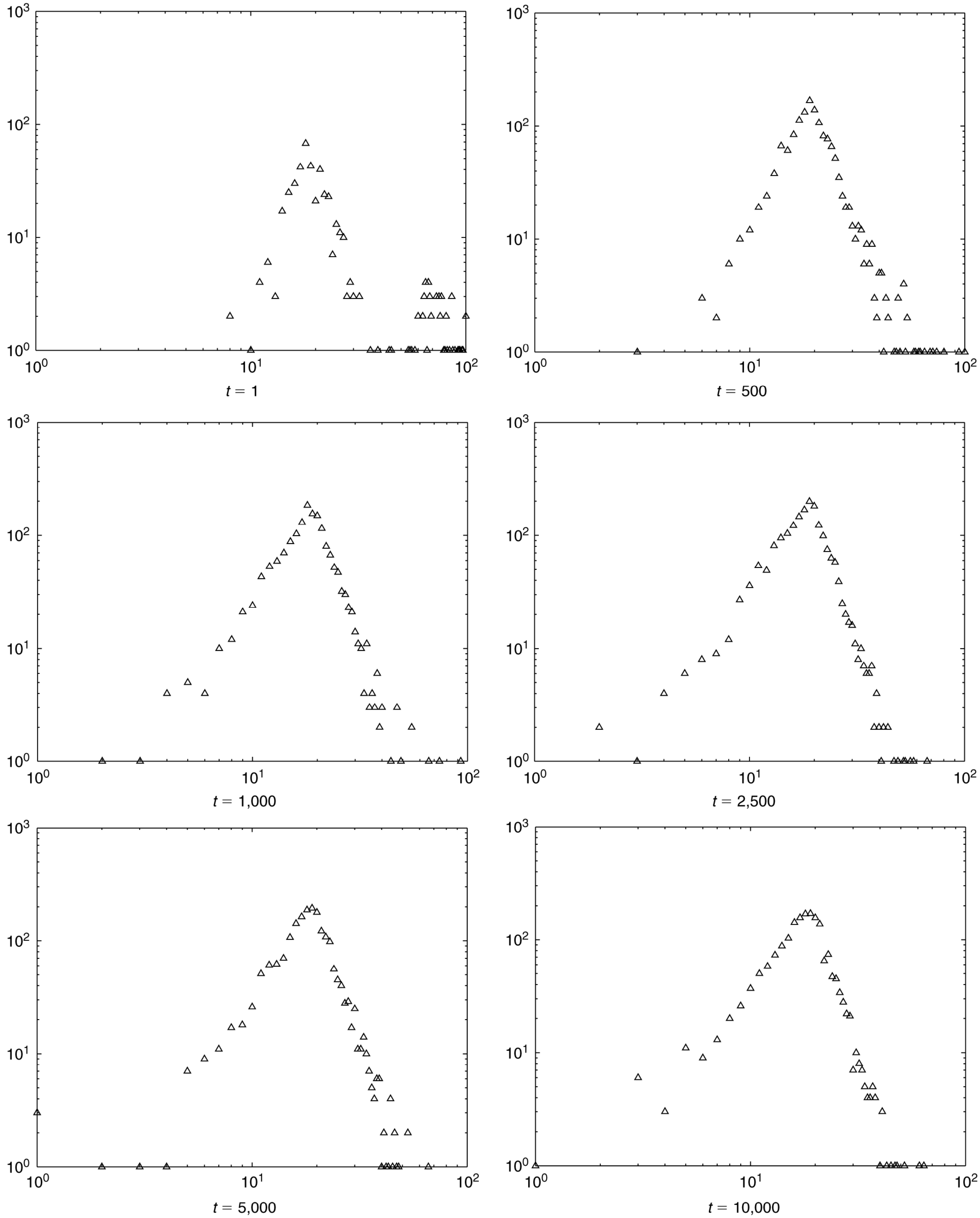

Figure 11. Node degree distributions measured at different time for the improved connection mechanism.

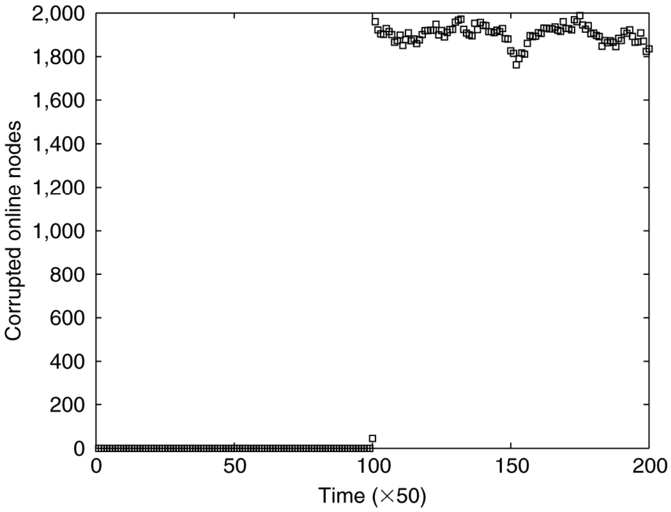

Figure 12. The number of corrupted nodes over time for the improved connection mechanism.

of TTL 2. That is to say, each PINGed node will reply with the *IP addresses of all its neighbours within 2 hops distance*. Observing the shape of the node degree distribution, it is found that the shape is quite different from those measured from Gnutella (Fig. 2). In Gnutella, the distribution is of decreasing slope. In this control model, the distribution is also single modal with the peak at node degree around m. It is different from the shape of the node degree distribution that is measured from the true Gnutella. As the success of Gnutella relies on how much resource a node can search. Disconnected network structure will definitely hamper the amount of resource a node can explore. In-depth investigation along this line will be valuable for future work.

7. Conclusion

In this paper, we have presented a study of the connectedness of Gnutella. As the actual network topology of Gnutella is unable to be obtained, a simplified model that is inspired from S.N. Dorogovtsev and J.F.F. Mendes [14] is used. In accordance with the node degree distribution and node lifetime distribution measured from Gnutella, the evolutionary behaviour of Gnutella has been modelled and simulated by computer simulation. By using the Molloy–Reed criteria and a virus outbreak simulation, it is also observed that the topology of the simulated Gnutella is not a connected graph. So, we suspect that Gnutella might not be a connected network. Finally, an improved connection mechanism has been proposed in Section 5. Each new node *PINGs two seed nodes with TTL 2* instead of PINGs only one seed node. Simulation results have shown that even the total number of new connections is 4, the network being generated can be a connected graph. This new result might shed the light for an efficient connection mechanism for future P2P or other *ad hoc* networks.

References

[1] D. Watts, *Small worlds* (Princeton University Press, 1999).

[2] M. Faloutsos, P. Faloutsos, & C. Faloutsos, On power-law relationships of the Internet topology, *Computer Communication Review*, *29*, 1999, 251.

[3] R. Albert, H. Jeong, & A.-L. Barabasi, Diameter of the World Wide Web, *Nature*, *401*, 1999, 130–131.

[4] M.A. Jovanovic, Modeling Large-scale Peer-to-Peer Networks and a Case Study of Gnutella, Master Thesis, Department of Electrical and Computer Engineering and Computer Science, University of Cincinnati, April 2001.

[5] S. Saroiu, P.K. Gummadi, & S.D. Gribble, A measurement study of peer-to-peer file sharing systems, *Proc. Multimedia Computing and Networking (MMCN)*, San Jose, January 2002.

[6] M. Ripeanu *et al.*, Mapping the Gnutella Network, *IEEE Internet Computing*, January/February 2002, 50–57.

[7] D. Stutzbach & R. Rejaie, Towards a better understanding of churns in peer-to-peer networks, Technical Report CIS-TR-04-06, Department of Computer Science, University of Oregon, November 2004.

[8] C. Xie & Y. Pan, Analysis of large-scale hybrid peer-to-peer network topology, *Proc. 49th Annual IEEE Global Communications Conference (GLOBECOM'06) San Francisco*, 2006.

[9] C. Xie, S. Guo, R. Rejaie, & Y. Pan, Examining graph properties of unstructured peer-to-peer pverlay topology, *Proc. 10th IEEE Global Internet Symposium (GI'07)*, May 2007, 11–12.

[10] Q. Lv *et al.*, Search and replication in unstructured peer-to-peer networks, *Proc. ACM ICS'02*, 2002.

[11] D.A. Menasce, Scalable P2P search, *IEEE Internet Computing*, March/April, 2003, 83–87.

[12] B. Yang & H. Garcia-Molina, Improving search in peer-to-peer networks, *Proc. ICDCS*, July 2002.

[13] R.H. Wouhaybi & A.T. Campbell, Phenix: Supporting resilient low-diameter peer-to-peer topologies, *INFOCOM'2004*, 2004, 108–119.

[14] S.N. Dorogovtsev & J.F.F. Mendes, *Evolution of networks* (Oxford University Press, 2003).

[15] M. Molloy & B.A. Reed, A critical point for random graphs with a given degree sequence, *Random Structures and Algorithms*, *6*, 1995, 161–180.

[16] Clip2 Distributed Search Services, The Gnutella protocol specification v0.4, http://dss.clip2.com, 2001.

[17] K. Ho, J. Wu, & J. Sum, On session lifetime distribution of Gnutella, *International Journal of Parallel, Emergent and Distributed Systems*, *23*(1), 2008, 1–15.

Biographies

John Sum received his B.Eng. degree in Electronic Engineering from Hong Kong Polytechnic University in 1992. Then, he received his M.Phil. and Ph.D. degrees in Computer Science and Engineering from Chinese University of Hong Kong in 1995 and 1998, respectively. In between 1998 and 2004, John had been teaching in several universities in Hong Kong, including the Hong Kong Baptist University, the Open University of Hong Kong and the Hong Kong Polytechnic University. In 2005, John moved to Taiwan and started to teach in the Chung Shan Medical University. Since August 2007, Dr. John Sum has been with the Institute of Electronic Commerce, the National Chung Hsing University, Taichung, ROC. His current research interests include neural computation, mobile sensor networks and scale-free network. John Sum is a Senior Member of IEEE and an Associate Editor of the *International Journal of Computers and Applications*.

Jie Wu is a Distinguished Professor at the Department of Computer Science and Engineering, Florida Atlantic University. He has published over 350 papers in various journals and conference proceedings. His research interests are in the areas of wireless networks and mobile computing, routing protocols, fault-tolerant computing and interconnection networks. Dr. Wu was on the editorial board of *IEEE Transactions on Parallel and Distributed Systems* and was a Co-guest-editor of *IEEE Computer and Journal of Parallel and Distributed Computing.* He served as the Program Co-chair for MASS 2004, Program Vice-chair for ICDCS 2001 and Program Vice-chair for ICPP 2000. He was also General Co-chair for MASS 2006 and is General Vice-chair for IPDPS 2007. He is the Author of the text *Distributed System Design* published by the CRC Press. He was also the recipient of the 1996–1997 and 2001–2002 Researcher of the Year Award at Florida Atlantic University. Dr. Wu has served as an IEEE Computer Society Distinguished Visitor and is the Chairman of IEEE Technical Committee on Distributed Processing (TCDP). He is a Member of ACM and a Senior Member of IEEE.

Chi-Sing Leung received his B.S. degree in Electronics from the Chinese University of Hong Kong in 1989. He received his M.Phil. degree in Information Engineering and Ph.D. degree in Computer Science from the same university in 1991 and 1995, respectively. He is currently an Associate Professor in the Department of Electronic Engineering, City University of Hong Kong. His research interests include neural computing, data mining and computer graphics. He has published over 60 international journal papers and 4 book chapters. He received the IEEE Transactions on Multimedia 2005 Prize Paper Award for the paper titled, "The Plenoptic Illumination Function" in *IEEE Transactions on Multimedia*, *4*(3), September 2002.

Kevin I.-J. Ho received his B.S. degree in Computer Engineering from National Chiao Tung University (Taiwan) in 1983. From 1985 to 1987, he was an Assistant Engineer of Institute of Information Industry, Taiwan. Then, he received his M.S. and Ph.D. degrees in Computer Science from University of Texas at Dallas in 1990 and 1992, respectively. Currently, he is an Associate Professor of Department of Computer Science and Communication Engineering, Providence University (Taiwan). His current research interests include image processing, algorithm design and analysis, scheduling theory, and computer networks.