Analysis of BitTorrent Networks

Daniel Kowalczyk, Leszek Koszalka
Department of Systems and Computer Networks,
Wroclaw University of Technology,
Wroclaw, Poland
E-mail: {daniel.kowalczyk, leszek.koszalka}@pwr.wroc.pl

***Abstract*—Peer-to-peer systems have emerged as an attractive alternative to client/server approaches. By efficiently leveraging the upload bandwidth of the end users, BitTorrent becomes a standard for scalable content distribution. In this paper, we concentrate on the overall performance of BitTorrent, in particular on impact of physical media distribution, such as fluctuations in cable and inside protocol parameter configuration. It is concluded that the decrease of the default optimistic un-choking time has a highly positive impact on the protocol performance. Moreover, it is shown that the delay of the client's network connection has also remarkable impact on the performance of the BitTorrent protocol.**

Keywords- network traffic; peer-to-peer network; network protocol; simulation experiment; efficiency.

I. INTRODUCTION

In the past few years we can observe growing popularity of peer-to-peer (P2P) networks. One of the peer-to-peer protocols is the BitTorrent (BT) [1], released ten years ago [2]. A recent analysis of the latest P2P trends word-wide shows that BT is still the most popular file-sharing protocol being recognized as the king of P2P traffic, because of generating approximately 45-78% of all P2P traffic, and 27-55% of all Internet traffic [3]. In our opinion, such a big popularity of this protocol is caused mainly by relatively simple architecture and so-called tit-for-tat mechanism. Tit-for-tat (TFT) policy [4] is used in BT to encourage each peer to upload to other peers while downloading [5]. Dongyu et al. [6] showed that with TFT a peer with a smaller upload rate will get slower download speed.

In this paper, we do make an attempt of some-aspects analysis of the BT protocol on the basis of results of simulation experiments. The simulation environment may have some advantages in comparison to the real network. Firstly, we can manage the peers in swarm parameters that in the real environment are random, e.g., the number of peers in swarm. Secondly, the whole experiment in the real environment would last days, weeks or even months [7], when using BIT-SIM it takes up to couple of minutes. Thirdly, in simulation environment we have an opportunity to create a proper scenario, proper bandwidth and type of internet connection that would be in real world also very time-consuming activity.

Our experimentation system was implemented in OMNeT++ simulation environment. This implementation was based on De Vogeleer, Erman, and Popescu ideas described initially in [8] and next, developed as BIT-SIM simulator [9]. After analysis of the obtained results of preliminary experiments, we stated the following research theses: (i) that optimistic un-choking session times have a big impact on overall BitTorrent performance, (ii) that link delay fluctuations may have a negative influence on BitTorrent downloading mechanism of efficiency.

The rest of the paper is organized as follows. Section II provides a review of simulators used for modeling and P2P network and BitTorrent file-sharing system. This review is based on the related works. In Section III the simulation environment is described, including models of network processes. In Section IV, we present the concept of research and an experiment design. The results of investigations are presented and discussed in Section V in correspondence to our research theses. Final remarks and some ideas for future research appear in Section VI.

II. RELATED WORKS

There are number of existing P2P simulators. In this section, we will overview some of these simulators. Traditional packet-level network simulators provide accurate low-level models of the network hardware and protocols but are too detailed to be effective in analysis of large scale P2P networks. For example, the ns-family environment, including the most widely used ns-2 simulator [10]. However, ns-2 and ns-3 simulators have weaknesses: (i) they are too detailed to be effective in analysis of large scale P2P networks, and (ii) they are very troublesome in adapting to P2P simulation because of the complexity and interdependency between simulation modules (scheduler, core simulator models, protocol models, and application level models). These disadvantages cause difficulties in attempt to extend the functionality of simulator with new models.

Many research teams have created their own overlay simulators. Some of them are used by experimenters, including PeerSim [11], P2PSim [12], OverSim, TOSim (Trust Overlay Simulator) [13]. Some of them are for only specific purposes and, thus, they are not efficient for general P2P protocol evaluation. NeuroGrid Simulator [14] is focused on simulating searches over content distribution network. Query-Cycle Simulator [15] is a cycle-based simulation framework for file-sharing P2P network. However, CANSimulator [16], FreeNet Simulator [17] may support network protocols, including BT protocol, either.

 ISBN: 978-1-61208-139-7

There is an existing implementation of BT in GPS (General P2P Simulator) environment [18]. Moreover, validation of this BT model was made by comparison it to a small scale network, also performance of the simulator is not so good, even the current version [19]. The other known implementation of BT protocol is in OMNET++ simulator [9]. This BT model was composed from three basic modules, including Tracker, Tracker Client, and Peer-wire. These modules correspond to the principles of BT actions.

The performances of many various P2P file sharing systems and BT system have been modeled in various ways. Qiu and Srikant [6] constructed a simple fluid model based on the Markov chains [20] describing the dynamics of the BT system. In [21], a statistical mathematical model is presented, which describes the evolution of BitTorrent.

In [22], Jun and Ahamad discussed the properties of the incentive mechanism of BT. Their analysis, based on the experimental results, showed that the original incentive mechanism of BT can induce free riding. They proposed a game theoretic framework that is more robust against free riders than the original mechanism.

Also several analytical studies of BT incentive mechanisms are presented in [23][24][25]. It was shown in [23] that BT mechanisms cannot prevent a systematic fairness through a set of simulations. Tian, Wu, and Ng [25] found that the standard tit-for-tat strategy cannot improve file availability. They proposed an innovative tit-for-tat strategy.

This brief review might show that various aspects of P2P networks and properties of BT protocol have been just discussed, analyzed and described in literature. The objective of this paper is checking the impact of some mechanisms on efficiency of BT, and making an attempt in finding some improvement of tit-for-tat mechanism.

III. SIMULATION ENVIRONMENT

The implementation of the BT protocol in OMNeT++ environment was based upon the mainline client version 4.0.2 presented in [26]. The mainline client is considered as the reference implementation of the BT protocol. The choking and rarest first algorithms are implemented just as they have been presented in [27].

The algorithms associated with BT, e.g., the peer selection and the piece selection can be implemented in different ways. Creating simulation environment, we take into consideration the following assumptions:

- All messages are responded immediately. Processing time for a message is zero (in simulation time), except for piece requests, which have a configurable response delay.
- Leecher starts downloading from another peer at the moment it is un-choked by that specific peer.
- A new block is requested immediately after a block has arrived from a peer, provided that the client is not choked in the meantime by that peer.
- Handshakes and bit-fields are exchanged without processing delay.
- Response delay is used only when handling piece requests.
- Piece selection algorithm can be configured from the default *.ini file.
- Swarm sizes are easy configurable by changing only one parameter in the default *.ini file.
- BitTorrent clients are created dynamically upon start-up of the simulation.
- Maximum swarm size of a simulation is not explicitly defined - it can be altered by configuring two parameters: the amount of clients connecting during the simulation run and the session inter-arrival time.

Many of the parameters of a BitTorrent swarm have not been previously considered that is why some assumptions regarding the input distributions of these parameters were necessary, including the exponential seeding time which was taken as proposed in [19].

IV. EXPERIMENTAL SETUP

The main part of experiment design – parameters used for the simulation scenarios - is presented in Table I.

TABLE I. SIMULATION PARAMETERS FOR SCENARIOS

Parameter	Value
Number of seeds	1
Number of peers	200
Max number of connections	200
Number of peers requested	50
Session inter-arrival time	Exponential, μ = 12,395s
Link delay	Uniform, [50, 400]ms
Bandwidth throughput	Asymmetric - 4Mbps/1Mbps
Asymmetric link	Yes
Initial piece distribution	0%
Request waiting time	Exponential, μ = 100ms
Block size	215 (32768)bytes
Simulation runs	20

TABLE II. PARAMETERS OF TORRENT FILES

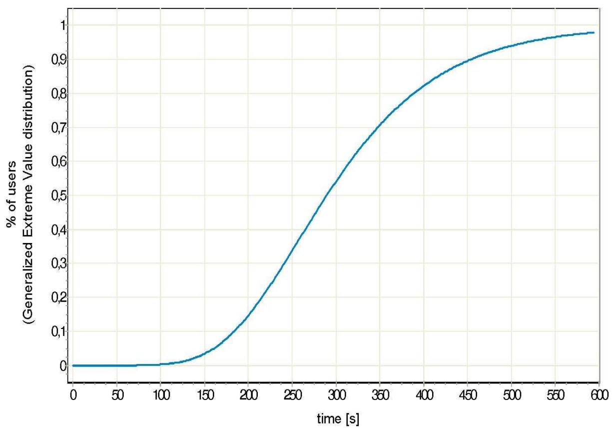
Parameter	Value
Number of pieces	1205
Piece size	62144 bytes
Download piece size	74825472 bytes

Choosing the number of peers equal to 200 and the number of repeating simulations per scenario equal to 20 represent an acceptable trade-off between simulation time and the number of resulting data to be obtained. The reason of taking a single seed is giving opportunities for starting to upload a new content in the network. The maximum number of connections equal to 200 is the sum of 199 peer connections and the single tracker connection. The value of the swarm inter-arrival time was selected at random from the results reported in [28]. The distribution parameters for link delay and bandwidth were selected to cover a large spread of possible link types. However, it may be observed that with this chosen data the session and message dynamics correspond to real-world environment. The initial piece distribution of each peer was chosen to be uniformly distributed. The parameter represents the ratio of pieces available at a joining peer when it enters the swarm (the initial seed always has all pieces). Which specific pieces are available is selected randomly. This was done to reflect the fact that the measurements reported in [28] were performed on swarms that already contained active peers already in possession of pieces of the content. Request waiting time, generated by different delay created processes, was taken as of exponential distribution with mean value equal to 100 [ms]. The block size of 32768 bytes was selected, following default size recommended in [26]. Remark: The experiment design does not take into consideration any information below the application layer, such as host names, IP addresses, or port numbers, thus, meta data required to join a swarm were proposed (Table II) for characteristic of the torrent file.

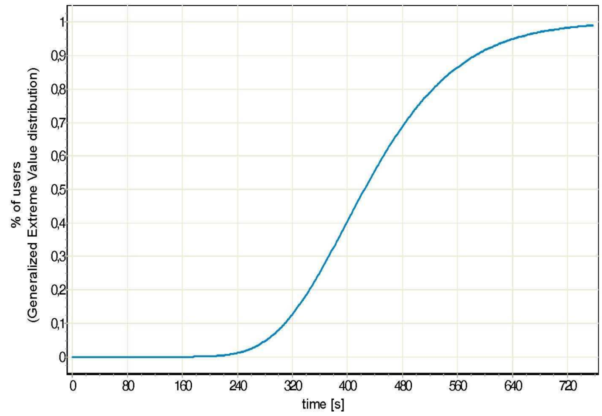
V. INVESTIGATIONS

A. *Link delay*

It is well-known that the throughput in TCP protocol depends strictly on the RTT - the elapsed time for transit of a signal over a closed circuit, or time elapsed for a message. Thus, if two TCP flows compete for the resources of the same bottleneck link, the connection with a smaller RTT can receive a higher bandwidth share than the other. Since a BT peer uploads to those peers from whom it downloads with high rates, peers on links with large delays may be characterized with the worse performance. To confirm this thesis we made the simulation experiment. The cumulative distribution functions for 20 runs are shown in Figure 1.



(a)



(b)

Figure 1. CDF for download time: (a) 10[ms] delay, (b) 100[ms] delay.

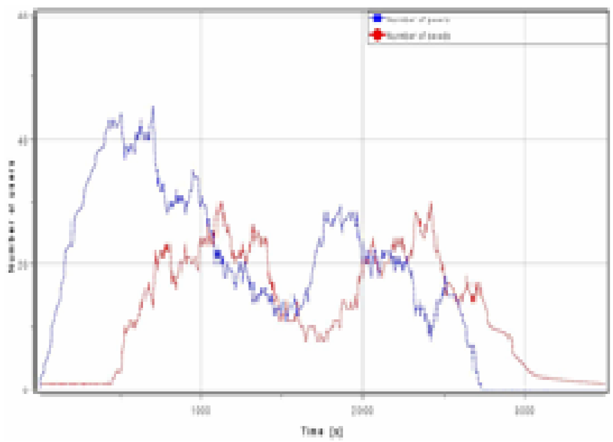
It may be observed that the download time for the case with 10 [ms] delay (Figure 1a) is of 290 [sec] and is less than 430 [sec] in the case with 100 [ms] delay (Figure 1b). Therefore, the mean download performance deteriorates by 33% for peers with greater delays.

It may be also observed, that the seeds and peers behaviors (see Figure 2) are different in relation to the delay. For delay of 10 [ms], the number of peers is changing in dynamical way. This is the result of the fast message exchange between users. In seed case two plots are almost identical and no major differences are observed.

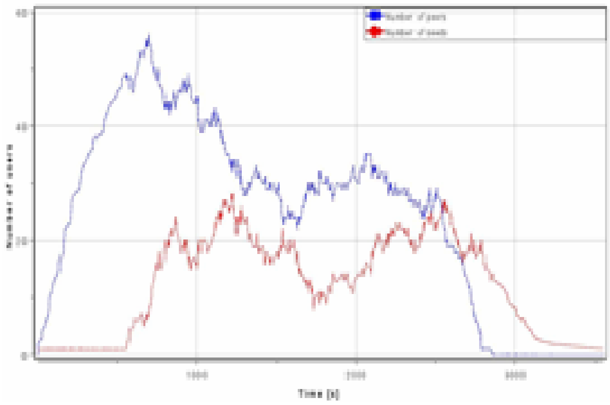
B. *Modified default tit-for-tat mechanism*

The second scenario provides some minor changes in default tit-for-tat mechanism, exactly in optimistic un-choking. The default un-choking times in optimistic un-choking mechanism is 30 [sec].

In order to check the relation of optimistic un-choking time to other parameters, the value of optimistic un-choking time was lowered to 10 [sec].



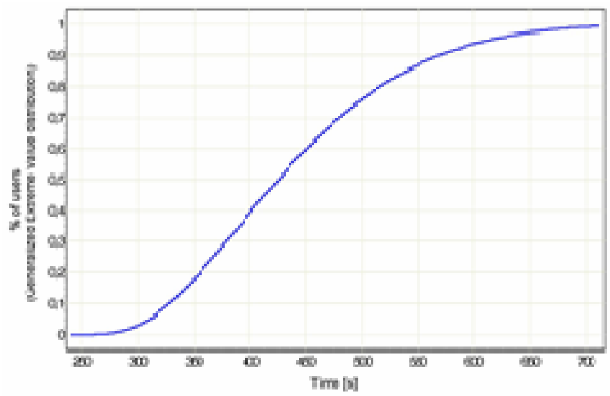
(a)



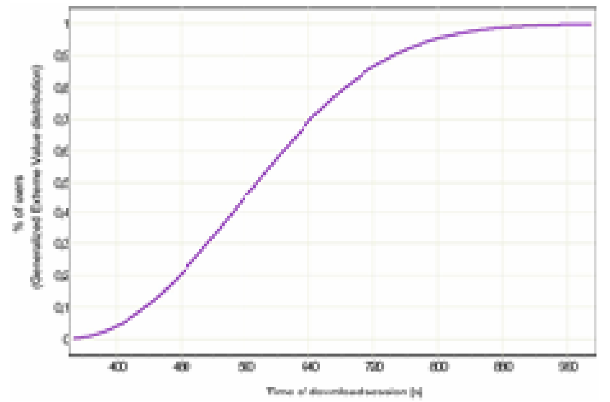
(b)

Figure 2. Seed vs peer session: (a) 10[ms] delay, (b) 100[ms] delay.

This parameter was not configurable from the default *.ini file in BIT-SIM simulator (some minor interference in source code was necessary).



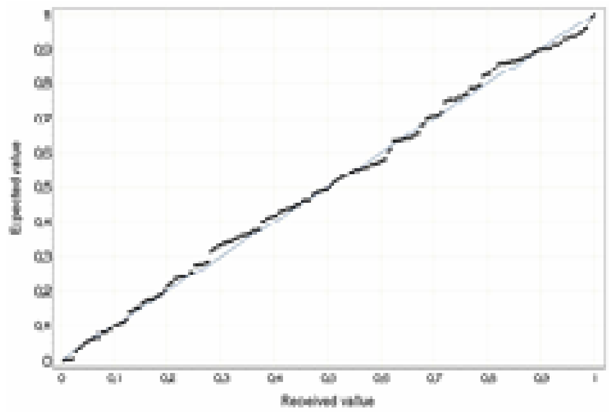
(a)



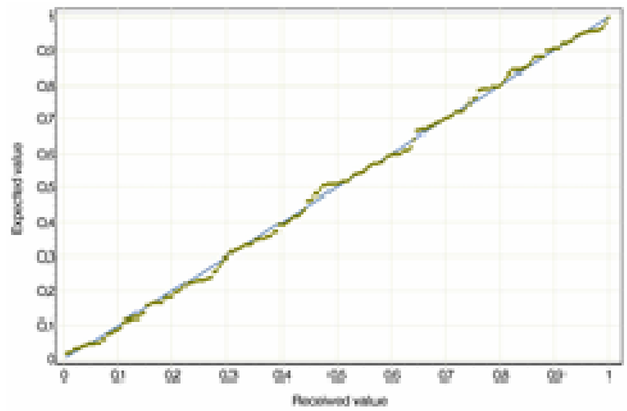
(b)

Figure 3. CDF for download time: (a) with tit-for-tat change, (b) without tit-for-tat change.

Measurement results are presented in Figure 3 and Figure 4. When we compare the plots from Figure 3(a) and 3(b), we can observe some minor improvement. The mean value of 50% for Figure 3(b) is equal to 580[sec] in turn in the optimistic un-choking mechanism with 10 [sec] un-choking times.



(a)



(b)

Figure 4. P-P plot for share ratio: (a) with tit-for-tat change, (b) without tit-for-tat change.

In Figure 3(a), the considered value is equal to 425 [sec], resulting in profit (improvement) of 27%. It may be observed that the share ratio value shown in Figure 4 is almost the same in two plots presented in 4(a) and in 4(b). Thanks to that, the change does not affect tit-for-tat in overall negative way.

C. Overall conclusion

In Table III, the main results of experiments have been gathered. In the table, in the fourth column, the 'change' is referring to percentage gain which resulted after changing the values of the considered parameters: *link delay* and *optimistic un-choking time* from the 'old' one to the 'new' one.

TABLE III. MAIN RESULTS OF EXPERIMENT

Parameter	Value		Change
	Old	New	
Link delay	10 [ms]	100 [ms]	33%
Optimistic unchoking time	30 [s]	10 [s]	27%

VI. FINAL REMARKS

In this paper, we consider the two theses formulated in Section I, concerned performance and behavior of BitTorrent protocol according to (i) its internal tit-for-tat mechanism changes and (ii) its external one like link delay (independent of protocol).

In the case (i), we confirmed that even small appropriate changes in the internal protocol (like change of optimistic un-choking time) can ensure more efficient data exchange. That is why some further investigations should be made with adjusted parameters inside BitTorrent mechanism configurations and using the idea of multistage experiments as proposed in [30].

In the case (ii) concerning link delay, we have shown that TCP protocol (which is used in BitTorrent) is not a most efficient which can be used. The link delay fluctuations can cause a visible protocol performance decrease. Improvement can be made by possible using the UDP protocol - to which the BitTorrent is migrating right now. The prototype of that can be treated as a kind of protocol already created in the µTorrent client and from the name of the client can be named µTP. Although all the BitTorrent protocol codes are open to the Internet community, the µTP protocol is closed for now.

It also should be mentioned that the simulator used in this paper, may be regarded as a useful tool for conducting experiments, however, in limited range because there are some functionalities not available such as modular peer selection and peer snubbing, (i.e., dropping peers that do not respond quickly enough), trackerless Distributed Hash Table (DHT) protocol, encryption and super seeding. Moreover, torrent file used in simulation does not use additional information like current download status, connected peers and QoS information. The investigations in the nearer future should take into consideration these aspects to allow making detailed analysis of BitTorrent efficiency.

REFERENCES

[1] B. Cohen, "Incentives build robustness in BitTorrent", Proceedings to 1st Workshop on Economics of Peer-to-Peer Systems, 2003.

[2] B. Cohen, "A new P2P application", Yahoo e-Groups, Posted February 2001, Retrieved 2007

[3] TorrentFreak. http://torrentfreak.com/bittorrent-still-king-of-p2p-traffic- 090218, February 2009.

[4] R. Thommes and M. Coates, "Deterministic packet marking for time-varying congestion price estimation", IEEE/ACM Trans. Networking, vol. 14, 2006, pp. 592-602.

[5] D. Guan, J. Wang, Y. Zhang, and J. Dong, "Understanding BitTorrent download performance", Proceedings to Int. Conf. on Networking, ICN'08, pp. 330-335.

[6] D. Qiu and R. Srikant, " Modeling and performance analysis of BitTorrent like peer-to-peer networks", Proceedings to IEEE SIGCOMM '04, doi:10.1145/1030194.1015508, USA, 2004.

[7] M. Izal, G. Urvoy-Keller, E. W. Biersack, P. Felber, A Al Hamra, and L. Garces-Erice, "Dissecting BitTorrent: five months in a Torrent's lifetime**",** , PAM 2004, Lecture Notes in Computer Science, vol. 3015, Springer 2004, pp. 1-11.

[8] K. De Vogeleer, D. Erman, and A. Popescu. "Simulating BitTorrent", Proceedings to International Conference SimuTools'08, Marseille, France, ISBN 978-963-9799-20-2, March 2008.

[9] BIT-SIM simulator. http://www.its.bth.se/staff/kdv/BIT-SIM/, June 2009.

[10] Berkeley/LNBL/ISI, "The ns-2 simulator", http://www.isi.edu/nsnam/ns/, 2007.

[11] A. Montresor and M. Jelasity, "PeerSim: A scalable P2P simulator", Peer-to-Peer Computing, 2009, pp. 99-100.

[12] P2Psim. http://pdos.csail.mit.edu/p2psim/, 2009.

[13] W. Wang and G. Zeng,"A Generic trust overlay simulator for P2P networks", Proceedings to PRDC'06, December 2006, pp. 401-402.

[14] NeuroGrid. http://www.neurogrid.net/, 2009.

[15] T. Condie, S. D. Kamvar, H. Garcia-Molina, "Adaptive peer-to-peer topologies", Peer-to-Peer Computing, 2004, pp. 53-62.

[16] CANSimulator. http://sourceforge.net/projects/cansimulator.

[17] J. Pfeifer,"Freenet casching algorithms under high load", http://www.cs.usask.ca/classes/498/t1/898/W7/P2/freenet, 2009.

[18] W. Yang and N. B. Abu-Ghazaleh, "GPS: A general Peer-to-Peer simulator and its use for modeling BitTorrent." Proceedings to MASCOTS'2005, 2005, pp. 425- 434.

[19] GPS simulator. http://www.cs.binghamton.edu/, 2009.

[20] X. Yang and G. de Veciana, "Performance of peer-to-peer networks: service capacity and role of resources sharing policies", Journal Performance Evaluation, vol. 63, March 2006.

[21] Z. Liu and C. Chen, "Modeling fetch-at-most-once behavior in peer-to-peer file-sharing systems. Proceedings to APWeb Workshop, 2006, pp. 717-724.

[22] S. Jun and M. Ahamad, " Incentives in BitTorrent induce free riding", Proceedings to ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems, USA, ACM Press, August 2005, pp. 116-121.

[23] A. R. Bharambe, C. Herley, and V. N. Padmanabhan, "Analyzing and improving BitTorrent performance", Technical Report MSR-TR-2005-03, Microsoft Research, Redmond, WA, February 2005.

[24] K. Csernai, M. Jelasity, J. A. Pouwelse,, and T. Vinko, ":Modeling un-connectable peers in private BitTorrent communities", Proceedings to PDP'11, 2011, pp. 582-589.

[25] Y. Tian, D. Wu, and K.-W. Ng, "Modeling, analysis and improvement for BitTorrent-like file sharing networks", Proceedings IEEE INFOCOM 2006, Barcelona, Spain, 2006.

[26] BitTorrent, Inc. Bittorrent. http:www.bittorrent.com.BitTorrent, 2009.

[27] A. Legout, G. Urvoy-Keller, and P. Michiardi. "Rarest first and choke algorithms are enough", Proceedings to ACM SIGCOMM/USENIX ICM, October 2006.

[28] D. Erman, D. Saavedra, J. A. Sanchez-Gonzales, and A. Popescu, "Validating BitTorrent models", Telecommunications Systems, vol. 39, 2008, pp. 103-116.

[29] S. Tewari, L. Kleinrock, "Optimal search performance in unstructured peer-to-peer networks with clustered demands", IEEE Journal on Selected Areas in Communications, vol. 25, 2007, pp. 84-95.

[30] D. Ohia, L. Koszalka, and A. Kasprzak, "Evolutionary algorithm for congestion problem in computer networks", Springer, Lecture Notes in Artificial Intelligence, vol. 5711, 2009, pp. 113-122.

 ISBN: 978-1-61208-139-7