



Article A Bit Torrent Traffic Optimization Method for Enhancing the Stability of Network Traffic

Deguang Li^{1,*}, Yongxin Zhang¹, Shijie Jia¹, Dong Liu¹, Yanling Jin¹ and Yuansheng Wu²

- ¹ School of Information Technology, Luoyang Normal University, Luoyang 471934, China; zyx@lynu.edu.cn (Y.Z.); jiashijie@lynu.edu.cn (S.J.); liudong@lynu.edu.cn (D.L.); jinyanling@lynu.edu.cn (Y.J.)
- ² College of Computer Science, Sichuan University, Chengdu 610065, China; wys_lidg@sina.com
- * Correspondence: lideguang@lynu.edu.cn; Tel.: +86-186-3927-7037

Received: 11 October 2019; Accepted: 19 November 2019; Published: 20 November 2019



Abstract: With the extensive use of peer-to-peer applications in recent years, the network traffic becomes more dynamic and less predictable, which leads to the decline of network resource utilization and the degradation of network performance. Aiming towards the above problems, we explore how to strengthen the cooperation between peer-to-peer applications and networks, making the application adjust its own traffic mode according to current network traffic status to enhance the stability of network traffic. We improve two key algorithms of peer selection and choking/unchoking in the protocol and introduce traffic relaxation to characterize traffic state while taking the current most popular peer-to-peer application (bit torrent protocol) as an example. In our improved method, peers are selected probabilistically according their traffic relaxation and transfer rate of peers is also solved. Finally, we conduct simulation experiments in two real network typologies with real traffic matrix data and different sizes of bit torrent swarms; the experimental results show that our method can significantly improve the stability of the network traffic without sacrificing or even improving the performance of the bit torrent protocol when compared with original BT protocol.

Keywords: traffic optimization; bit torrent; traffic stability; traffic relaxation; peer selection; choking/unchoking

1. Introduction

With the rapid development of 4G/5G network and intelligent mobile terminals, the number of mobile Internet users worldwide has reached 3.3 billion and the total number of Internet users worldwide has reached 4.1 billion up to now, and it is keeping rapid and sustained growth [1]. The computer network has entered a stage of diversified development, various applications, such as big data, cloud computing, intelligent internet of things, live video broadcasting, online games, e-commerce, and social networks are booming, which results in rapid growth of internet traffic. It is expected that, by 2020, Internet traffic will increase by 50 times, and the exchange capacity of Internet core nodes will increase from Tb/s to Pb/s [2]. With the increasing variety of network applications and the increasingly complex demand for network resources, the idea that is envisaged at the beginning that the Internet only needs to guarantee that the connectivity and reliability of the network can no longer meet the higher real-time and stability requirements of these emerging businesses.

Peer-to-Peer (P2P) network with its characteristics of node equivalence, resource dispersion, topological dynamics, centrality-free dependence, real-time, expansible, and high utilization rate of resources rapidly develops [3,4]; its traffic has become an important part of Internet traffic, and research shows that P2P traffic has consumed more than 60% of network bandwidth [3]. As the number of P2P

users and the availability of resources are all dynamic changes, there are users constantly arriving and leaving, and the state of resources owned by users is also constantly changing. As a result, the core network traffic is adversely affected, which makes the network traffic more dynamic and less predictable. Adding P2P network-to-network traffic management, especially enhancing the stability of network traffic, reducing the link congestion and bandwidth problems that are caused by P2P traffic, helping network managers to control and schedule the bandwidth of network [5,6], effectively improving user experience of data services has become an urgent problem for network managers.

The above problems are caused by the constant changeability of the application layer and increasing demands on the network traffic of data services. However, the development of network layer is not compatible and flexible to keep up with the pace. For such situation, which requires network and application to both make more effective and adaptable decisions, but only relying on the network or application is not enough. Existing P2P traffic optimization work is mainly divided into three aspects. (1) Reverse engineering technology. P2P applications use reverse engineering technology to infer the topology and the state of the network to make the constructed overlay network topology match the underlying network topology. (2) P2P cache. Internet service provider (ISP) establishes the P2P cache to reduce the inter-network (or inter-domain) traffic and traffic fee. (3) Application Layer Traffic Optimization. This method is a cooperative solution for the application layer and the network layer; the network actively provides its state information to the P2P application in the upper layer, and the P2P application uses the information to adjust its behavior to optimize the performance of the network and the application. While all of these works doe not consider the stability of network traffic, in our method, we first use the state information of the network to actively optimize the P2P traffic mode to achieve the purpose of enhancing traffic stability, and then we improve two key algorithms of the bit torrent (BT) protocol to make it actively perceive the traffic state of the network and adjust the generated BT traffic mode.

Based on the comparison between previous work and our work, we know that information exchange between the network and the application is very important; our idea is also consistent with software defined network (SDN) [7,8]. SDN has three main characteristics when compared with traditional network: (1) Control and forwarding separation. Decoupling and separation of control plane and data plane of the network. (2) Centralized control in logic. The controller uniformly collects and manages distributed network state information [9]; the control plane in the set can control multiple forwarding surface devices, that is, the whole physical network. (3) Open interface between control plane and forwarding plane [10]. SDN provides an open programmable interface for the control plane. In this way, the application only needs to pay attention to its own logic, and it does not need to pay attention to the underlying implementation details. The northbound interface of SDN is a platform for realizing the interaction of state information between the network and the application; the application can actively inform the network of its resource requirements and status information, and the network can also pass the northbound interface and transmit information, such as the network topology and the link state to the application, so that the cooperation between the application layer and the network layer is realized, so the northbound interface is a set of APIs that allow for developers to create software for a given platform. Vice versa, the southbound interface is a set of APIs that allow that code to be instantiated on the physical hardware. In short, the southbound interface to network devices and the northbound interface to applications. Mostafaei [11,12] designed an enhanced release of the widely used Netkit network emulator for experimenting with SDN enabled networks, and they proposed SDNS [12], which is an SDN-system that opens the application fields of federated networks to federated connectivity services, which all prove the applicability and scalability of SDN.

We try to explore how to enhance cooperation between P2P application and the network in this paper based on the idea of information exchange between the network and the application brought by SDN, so that P2P application can actively adjust its own traffic mode according to the current network traffic status to enhance the stability of network traffic. We make BT protocol while actively considering the current network traffic status to adjust its traffic mode by improving two key algorithms of peer

selection and choking/unchoking in the protocol and introducing traffic relaxation to characterize traffic state. Finally, we use real traffic matrix data to simulate BT groups of different sizes in two real network typologies, and the experimental results show that, when compared with the original BT protocol, our improved BT protocol can significantly enhance the stability of network traffic matrix without sacrificing or even increasing the performance of BT protocol. Thus, our contributions in this paper are:

- (1) The concept of traffic relaxation to describe traffic state of network is introduced and used in our improved algorithm.
- (2) Peers are selected probabilistically according to the relative magnitude of each peer's apportioned traffic relaxation in the BT group when selecting peer.
- (3) The allocation traffic relaxation and upload rate are simultaneously considered when choosing peer to be choked or unchoked.
- (4) We improve the adaptability of the general peer-to-peer simulator (GPS) and realize the simulation of our improved BT protocol.

The organization of this paper is as follows. Section 2 presents the related work of BT traffic optimization strategy of different researchers, and a detailed description of our method for traffic optimization, including peer selection algorithm and blocking/unblocking algorithm, is presented in Section 3. In Section 4, we evaluate the performance of our improved BT protocol and the original BT protocol by detail experiments, and the conclusion is outlined in Section 5.

2. Related Work

Many researches have shown that Internet traffic has long correlation and burstiness. However, Internet core network traffic shows a strong circadian pattern and it has atypical periodic stability characteristics (the cycle is 24 h) [13]. The reasons are mainly from two aspects: first, with the increase of network traffic, the result of statistical reuse reduces the sudden decrease of Internet traffic and enhances the stability of Internet traffic [14]. Second, the traditional Internet is based on web traffic, and the web users have relatively stable daily rest and use mode. For the periodic stability characteristics of the core network traffic in previous work [15,16], we optimized the energy consumption for each traffic in a stationary period (a typical stationary period of the core network traffic is 1–1.5 h [17]), that is: given a peak traffic matrix of each traffic in a stationary period, study how to allocate the network resources and conduct a routing service to improve the utilization of network resources, try to make many network resources as idle as possible, and then use muti-granularity modular sleep technology to reduce the energy consumption of the network. Obviously, during each traffic stagnation period, the more stable the network traffic is, the stronger the predictability of the traffic. The smaller the difference between the peak traffic matrix and the average traffic matrix is, the higher the resource utilization rate after traffic optimization is, and the better effect of the performance of the network.

However, due to the widespread use of P2P applications in recent years, P2P traffic has become an important part of Internet traffic. Users are constantly arriving and leaving, and the state of resources owned by users is also constantly changing since the number of P2P users and resource availability are dynamically changing. As a result, the core network traffic is adversely affected, which makes the network traffic more dynamic and less predictable. The increasing dynamic traffic leads to the increase of the difference between the peak traffic matrix and the average traffic matrix in a stationary period, the decrease of the utilization rate of network resources, and the deterioration of the performance of network. In addition, the network planning and the traffic engineering all require relatively stable network traffic, and the unpredictable traffic will seriously degrade the network performance, make the network performance degradation, and the failure of the traffic engineering [18,19]. Therefore, enhancing the stability of network traffic has an important effect on network planning and traffic engineering, and it can significantly improve the resource utilization of the network.

Existing P2P traffic optimization work is mainly divided into three aspects to solve the problems caused by the application of P2P.

(1) Reverse engineering technology. P2P applications use reverse engineering technology to infer the topology and state of the network to make the constructed overlay network topology match the underlying network topology. For example, Liu [20] proposed a location-aware topology matching (LTM) technique, which builds an efficient overlay by disconnecting the slow connections and choosing physically closer nodes as logical neighbors, while still retaining the search scope and reducing the response time for queries. Ren [21] presented a topology-aware BT system, which discovers network proximate (to connected peers) actively and it uses both proximate and transmission rates to maintain fast downloading while also reducing the transmitting distance of the BT traffic and the Internet traffic, experiments showed that the proposed method could reduce approximately 25% download traffic while achieving a 15% faster download speed as compared to several prevalent BT clients on average.

(2) P2P cache. Internet service provider (ISP) establishes the P2P cache to reduce inter-network (or inter-domain) traffic and traffic fee, as shown in references [22–24]. Zhou [22] noticed that how to efficiently use network resources to reduce the traffic burden on the ISP was crucial in the sustainable development of P2P systems, and the comparison of P2P cache, locality-awareness, and data scheduling for P2P traffic optimization were also discussed. Hefeeda [23] pointed out that the popularity distribution of P2P objects cannot be accurately captured while using Zipf-like distribution, thus they developed a novel-caching algorithm for P2P traffic that was based on object segmentation, proportional partial admission, and eviction of objects. By using the trace-driven simulation, the result indicated that partial caching was crucial to the performance of P2P caching algorithms in P2P systems. Aggarwal [24] proposed using an oracle hosted by the ISP, so that ISP and P2P users can cooperate for improved performance. Such an oracle can be queried by P2P nodes while choosing neighbors or while deciding from whom to download content and it will rank the possible neighbors of the querying node according to a locality indication.

(3) Application Layer Traffic Optimization (ALTO) [25–27]. This method is a cooperative solution for the application layer and the network layer; the network actively provides its state information to the P2P application in the upper layer and the P2P application uses the information to adjust its behavior to optimize the performance of the network and the application. Bruni [25] explored the techniques that aimed to optimize the distribution of contents in the BT protocol by limiting the interactions between the peers to a circumscribed area, and the application of their locality-aware choking algorithm allows for obtaining both a significant reduction of inter-AS traffic and average download rates similar to those that were obtained while using a non-optimized version of the choking algorithm. Gurbani [26] argued that it is a beneficial solution to the ALTO problem to involve cross-layer cooperation, allowing for communications between applications and network elements aware of the underlying network topology. In particular, a look-up mechanism to be used by applications to discover the appropriate network elements to query for obtaining topology information that they need for ALTO, and a protocol to be used in communications between applications and networks, these two elements are very important for the cross-layer cooperation. Madhyastha [27] pointed out that the way to solve the ALTO problem is to build distributed application-layer services for location and path selection to enable peers to estimate their position in the network and efficiently select their neighbors, similar solutions have been embedded into P2P applications, such as Azureus.

The third method is consistent with the idea of opening interface between the control plane and forwarding plane proposed by SDN, that is, information exchange between the network layer and the application layer to achieve cooperation between these two layers. The greatest value that is brought by SDN is to improve the utilization efficiency of the whole network resources, enhance the network virtualization capability and accelerate the network innovation. The application obtains network information through the open northbound interface, adopts software algorithm and network resource scheduling to improve the utilization and network quality of the whole network, and opens the capacity of virtual network configuration to users. However, existing work does not consider

enhancing the stability of network traffic while improving the network performance. The network traffic model plays an important role in the design, planning, and performance analysis of computer networks. The instability of the traffic will adversely affect the establishment of the traffic model, so it is necessary to smooth the traffic to make it smooth or near stable.

Inspired by the cooperation between application layer and network layer of SDN, this paper proposes a BT traffic optimization method for enhancing the stability of network traffic, that is, P2P applications use the state information of the network to actively optimize the P2P traffic mode to achieve the purpose of enhancing traffic stability. While taking the BT protocol as an example, two key algorithms of the BT protocol are improved to make it actively perceive the traffic state of the network and adjust the generated BT traffic mode, and the concept of traffic relaxation is also introduced to describe the traffic state of the network. When selecting a peer in our algorithm, the neighbor peer is selected according to the probability of each peer traffic relaxation in the BT group. The two-criterion selection problem that considers the traffic relaxation and transfer rate of peers simultaneously is also solved in the process of choking/unchoking. The experiments show that our improved BT protocol can significantly improve the stability of network traffic as compared with the original BT protocol, and improve the network traffic pattern, traffic distribution, and predictability of traffic development, also reducing the difference between peak traffic matrix and average traffic matrix of the network, so the resource utilization of the network is also changed.

3. Our New BT Traffic Optimization Strategy

According to the idea of SDN, the network and BT protocol can cooperate with each other. The network provides the state information of the network to the BT protocol, and the BT protocol considers the state information of the network in peer selection and choking/unchoking. In this way, the BT protocol and network performance can be simultaneously optimized. The network state that is considered in this paper includes network topology and traffic state. Briefly, we call a stationary traffic period a section. For each section, resource allocation and service routing of the network are fixed, and the traffic of each link is determined by the service requirements of the network (the traffic matrix). Therefore, the goal of our improved BT protocol in this paper is to enhance the stability of traffic matrix in a section.

3.1. Peer Selection Algorithm

The BT protocol randomly selects a certain number of peers to establish connections, and then exchanges the fragments of the interest files to each other. We add network state in peer selection to improve the peer selection algorithm. We introduce the concept of traffic relaxation to describe traffic state of the network for the convenience of our description.

Definition 1. For a core network node *i* at a given time *t*, the traffic relaxation *r* between node *i* and node *j* is defined as the difference between m_j and c_j , that is $r_j = m_j - c_j$, m_j is the peak value of the service demand in the section and c_j is the current value of the service demand in the section.

The peak traffic demand m_j can be considered as known, and the current value of business requirements c_j can be measured in real time since the network resources in each section are configured and calculated according to the peak traffic matrix of the section. Our goal is to keep the traffic relaxation of all business requirements in the network within a moderate range, avoiding situations where the traffic relaxation is too small and too large.

We introduce our basic idea of peer selection algorithm while considering the network state by Figure 1. As shown in the figure, if peer 1 chooses peer 5 as its neighbor, then it is likely that they will share existing file fragments with each other, thus increasing the business requirements between routers A and C. If the traffic relaxation of router A to C is smaller than that of router A to other routers, then peer 1 should try not to choose the peer under router C, because it is possible to increase

traffic between A and C, and the traffic relaxation between A and C is further reduced. Therefore, peer 1 should select the peer under the router with a large amount of traffic relaxation to establish a connection. We need the traffic relaxation of peer 1 (router A) to other routers to be distributed to each peer under it, since the number of peers under different routers is usually different. There will be a situation that multiple nodes choose the same node, since each peer does not coordinate with each other in peer selection. We use the probability selection method to make each peer's probability of being selected proportional to its allocated traffic relaxation, so as to achieve distributed coordination among peer neighbor selection. In addition, peer 1 chooses peer 2 under the same router, which only generates intra-domain traffic, and it does not increase the traffic between routers; therefore, peer under the same router can be given higher priority, so that P2P traffic optimization that is based on location awareness can be realized [28].



Figure 1. The Structure of BT Network.

Figure 2 shows the pseudo code of the peer selection algorithm. Algorithm 1 selects k peers from the candidate set L of peer p to set S while considering the traffic state (peak service demand vector m and current service demand vector c). Note that the business requirement in this paper refers to the aggregation service requirements between core network nodes (hereinafter referred to as nodes), where each component m_i (or c_i) represents the peak business requirement (or current business requirement) from node p to node i. In the first line of Algorithm 1, if the number of peers in Algorithm 1 needs to select k peers from L according to the traffic relaxation of the peer to be selected (lines 2–7). The set L is divided into two parts L_1 and L- L_1 , where the peer p and in L_1 are under the same node or the relaxation of the corresponding node i (m_i - c_i) is positive (line 2). In Algorithm 1, peer in L_1 is selected first (lines 3–5). If the number of peers in the L_1 is less than k, then the missing peer is selected from L_1 and L- L_1 (lines 5). The selection method is to first calculate the traffic relaxation of each peer, which is represented by a vector r (line 6), and then select the peer (line 7), according to the probability with the traffic relaxation as the weight.

Function 1 calculates the traffic relaxation in each peer in the candidate peer set *L* of peer *p*, and then returns the result as a vector. The node set of network is *N* (line 1), L_1 represents the peer with positive traffic relaxation (line 2), and L_2 represents the peer (line 3), which is under the same node with *p*. Line 4 calculates the number of peers under each node and then divides the traffic relaxation between nodes (lines 5 and 7) to obtain the traffic relaxation of each peer. The peer in L_2 is under the same node with *p*, so they are given higher priority and they can be adjusted by parameter α (line 6). The traffic relaxation of peer in $L-L_1-L_2$ is not positive, thus special treatment is carried out, so that the absolute value of the traffic relaxation obtained is still proportional, but with a negative sign (line 7). Finally, the result is returned, that is, the traffic relaxation vector *r* of the peer (line 8). Function 2 chooses *k* peers from set *T* by probability according to the value of the weight vector *w* and returns the result as a set. If the number of peers in *T* is less than *k*, then it directly returns. Otherwise, select *k* peers from *T* to set *S* (lines 2–6). For each selection, the weights of the remaining peers in *T* are normalized (line 3) as probability, and then a peer (line 4) is selected while using roulette gambling [21] by the probability, and the peer is removed from *T* to *S* (line 5). Finally, return to the set *S* (line 6).

Input: L is the candidate set, p is the node that initiates the selection, k is number of peers to be selected, m is peak service demand vector and c is current service demand vector. Output: S collection of selected peers

1. if $|L| \le k$ return $S \leftarrow L$ 2. $L_1 \leftarrow \left\{ u \mid u \in L \land i = u.node \land \\ (i = p.node \lor m_i > c_i) \right\}$ 3. if $|L_1| = k$ $S \leftarrow \emptyset$, $T \leftarrow L_1$, $k_1 \leftarrow k$, return $S \leftarrow L_1$ 4. if $|L_1| < k$ $S \leftarrow L_1$, $L_1 \leftarrow L_3$, $k_1 \leftarrow k - |S|$ 5. $r \leftarrow \text{computeTrafficRelaxation}(p, L, m, c)$ 6.return $S \leftarrow S \cup$ selectPeers (T, k_1, r) 7. Function 1 computeTrafficRelaxation $(p, L, m, c, \alpha = 0)$ 8. count the peers in each node $i \in N$ as c_i , where $N \leftarrow \{i \mid i = node _ of(u) \land u \in L\}$ 9. $r_u \leftarrow (m_i - c_i) / n_i$ for each peer $u \in L_1$ where $i \leftarrow node$ of (u) and $L_1 \leftarrow \{u \mid m_i > c_i \land i = node _of(u) \land u \in L\}$ 10. $r_u \leftarrow \alpha \cdot \max_{v \neq L} (r_v)$ for each peer $u \in L_2$ where $L_2 \leftarrow \{u \mid node_of(u) = node_of(p)\}$ 11. $L_1 \leftarrow L_1 \cup L_2$, $r_u \leftarrow (c_i - \beta \cdot m_i) / n_i$ for each peer $u \in L_3$ where $i \leftarrow node_of(u)$, $\beta \leftarrow \max_{i \in \mathcal{N}} (c_i / m_i)$ and $L_3 = L - L_1$ 12. $r_u \leftarrow -1$ for each node $u \in L_3$ if $r_u = 0$ 13. Function 2 selectPeers(T, k, w)14. if $|L| \leq k$ then $S \leftarrow S \cup L$ 15. else 16. for $i \leftarrow 1$ to k do compute selection probability as $w_u / \sum_{v \in I} w_v$ for each peer $u \in L$ 17. 18. select a peer u from L using roulette wheel selection approach $L \leftarrow L - \{u\}, S \leftarrow S \cup \{u\}$ 19. 20. end for 21. end if

Figure 2. Pseudo code of the peer selection algorithm.

3.2. Choking/Unchoking Algorithm

When peers connect with other peers, peers exchange fragments of their own files with each other until the file is downloaded. BT protocol stipulates that, for each connection of peer p (expressed as $p \rightarrow q$), only when q is interested in p and p is unchoking q, allows for q to download p, q is called a downloader. Therefore, the key point of choking/unchoking algorithm is to decide which peers to unchoking, that is, to select the peers to be unchoked from all peers that have established connections (and the remaining peers are chocked). Choking/unchoking mechanism determines whether peer A offering uploading service to peer B, if peer A reject the service, which is called choking peer B; in the other case, if peer A offers the service to peer B, which is called unchoking peer B.

The choking/unchoking algorithm of the original BT protocol [29] embodies the idea of "Tit-for-Tat", namely that deciding the node to be unchoked by us according to the upload rate of the node to us, which is the so-called "All for one and one for all". We improve the choking/unchoking algorithm by considering the network state, which becomes a double-parameter selection problem. In other

words, the upload rate and the trafffic relaxation of the peer should be simultaneously considered when selecting the unchocked peer. The priority of any peer with a smaller value should be lower in order to maintain fairness between all the peers and the whole network performance. In our algorithm,

the double parameters are normalized first, so that they have the same value range. Subsequently, the smaller of the two parameters is taken to use, so that the double-parameter selection problem is transformed into a single parameter selection problem. Finally, the peer that is unchocked is selected by a probabilistic selection method that is similar to algorithm1. Figure 3 shows the detail pseudo-code of our improved choking/unchoking algorithm.

Algorithm 2: Choking/ Unchoking algorithm Input: the set of candidate peers to be unchoked L, peer which conducts rochoking p, number of unchoked peers k, number of unchoked downloaders k_d , peak value traffic demand vector m, current traffic demand vector cOutput: the set of unchoked peers S 1. if $|L| \leq k$ return $S \leftarrow L$ 2. compute-traffic-relaxation (L, p, m, c, 2)3. if $|L_1| = k$ return $S \leftarrow L_1$ 4. if $|L_1| < k \quad S \leftarrow L_1, \quad L_1 \leftarrow L_3$ count the peers in S which are interested in peer p as $k_1, k_2 \leftarrow k_d - k_1$ 5. compute the transferring rate for each peer u in L_1 as t_u , if p is a seed, t_u is the downloading rate from p to u, otherwise t_u is the uploading rate from u to p. 6. $t_u \leftarrow \max\{\min_{u \in I_1} (t_u)/2, 1\}$ for each $u \in L_1$ if $t_u = 0$ 7. $w_u \leftarrow \min(r_u / \sum r_v, t_u / \sum t_v)$ for each peer $u \in L_1$ 8. if $k_2 > 0$ then 9. $L_4 \leftarrow \{u \mid u \in L_1 \land u \text{ is interested in } p\}$ 10. select-peers-randomly (L_4, k_2, S, w) $L_1 \leftarrow L_1 - S$ 11. 12. end if $k_3 \leftarrow k - |S|$, select-peers-randomly (L_1, k_3, S, w) 13. return S

Figure 3. Pseudo code of the choking/unchoking algorithm.

Algorithm2 considers the network traffic state (peak service demand vector *m* and current service demand vector c) from the candidate set L of peer p to select k peers to be unchoked, where the number of downloaders is k_d . In Algorithm2, if the number of peers in the candidate set L is less than or equal to k, then L (line 1) is directly returned. Otherwise, similar to Algorithm 1, L is divided into L_1 and L- L_1 (line 2), and the peer in L_1 is preferred to be selected (lines 3–5). Line 6 calculates the transmission rate of peer in *T* between *p*. Specifically, if *p* is a seed, then the transmission rate represents its download rate (since the seed is only downloaded by other peers), otherwise it represents the upload rate of the peer. Lines 7–12 ensure the transmission rate of the peer in T is not zero and it guarantees the size sequence. Line 13 calculates the traffic relaxation of the peer in L, and then combines the two parameters of the shared traffic relaxation and transmission rate, calculates the weight of each peer in T (line 14), that is, it separately normalizes the two parameters, and then takes the minimum as its weight. Line 15 calculates the number of downloaders already in S, and then selects the remaining downloaders k_2 from T according to the weight of the peer (lines 17–21), and make the total number of downloaders as k_d (the 16 lines). Finally, k_3 peers (line 23) are selected from the remaining peers in T according to the weight of peers by probability (line 23), so that the total number of peers that are unchoked is *k* (line 22), and the set *S* is returned (line 24).

4. Experimental Verification and Analysis

We use GPS [30] to simulate the standard BT protocol and our improved BT protocol in order to evaluate the performance of the improved BT protocol. We use real network topology and traffic matrix data [31] to evaluate bit-torrent swarm of different sizes. The experimental results show that, when compared with the original BT protocol, our improved BT protocol can significantly enhance the stability of network traffic matrix without sacrificing performance or even increasing performance.

4.1. Experiment Setup and Performance Indicators

4.1.1. General Peer-to-Peer Simulator (GPS) Simulator

We make adaptive improvements to the GPS simulator for our experiment, as our paper focuses on the traffic stability of backbone network. As shown in Figure 1, peer accesses the backbone network through corresponding backbone network nodes or point-of-presence (PoP), so it is necessary to replace the transit-stub topological model of GPS with the transit topological model, that is, only the topology of backbone network (transit network) needs to be considered. Second, at present, the bandwidth bottleneck of BT network is mainly located at the edge of the network, namely, the link between the peer and its corresponding PoP. Therefore, we use the PBB (Peer Based Bandwidth) [28] traffic model to allocate bandwidth resources. Third, when GPS allocates bandwidth resources, the positive and negative directions of the link bisects the total bandwidth of the link, which is seriously inconsistent with the actual situation. We have improved it by managing and allocating the bandwidth of the positive and negative directions of the link. Fourth, we have added a network information provider (NIP) in the GPS, which is responsible for monitoring the network traffic information and providing the current traffic matrix information of the network to the Tracker, and the Tracker uniformly distributes the network traffic information to the peers.

4.1.2. Network Topology

Our simulation experiment uses two real network typologies, the Abilene network and the germany50 network. Abilene network is a backbone network of United States, which consists of 12 nodes and 30 directed edges, and Germany50 network is a backbone network of Germany with 50 nodes and 176 directed edges, the topology of these two networks is shown in Figures 4 and 5.



Figure 4. Topology of Abilene.



Figure 5. Network Topology of Germany50.

4.1.3. Traffic Matrix

We use the real traffic matrix data of Abilene network and Germany 50 network on 10 September 2004 and 15 February 2005, respectively, with a time granularity of 5 min, so there were a total of 288 traffic matrices in one day. The choice of our experimental network is determined by the network size and the availability of traffic data, the network topology and traffic matrix data that are used in our paper can be obtained from the literature [29].

4.1.4. Network Status

In the experiment, the number of downloaders (swarm size) of different BT group is 100, 200, and 300, respectively. Each group has one seed, one tracker, and several downloaders. The construction method of the BT group is as follows: peers are randomly allocated to each node of the network and the upload and download bandwidth of the access link between each peer and its node is 20 Mb/s. All of these three BT groups share a 1 GB file, which is divided into 1024 pieces, each piece is 1 MB in size, and each is divided into 16 blocks, each block is 64 KB in size.

We divide a day into 24 sections, each section is one hour, one hour is divided into 12 slots, and each slot is 5 min. The value of each element of the peak traffic matrix (*m*) of each section is the maximum value of the corresponding real traffic matrix of 12 slots in that section, and the peak traffic demand vector of each node is the row vector of the matrix. In the simulation test, we make the real traffic matrix data of each slot as the background traffic, since new BT traffic will also be generated, so the real current traffic of the network is the background traffic that adds the new BT traffic, which is the real traffic matrix of the network, and the current service demand vector of each node is the row vector of the real traffic matrix.

4.2. Performance Indicators

This paper considers the following performance indicators:

(a) Completion time: the time when all of the peers in the BT group complete file download, which is a direct performance indicator of the BT protocol.

(b) Intra-domain traffic: that is, the proportion of BT traffic beside in backbone network nodes (within the domain) to the total BT traffic. ISPs want the traffic to be confined within their networks as much as possible since ISPs need to pay for inter-domain traffic, that is, the proportion of intra-domain traffic is expected to be as large as possible.

(c) Traffic relaxation: traffic relaxation refers to the difference between the peak value of the business demand and the current value in a section. If traffic relaxation is negative, which indicates that the traffic greatly changes and exceeds the peak of service demand. Therefore, the stability of network traffic can be reflected by the relaxation of traffic with negative value.

4.3. Experiment Result

4.3.1. Completion Time

Figures 6 and 7 show the completion time of the completes file download of the three BT groups in Abilene and Germany50 networks. First, it can be seen from Figures 6a and 7a that the completion time of our improved BT protocol (identified by the IMBT in the figure) is, in most cases, shorter than the original BT protocol [27] (identified by the BTv1 in the figure).



Figure 6. Download Completion Time under Different BT Group of Abilene. (**a**) is completion time comparision of IMBT and BTv1 under different swarm size, (**b**–**d**) is the detail downloading process of cumulative fraction under different swarm size of IMBT and BTv1.

Second, from the distribution of the cumulative completion ratio with the download time, the download rate of our improved BT protocol is similar to the original BT protocol (the slope of the line in the figure). The download rate of our improved BT protocol is similar to the original BT protocol when the BT group is small, but, as the size of the BT group increases, the download rate of our improved BT protocol is faster than the original BT protocol.

Third, the original and our improved BT protocol almost maintains uniform download rate throughout the completion time, but the download rate of the whole BT group becomes very slow when the download is near completion, mainly because most of the peers have finished downloading, and the download rate of the BT group at this time is limited by the bandwidth of a very small number of access links of the peer that have not completed downloading. As can be seen from Figure 7b, this is the reason why alto's download completion time is slower than that of native when the BT group size of Germany50 network is 100. In summary, our improved BT protocol has similar or even better performance than the original BT protocol from the point of view of completion time.



Figure 7. Download Completion Time under Different BT Group of Germany50, (**a**) is completion time comparision of IMBT and BTv1 under different swarm size, (**b**–**d**) is the detail downloading process of cumulative fraction under different swarm size of IMBT and BTv1.

4.3.2. Intra-Domain Traffic

Figure 8 shows the proportion of BT traffic in the domain of the Abilene and Germany50 networks during the download period to the total BT traffic of the network. First, the intra-domain BT traffic of our improved BT protocol is much higher than the original BT protocol, for Abilene network, the intra-domain BT traffic of our improved BT protocol is 19.03–41.40% in different BT groups, while the original BT protocol is only 6.04–8.64% in different groups. The proportion of Germany 50 network is 15.33–36.17% of improved protocol and 2.33–2.71% of original protocol, respectively. Second, with the increase of the size of the BT group, the proportion of the intra-domain BT traffic of our improved BT protocol also increases, because our improved peer selection and chocking/unchoking algorithms prefer to select the peer under the same node, when the size of BT group increases, the number of peers that are allocated to the same node also increase, so the BT traffic in the domain increases, and so does the proportion of BT traffic in the domain. In summary, our improved BT protocol performs better than the original BT protocol in intra-domain BT traffic.



Figure 8. (a) is intra-domain traffic of IMBT and BTv1 under different swarm size of Abilene, and (b) is intra-domain traffic of IMBT and BTv1 under different swarm size of Germany50.

4.3.3. Traffic Relaxation

As shown in Figures 6a and 7a, the completion time of Abilene and Germany50 is 760–904 s and 941–1089 s, respectively, which contain three and four slots. Figure 9 shows the traffic distribution and traffic relaxation distribution during the download of the Abilene when the BT group size is 200. Figure 10 shows the traffic relaxation distribution during the download of the Germany 50 when BT group size is 300. Other sizes of BT groups have similar results.



Figure 9. Traffic distribution and traffic relaxation distribution of Abilene (under the size of BT group is 200). (a) is the traffic distribution of business requirements of IMBT, BTv1 and peak, (**b**–**d**) is the traffic relaxation in slot 1,slot 2 and slot 3.



Figure 10. (**a**–**d**) is the traffic relaxation in slot 1, slot 2, slot 3 and slot 4 of IMBT and BTv1 of Germany50 under the size of BT group is 300.

Due to space limitations, this paper only presents Figures 9 and 10 as an example. Figure 9a shows the traffic relaxation of Abilenenetwork in slot 1. The horizontal axis is the service requirement ID, which is arranged from large to small according to the peak value of the business requirement (marked by sec_max in the figure), and the vertical axis represents the service demand value. Note that the download completion time of both experimental networks is less than one section. As can be seen from Figure 9a, firstly, our improved BT protocol is more stable than the original BT protocol. Among the total of 132 business requirements, 56 of the original BT protocol exceeded its peak value (the value of traffic relaxation allocated is negative), while our improved BT protocol is much less, with only 24 services exceeding the peak value.

Second, service requirements that exceed the peak value are concentrated in the second half of the horizontal axis, which is relatively small. This indicates that the impact of BT traffic on the whole network traffic matrix is mainly concentrated on the demand with relatively small traffic. As our improved BT protocol considers the traffic state of the network and minimizes the impact on the traffic demand with smaller traffic values, which thereby enhances network traffic, and the stability of the matrix is manifested by a reduction in the number of business needs that exceed the peak. This paper uses the traffic relaxation distribution to illustrate how our improved BT protocol enhances the stability of the traffic matrix since the service requirement exceeding the peak can reflect the stability of the traffic matrix.

The other sub-graphs of Figures 9 and 10 are the traffic relaxation distribution of the traffic matrix in each slot of the BT group during download. Figure 9b is the traffic relaxation distribution of Figure 9a. The graph is arranged in the order from small to large. The horizontal axis represents the number of service requirements and the vertical axis represents the size of traffic relaxation. These sub-graphs

show that: first, as compared with the original BT protocol, our improved BT protocol can greatly reduce the number of service requirements with negative traffic relaxation. Second, for the most part of the business demands, our improved BT protocol has more traffic relaxation than the original BT protocol (most of the Alto curve in the figure is above the native), excepting that the traffic relaxation of a small part is smaller than the original BT protocol (a small part of the Alto curve in the figure is below the native). Note that, the larger the traffic relaxation of business demand, the smaller the amount of traffic exceeding the peak value, the more stable the traffic matrix. In summary, our improved BT protocol can significantly enhance the stability of the network traffic matrix when compared with original protocol in terms of the number of business requirements with negative traffic relaxation and the size of traffic relaxation.

5. Conclusions

This paper focuses on the impact of P2P traffic on the stability of network traffic. We make the protocol actively getting current network traffic status and adjusting its own traffic mode to improve the stability of network traffic in a section by improving the current popular BT protocol. The concept of traffic relaxation is used to describe the network traffic state, and the peer selection algorithm is also improved. In our new algorithm, peer is selected according to the relative size of each peer in the BT group. What js more, we also improve the chocking/unchocking algorithm by considering the peer's traffic relaxation and its upload rate when choosing unchocking peer, that is, solve the problem of double-parameter selection. We use the BT simulator GPS to simulate different sizes of BT groups while using real traffic matrix data in two real network typologies. The experimental results show that, when compared with the original BT protocol, our improved BT protocol can significantly enhance the stability of the network traffic matrix without sacrificing or even increasing the performance of the BT protocol.

Author Contributions: Investigation, D.L. (Deguang Li) and Y.W.; Methodology, Y.W. and S.J.; Project administration, Y.Z.; Writing—original draft, D.L. (Deguang Li) and Y.W.; Writing—review and editing, D.L. (Dong Liu) and Y.J.

Funding: This work was supported in part by the State Key Program of National Natural Science Foundation of China under Grant No.61802162. the Major Special Science and Technology Project of Henan Province No. 171100110600 and the Science and Technology Key Project of Henan province under Grant no. 182102210105, 182102210434 and 192102210259.

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

- 1. Wang, Q.; Lu, P. Research on application of artificial intelligence in computer network technology. *Int. J. Pattern Recognit. Artif. Intell.* **2019**, *33*, 1959015. [CrossRef]
- Alshawwa, I.A.; Al-Shawwa, M.O.; Abu-Naser, S.S. An Intelligent Tutoring System for Learning Computer Network CCNA. 2019. Available online: http://dstore.alazhar.edu.ps/xmlui/handle/123456789/129 (accessed on 19 November 2019).
- 3. Awasthi, S.K.; Singh, Y.N. Simplified Biased Contribution Index (SBCI): A mechanism to make P2P network fair and efficient for resource sharing. *J. Parallel Distrib. Comput.* **2019**, *124*, 106–118. [CrossRef]
- Essaid, M.; Park, S.; Ju, H. Visualising Bitcoin's Dynamic P2P Network Topoogy and Performance. In Proceedings of the 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Seoul, Korea, 14–17 May 2019; pp. 141–145.
- Padmavathi, M.; Suresh, R.M. Secure P2P Intelligent Network Transaction using Litecoin. *Mob. Netw. Appl.* 2019, 24, 318–326. [CrossRef]
- Puthusseri, K.S.; Shetty, A.; Patil, M.; Devadkar, K.K. Storage and Availability Aware Fragment Placement for P2P Storage Systems. In Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP), Tamil Nadu, India, 4–6 April 2019; pp. 867–871.
- 7. Zhang, Q.Y.; Wang, X.W.; Huang, M.; Li, K.Q.; Das, S.K. Software defined networking meets information centric networking: A survey. *IEEE Access* **2018**, *6*, 39547–39563. [CrossRef]

- 8. Kim, H.; Feamster, N. Improving network management with software defined networking. *IEEE Commun. Mag.* 2013, *51*, 114–119. [CrossRef]
- 9. Cox, J.H.; Chung, J.; Donovan, S.; Ivey, J.; Clark, R.J.; Riley, G.; Owen, H.L. Advancing software-defined networks: A survey. *IEEE Access* 2017, *5*, 25487–25526. [CrossRef]
- Jimson, E.R.; Nisar, K.; Hijazi, M.H.A. The State of the Art of Software Defined Networking (SDN): Network Management Solution in Current Network Architecture Using the SDN. *Int. J. Inf. Commun. Technol. Hum. Dev.* (*IJICTHD*) 2018, 10, 44–60. [CrossRef]
- 11. Mostafaei, H.; Lospoto, G.; Brandimartey, A.; Di Lallo, R.; Rimondini, M.; Di Battista, G. SDNS: Exploiting sdn and the dns to exchange traffic in a federated network. In Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, Italy, 3–7 July 2017; pp. 1–5.
- 12. Mostafaei, H.; Lospoto, G.; di Lallo, R.; Rimondini, M.; Di Battista, G. Sdnetkit: A testbed for experimenting sdn in multi-domain networks. In Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft), Bologna, Italy, 3–7 July 2017; pp. 1–6.
- 13. Soule, A.; Nucci, A.; Cruz, R.; Leonardi, E.; Taft, N. How to identify and estimate the largest traffic matrix elements in a dynamic environment. *ACM SIGMETRICS Perform. Eval. Rev.* **2004**, *32*, 73–84. [CrossRef]
- 14. Cao, J.; Cleveland, W.S.; Lin, D.; Sun, D.X. Internet traffic tends toward Poisson and independent as the load increases. In *Nonlinear Estimation and Classification*; Springer: New York, NY, USA, 2003; pp. 83–109.
- 15. Wu, Y.; Guo, B.; Shen, Y.; Wang, J.; Liu, X. Green virtual topology design algorithm based on constraint-based routing. *J. Commun.* **2014**, *35*, 112–123.
- Wu, Y.; Guo, B.; Shen, Y.; Wang, J.; Liu, X. Toward energy-proportional Internet core networks: An energy-minimized routing and virtual topology design for Internet protocol layer. *Int. J. Commun. Syst.* 2015, 28, 513–533. [CrossRef]
- 17. Nucci, A.; Sridharan, A.; Taft, N. The problem of synthetically generating IP traffic matrices: Initial recommendations. *Comput. Commun. Rev.* **2005**, *35*, 19–31. [CrossRef]
- Karagiannis, T.; Rodriguez, P.; Papagiannaki, K. Should internet service providers fear peerassisted content distribution? In Proceedings of the Internet Measurement Conference, Berkeley, CA, USA, 19–21 October 2005; pp. 1–14.
- Yang, P.; Xu, L. On tradeoffs between cross-ISP P2P traffic and P2P streaming performance. *Comput. Netw.* 2011, 55, 4033–4042. [CrossRef]
- 20. Liu, Y.; Xiao, L.; Liu, X.; Ni, L.M.; Zhang, X. Location awareness in unstructured peer-to-peer systems. *IEEE Trans. Parallel Distrib. Syst.* **2005**, *6*, 163–174.
- 21. Ren, S.; Tan, E.; Luo, T.; Chen, S.; Guo, L.; Zhang, X. TopBT: A topology-aware and infrastructure-independent BitTorrent client. In Proceedings of the IEEE INFOCOM, San Diego, CA, USA, 15–19 March 2010.
- 22. Zhou, R. P2P traffic optimization based P2P cache (in Chinese). Telecommun. Netw. Tech. 2009, 1, 11–15.
- 23. Hefeeda, M.; Saleh, O. Traffic modeling and proportional partial caching for peer-to-peer systems. *IEEE/ACM Trans. Netw.* **2008**, *16*, 1447–1460. [CrossRef]
- 24. Aggarwal, V.; Feldmann, A.; Scheidler, C. Can ISPs and P2P Systems Co-Operate for Improved[22]Performance? ACM SIGCOMM Comput. Commun. Rev. 2007, 37, 29–40. [CrossRef]
- 25. Bruni, L.; Giacomazzi, P.; Poli, A. Performance optimization of the BitTorrent protocol in the framework of P4P IETF ALTO. In Proceedings of the IWCMC 2013, Sardinia, Italy, 1–5 July 2013; pp. 431–436.
- Gurbani, V.K.; Hilt, V.; Rimac, I.; Tomsu, M.; Marocco, E. A Survey of Research on the Application-layer Traffic Optimization Problem and the Need for Layer Cooperation. *IEEE Commun. Mag.* 2009, 47, 107–112. [CrossRef]
- 27. Madhyastha, H.V.; Isdal, T.; Piatek, M.; Dixon, C.; Anderson, T.; Krishnamurthy, A.; Venkataramani, A. iPlane: An information plane for distributed services. In Proceedings of the 7th Symposium on Operating Systems Design and Implementation, Seattle, WA, USA, 6–8 November 2006; USENIX Association: Berkeley, CA, USA, 2006.
- 28. Zhang, G.; Tang, M.; Cheng, S.; Zhang, G.; Song, H.; Cao, J.; Yang, J. P2P traffic optimization. *Sci. China Inf. Sci.* **2012**, *55*, 1475–1492. [CrossRef]
- 29. Chokkalingam, A.; Riyaz, F. BitTorrent Protocol Specification v 1.0. CSI 5321. 2004. Available online: cs.ecs.baylor.edu/~donahoo/classes/5321/projects/bittorrent/BitTorrent%20Protocol%20Specification.doc (accessed on 20 November 2019).

- 30. Yang, W.; Abu-Ghazaleh, N. GPS: A general peer-to-peer simulator and its use for modeling BitTorrent. In Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Atlanta, GA, USA, 27–29 September 2005; pp. 425–432.
- 31. Orlowski, S.; Wessaly, R.; Pioro, M.; Tomaszewski, A. SNDlib 1.0—Survivable network design library. *Networks* **2010**, *55*, 276–286. [CrossRef]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).