CoolStreaming and Gossip Algorithm in P2P Service Networks

Weiqi Feng¹, Lei Liu²
¹College of Engineering, Informatics, and Applied Sciences; Northern Arizona University, 2112 S Huffer Ln, Flagstaff, AZ 86011, USA
²School of Electronic Science and Engineering; Nanjing University, No.163 Xianlin Ave; Qixia Dist; Nanjing City, Jiangsu Prov; China
angela@cas-harbour.org

Abstract. In recent years, peer-to-peer (P2P) as a new network application model has become more and more popular for its scalability, high fault tolerance and other outstanding advantages. The main feature of P2P technology is to ensure the network resources distributed on terminal computers, including computing resources, bandwidth resources, content resources, etc., are fully utilized to reduce the consumption of central server resources. This paper analyzes the performance differences of nodes in P2P networks from many aspects, such as available bandwidth, online time and storage space. Based on this mechanism and with the idea of mission algorithm, the data distribution efficiency of this algorithm is significantly improved compared with the classical mission algorithm, which can ensure the minimal network load and achieve the efficiency of flood data distribution strategy.

1. Introduction
In recent years, P2P (peer-to-peer) has been thoroughly studied from the first generation to the third generation and has been used extensively on the Internet. With the rapid development of the Internet, streaming media technology that can meet the requirement of "download and play" has been developed rapidly [9]. P2P technology transfers multicast from the network layer to the application layer, making full use of the peer-to-peer computing of each node in the Internet, which has great advantages in utilization, scalability, fault tolerance, etc., so it is quickly applied to streaming media services, and forms P2P streaming media technology. Since it was first proposed by Demers et al., in 1987, gossip algorithm has gained wide attention in the field of distributed computing. Because of its simplicity, high efficiency and strong fault tolerance, the gossip algorithm has been implemented in many fields, such as information dissemination, load balancing, routing algorithm and so on. In peer-to-peer networks, there are three main applications of the gossip algorithm: data distribution, aggregate computing and overlay network management. In the specific application process, countless researchers have improved the classic gossip algorithm according to different application environments and technical requirements [10].

2. P2P Technology
There are three main P2P systems in the first generation: Napster, Gnutella, and Freenet. Napster is the most famous. It provides users with the download of shared MP3 music files. Through the central server, it saves the information of all music tracks and storage locations uploaded by Napster users. Its
principal disadvantage is that there is only one point of failure. Nutshell is a pure P2P file sharing system. It does not have a central server. All queries are executed in a limited flooding manner through nodes in the network. The number of message forwardings is limited by TTL (time to live together, survival time). The main disadvantage is that the network generated a lot of traffic. Freenet is a Java based cross-platform distributed file storage system. Its biggest feature is anonymity. File publishers, queries and file holders are all anonymous in Freenet. The biggest problem of the first generation P2P system is the non-scalability of queries: the annoying central server is the bottleneck of the system, and armilla and Freenet cover a layer of virtual logical topology on the physical network topology. Each node is connected with several other nodes through the virtual logical topology, but the storage of data is independent of the virtual logical topology, and the search in the system is almost random. In order to solve the scalability problem of the earliest generation P2P system, a lot of research on the second-generation P2P system has focused on how to construct a highly structured system. In these structured systems, the overlay topology is strictly controlled, and the files are kept in a specific location. The system provides a mapping service from the file identifier to the node identifier of the saved files, and then queries the request path to the node. Through this method, “exact match” of file queries is achieved. In recent years, many research results have been based on DHT (Distributed Hash Table) distributed query and routing algorithms [11].

3. Based on the Gossip Protocol Model
In the gossip-based protocol model, the node first sends a message to a group of surrounding nodes, and the surrounding nodes forward the message as needed after receiving the message. The model has no centralized control and therefore has good reliability and is a unstructured distributed model.

In the model based on tree topology protocol and extension, the relationship between nodes is clearly defined, and the child node obtains data from the parent node. In the model based on the gossip protocol, there is no requirement to construct complex topological relationships between nodes, and there is no definite parent node. Each node estimates other nodes in the system through the gossip protocol. The scheduling algorithm exchanges data between nodes, each of which is the recipient and provider of the data. In this model, the data source of each node does not depend on a specific parent node, which can be obtained from multiple nodes, and can increase the efficiency of the use of each node. Therefore, the system has stronger robustness and more reliable quality assurance, but usually requires a larger cache, and the start-up delay is relatively large [6].

In recent years, a large number of such systems have appeared at home and abroad, such as the CoolStreaming system developed by Zhang Xinyan of Hong Kong University, the PPLive developed by Huazhong University of Science and Technology, the upstream developed by Ningbo University and the QQLive developed by Tencent. These systems have achieved good results, especially PPLive has been watched by more than 100000 people in the same channel. CoolStreaming is a typical model based on the gossip protocol. In CoolStreaming, each node is the recipient and the provider of data. The server is a special node. A data provide is called the source node. The CoolStreaming system mainly includes the following aspects: (1) node management; (2) data representation and exchange; (3) data scheduling algorithm; (4) error recovery and partner node optimization.

3.1 Node Management
For convenience, the terms used in the gossip protocol model are explained. Partner: each node maintains a list of some other nodes in the system. These nodes are called partner nodes or candidate nodes for that node. Requesting node and active node mean selecting a node from its partner node as a subset E, the nodes in this subset provide flow data for A. Here, we call node A the requesting node or the receiving node, and the node in subset E the provider node or the active node [8].

In CoolStreaming, each node has a unique identification ID throughout the system, such as an IP address. Each requesting node maintains a list of its partners nodes mcache. When a new node is inserted, it will first request the source node. The source node randomly selects a node from its mcache as a proxy for the new node. Different nodes get an initial list of partner nodes from the proxy node.
Each node in the list has a life cycle HL, which will decrease over time. When zoomed out to 0 inch, it is treated as an invalid node, and will be removed from the list. However, in dynamic systems, in order to prevent the TTL of a node in the mcache from being reduced to zero, each node periodically sends a new message to all partner nodes in the mcache to announce its existence (the TTL of the new message is the maximum survival time, and in order to ensure the normal operation of the system, the time that the node sends the message is less than the maximum survival time). After receiving the message, the partner node creates the message first if there is no information about the node in its mcache; if it is a new message (the propagation of the message may form a loop), it will update the TTL value of that node in its list and continue to relay the message to its partner node[5].

Taking figure 1 as an example, the node PR sends a message to its partner nodes PL, P2 and P3 periodically. When P1 receives the messages, it will determine whether it is a new message. If so, it will update the TTL of PR in plmcache, and continue to send the message to its partner nodes P4 and P5. P4 still sends the message to its partner node PN, which is the partner node of P2. When PN forwards the message to P2, P2 receives the previous message. The news is no longer partially broadcast. In order to avoid message flooding and network congestion, here it can control the maximum message lifetime or message propagation time. According to the literature, the spread of TTL is very helpful in the first four rebroadcasts. As the number of replays increases, the number of redundant messages increases dramatically. The main reason for generating a large number of redundant messages is the presence of loops in the network.

Figure 1 CoolStreaming model message propagation

3.2 Data Representation
In CoolStreaming, the direction of data transmission between node partners is not fixed. Partners exchange data based on their cached data. Therefore, nodes and partners begin to understand the content of cached data. In CoolStreaming, video data is divided into blocks of the same size. A buffer map (BM) is used to refer to whether a node has a data block. Nodes and partners exchange BMs and they keep learning about the cache between them. In the implementation of CoolStreaming, each data block represents one second of data, and a sliding window is utilized to represent the BM. The size is 120 pieces. Enter 120 bits in the BM. Each bit represents a data block. A bit value of 1 indicates a piece, and 0 indicates no pieces. Since the sliding windows of different nodes cannot guarantee the exact same data, CoolStreaming uses two bytes to represent the serial number of the first data block in the sliding window [4].

3.3 Data Scheduling
The purpose of scheduling is to make data blocks from partner nodes. In a static and homogeneous environment, since each node has substantially equal bandwidth, data blocks can be obtained randomly from each partner node. However, in a dynamic and heterogeneous network, more intelligent scheduling algorithms are needed. There are two constraints on scheduling: (1) each data block must be fetched before the maximum delay of playback, and the data block that misses the
maximum delay should be as few as possible; (2) the bandwidth of each partner is different. If there are fewer providers of data blocks, it is very difficult to meet the requirement of maximum delay. Therefore, in CoolStreaming, the algorithm of least block first is adopted [7].

3.4 Error Recovery and Partner Node Optimization

In the CoolStreaming system, nodes are also left or interrupted at any time. In any case, the TFRC (TCP Friendly Rate Control) protocol deployed in the timed BM exchange and CoolStreaming system can find the left nodes. In addition to these mechanisms, CoolStreaming also has built-in some mechanisms to enhance the fault tolerance of the system [3].

When a node leaves normally, it will generate a message from leaving. If the node is interrupted abnormally, its partner node will detect its interruption, and the partner will send out a message of leaving the node. The delivery method of the message is used when the node leaves and when the node joins. The message that the interrupt node leaves will be sent repeatedly by different partners. The node only forwards the message when it is received for the first time. All nodes that receive the message will update their mcache [2].

The partner node optimizes the strategy of CoolStreaming. The new node N obtains the initial partner nodes from the server and selects the active nodes from them. These nodes are not necessarily the "best" functional nodes of N, or even the "best" active nodes. With the change of system nodes and time, the number of partner nodes and the connection bandwidth of node N will change. Therefore, each node is required to update its activities regularly. The Nodes delete some active nodes with slow connection speed, and select some nodes with fast connection speed from the standby nodes to become innovative active nodes. Since each node is both a provider and a recipient, scores in both directions are considered. When a better active node is found, the node with the lowest score in the original active node will be deleted from the partner list [1].

4. Discussion

At present, in the overlay network management of peer-to-peer network, the gossip algorithm is primarily used to solve two problems: one is the construction and maintenance of the overlay network topology; the other is the overlay network member management. Currently, there are relatively few solutions to the above problems. In the process of establishing a topology of mobile peer-to-peer network, some control information needs to be propagated, and the relevant nodes need to be notified periodically of the changes caused by node exit or failure, so as to effectively control the topology. Therefore, in order to effectively transmit the topology control information in mobile peer-to-peer networks, and keep the copy consistency of the node resource list, the topology control of overlay networks is implemented, and the anti-interference of overlay networks is improved.

5. Conclusion

In the gossip protocol model, data is transmitted in a data-driven way, and the stream data is always provided by multiple nodes. Each node is not only the recipient but also the provider of the data, which is a kind of unstructured overlay network. It can be seen that the gossip protocol model can achieve higher data reception rate and smaller overall delay, and is more suitable for large-scale P2P Live Streaming in a dynamic environment. However, the dynamics of a P2P network will affect the network topology, especially in node-based P2P networks. In the near future, the reliability of super nodes is the basis for the stability of the entire network. Therefore, on the basis of this study, this paper introduces redundant super nodes to build virtual super nodes in the gossip protocol model, and further analyzes the problems that need to be solved and paid attention to in the process of constructing and maintaining virtual super nodes.

References

[1] Steve, M. (2003) Streaming Media Bible. Beijing: Electronic Industry Press. pp.30-40.
[2] Kangashaiju, Jussi. A T. (2002) Internet Content Distribution, PhD thesis, April.
[3] Stephen, E. D, David. C. (1990) Multicast Routing in Datagram Internet works and Extended LANS. ACM Transaction s on Computer Systems, 8(2):85-110, May.
[4] Stephen, E. D, Deborah E., Dino, F. et al. (1996) An Architecture for Wide-area Multicast Routing. IEEE/ACM Transaction on Networking, 4(2), April.
[5] Stephen, E. D. (1991) Multicast Routing in a Datagram Internetwork. PhD thesis, Stanford University, December. pp. 90-93.
[6] Wu, G. Y., Qiu, X. G., Wan, Y. Z. (2001) First class media technology and application of network video. Beijing: Beijing University of Posts and Telecommunications Press.
[7] Puri, R., Ramchandran, K., Lee, K. W. et al. (2000) Application of FEC based multiple description coding to Internet video streaming and multicast. Proc, Packet Video Workshop, Cagliari, Sardinia, Italy.
[8] Wang. Y, Orchard, M. T, and Reibman, A. R. (1997) Multiple description image coding for noisy channels by pairing transform coefficients, Proc. IEEE Workshop on Multimedia Signal Processing, pp. 419-424.
[9] Schularinrme, H. et al. (1999) RTP: A Transport Protocol for Real-Time Applications. Internet draft, IETF RFC1889.
[10] Yuan, H. L. (2005) Research and implementation of streaming media real-time transmission based on DirectShow. [Master's Thesis]. Dalian: Dalian University of technology.
[11] Network Working Group. (1998) RFC2326. Real Time Streaming Protocol (RTSP).