Protecting Agent from Attack in Grid Computing

Byungryong Kim

DongBang Data Technology Co., Ltd., Department of Computer Science and Engineering, Inha University, Korea
doolyn@gmail.com

Abstract. P2P networks provide a basic form of anonymity, and the participating nodes exchange information without knowing who is the original sender. Packets are relayed through the adjacent nodes and do not contain identity information about the sender. Since these packets are passed through a dynamically-formed path and since the final destination is not known until the last time, it is impossible to know who has sent it in the beginning and who will be the final recipient. The anonymity, however, breaks down at download/upload time because the IP address of the host from which the data is downloaded (or to which it is uploaded) can be known to the outside. We propose a technique to provide anonymity for both the client and the server node. A relay node along the path between the client and the server node is selected as an agent node and works as a proxy: the client will see it as the server and the server looks at it as the client, hence protecting the identity of the client and the server from anonymity-breaking attacks.

1 Introduction

The recent P2P retrieval systems can largely be divided into flooding-based and distributed hash table-based models. FreeNet[1] and Gnutella belong to Flooding model and Tapestry[2,3], CAN[4] and Chord[5] to distributed hash table model.

P2P systems are too free and irresponsible to secure reliability as achieved in server-client environment. However it cannot manage the nodes by nature and each node should maintain itself with independent authority that each node must have anonymity. Moreover it must be available for retrieval and must not be excessively expensive(CPU/Bandwidth), and this threshold is different for clients/servers/peers. And Peers are extremely sensitive to bandwidth usage. Hence we propose packet preemptive proxy service techniques that maintain both high speed and anonymity in flooding-based P2P file share systems requiring anonymity.

In a flooding-based model broadcasted retrieval query and ping packet basically provide anonymity but dynamic routing is not implemented so that they do not support anonymity when uploading and downloading. Therefore this may result in unintentional exposure of node information in P2P network in which non-specific majority is participating. Thus attacks, such as denial-of-service and storage overflow, may occur to the exposed node, whether it is malicious or not.

The existing techniques to secure anonymity include MUTE[6], Onion Routing[7,8], Crowds[9] and Mantis[10]. In order for MUTE to protect anonymity, file

Y. Luo (Ed.): CDVE 2006, LNCS 4101, pp. 115 – 122, 2006.

share is made through other clients such as Jondo. However this slows speed in a large capacity file because it transfers file through many nodes. In Mantis, model supplementing this consideration, UDF channel is used without passing through nodes but this requires additional control data communication for UDF channel. Onion Routing uses data encryption to secure anonymity. Client should connect to proxy performing encryption to firstly connect to Onion Routing. Crowds was developed for user privacy while browsing web site and which is similar to MUTE. Client participating in Crowds requests contents not to server but to other client participating in Crowds to get desired contents. Anonymity is guaranteed by this technique.

Our goal is to preserve anonymity and provide retrieval and file share service in a quick and easy way without using additional control data communication for UDP communication and file sending method passing through number of nodes.

2 Related Researches

Techniques to secure anonymity of server and client include MUTE, Onion Routing, Crowds, and Mantis. MUTE does not provide file share service because server is directly connected to client in order to secure anonymity. Passing through many intermediate nodes such as Jondoe it sends data(information/contents). So sending high capacity file such as .avi or .divx will incur sizable waste of bandwidth because it passes through many nodes.

In order to secure anonymous connection Onion Routing uses data encryption to conceal routing header and make statistical computation hard to detect routing path. To make the first connection to Onion Routing, client must make routing path and connect to proxy that encryptizes data. Then client gets to destination following the routing path through Onion routers. Each router releases(removes) encrytized data layers one by one. By repeating this process client knows the next onion router and finally reaches to the final destination. Inversely when getting back it reaches to the final destination by adding encrytized data layer one by one. Although it secures anonymity, it needs proxy between network infrastructure and application and costs a lot for the encryption.

Crowds is developed to protect user's privacy during web browsing. To get contents client participating in Crowds does not directly request contents to server having known the server's address but requests contents to other Jondoes participating in Crowds and anonymity is maintained this way.

Mantis is similar to Crowds but it instantly sends answer without passing through Jondo. Concealing its own IP address, server sends file using UDP but control data communication is necessary to control packet loss aroused in the course of UDP communication between server and client and to control resending. Control data communication is performed by passing through many Jondoes between server and client. Connection among each node is encryptized as Onion Routing.

3 Providing Anonymity

The packet preemptive proxy service model that we propose is based on Gnutella protocol[11]. Packets used here are Query, QueryHit, and Push packets. In this