

March 2006

Center for Global Communications,
International University of Japan

Harks Roppongi Bldg. 2F, 15-21
Roppongi 6-chome, Minato-City,
Tokyo 106-0032 Japan

URL <http://www.glocom.ac.jp/>

TEL +81-3-5411-6677

FAX +81-3-5412-7111

"Clarifying the Original Goals of Winnie Technology"

Isamu KANEKO

Winnie developer

Report: Satoshi HAMANO

Researcher

Center for Global Communications (GLOCOM)

International University of Japan.

Report of a presentation by Isamu Kaneko at a symposium "Winnie Technology and Ethics", organized by GLOCOM and the League of Software Engineers (LSE), 28 January 2006.

Winnie developer Isamu Kaneko took the microphone first at the "Winnie Technology and Ethics" Symposium. Winnie is a popular file-sharing software used for music and movie files as well as for personal data. Kaneko noted that there are two distinctly different versions of Winnie, the later version 2 being equipped with an "anonymous BBS system" based on the original file exchange mechanism.

The speech proceeded on the premise that two versions of the software exist and consisted of three parts: he described both Winnie V1.0 and the Winnie V2.0 BBS system and then discussed future prospects and issues regarding next-generation P2P systems.

The first Winnie beta version was released in May 2002, and by the time its official version was released a year later in April 2003 development work on Winnie V1.0 had been completed. Winnie V2.0 was released in May 2003. Winnie development was brought to a halt when Kyoto Prefecture Police arrested Kaneko in 2004 for aiding and abetting a

copyrights violation.

Below, GLOCOM Research Fellow, Satoshi Hamano reports on Isamu Kaneko's presentation.

1. Winny V1.0 file-sharing software

1-1. Freenet anonymity

Kaneko started his speech by saying he began developing Winny V1.0 because he was impressed by the "anonymity" provided by Freenet P2P software. Freenet was created by developer Ian Clarke to help bring about free speech on the Internet. It was designed not only for file sharing but also the anonymous transmission and reception of e-mail messages.

The idea of anonymity requires some explanation. People think that the Internet is highly anonymous, but they are wrong from a technological standpoint. In reality, every time users view a website they provide personal information including data about their IP address and software to the destination site. TV viewing is anonymous because it involves only the reception of electronic waves, but the Internet is bi-directional, meaning that users cannot receive information unless their personal data is transmitted, and there is no anonymity in that. Freenet helps to bring about anonymity by encrypting files and

scattering them in pieces over the network. This mechanism makes it so that no one can identify who is attempting to distribute what content over the network.

Kaneko pointed out that compared with other P2P software, Freenet has drawbacks relating to its file searching and transferring efficiency. Since Freenet disperses files in pieces, it has to search for the file fragments in various locations when recovering files. Since Freenet users typically do not have the software activated all the time some fragments may get lost. Before the advent of Freenet, earlier types of P2P software had better searching efficiency because users' file information was concentrated on a server. Napster and other services that Kaneko called first-generation P2P software used this approach.

P2P refers to "peer-to-peer" communication without a server in the middle, so the so-called first-generation approach cannot really be called a P2P network in the strict sense of the term, it was more a hybrid approach. Anonymity was difficult to maintain in these early hybrid systems since all data was collected in a server. There was a trade-off between anonymity and efficiency in the early development of P2P software.

1-2. Aiming to achieve both anonymity and efficiency

Kaneko's objective was to develop the ideas behind of Freenet and resolve the dilemma of anonymity and efficiency. The first technological issue he wanted to solve was how to conceal the information source while maintaining file searching and transferring efficiency. Kaneko was attracted to Freenet by its anonymity principle, but he was the one who was able to bring about true anonymity and he did it by developing the Winny system from scratch.

The Freenet method of scattering encrypted files was wasteful. As already explained, anonymity is enhanced when files are fragmented, but file recovery becomes difficult. Kaneko grasped the essential nature of anonymity as involving multi-stage information relaying rather than information diffusion. If files are distributed through multi-stage channels, it would be difficult to trace their original source. This is how Kaneko reinterpreted the concept of anonymity when he designed Winny.

Kaneko introduced proxy servers into the P2P system to conduct multi-stage relaying. As their name indicates, proxy servers are "substitute" servers and as such they can be used in diverse ways. They can be hooked up to a website

browser to assure anonymity. This way, only the IP address of the proxy server, and not the IP address of the sender's machine remains on the other web server. However, since the proxy server retained the sender's IP address complete anonymity could not be secured. However, if the proxy server used a multi-stage relay then it became difficult to trace the original sender.

Proxy technology was also used to enhance efficiency. In the days when communication speed was low it was wasteful to establish communication with websites in the United States each time parts of a file were to be sent or received, so proxy servers were installed within the LAN network to disperse network loads more locally. Each user temporarily stored (cached) data obtained from external web servers. When someone tried to view the same website, the cache could be displayed without having to access the external server directly to obtain data. This relieved the necessity of fetching data from distant servers with every website access.

Kaneko thought to utilize these two features of proxy servers to achieve both anonymity and efficiency. If communications were relayed in multiple stages, anonymity would be enhanced because it would be difficult to trace the original sender. Files can be cached in

each node when multi-stage relays are used, eliminating the need to communicate with the primary sender of the file, thereby enhancing transfer efficiency. Winny offers both anonymity and efficiency because each of the nodes connected to its network serves as a proxy server.

1-3. Winny cache mechanism

Data distributed over Winny are separated into a "key" which serves as file index information and the body, or "cache" that is the file itself. Key to this is meta data in which the file name, file size, and IP address that holds the file body and the "hash values"¹ have been stored. The "cache" refers to the encrypted contents of the file. Keys have smaller file sizes, so they can be exchanged in large volumes over the network. Caches accumulate at each of the nodes and are utilized as a transfer resource.

The Winny file transfer process has been designed to automatically scatter key files exchanged by users, enhancing file searching efficiency. When a target key file is hit during a search, Winny makes reference to the IP address or "position data" for the file cache. Winny then starts transferring data when the machine that possesses the cache is identified through this position data. Caches are not distributed constantly but are

transferred only on request from another user. Kaneko describes the key as a push-type feature and the cache as a pull-type feature.

"Position data" has been designed to be rewritten according to a certain algorithm. In other words, Winny was designed to rewrite the sender of letters randomly and then intentionally induce erroneous delivery. When rewriting the position data for the key, Winny also designates a node that does not have the file body as possessing a cache. This would seem to counter the goal of efficiency in order to achieve anonymity, but it is precisely this design that resolves the above-mentioned dilemma between anonymity and efficiency.

This means the erroneously designated Winny node refers to the key files it holds and traces the node that actually possesses the cache. When the node that really holds the cache is found, Winny starts the transfer. Winny may upload a file before the download of the file is completed, so relaying to the node that originally requested the file. The bridged Winny node stores the relayed file as a cache in its hard disk, and the stored cache is used as a resource for the next relaying.

Winny repeats this cycle, generating relaying opportunities following

erroneous delivery and accumulating caches for each relay. In this way, caches are scattered across the network. The in-built feature that at first appears to be an error is actually the source of Winny's enhanced efficiency. Winny has also been designed to rewrite position data more often for more popular files. Since popular files are exchanged often they tend to put pressure on network bandwidth, so enhanced file transfer efficiency was introduced by more frequently generating the cycle of erroneous delivery, to bridging, to cache storage. Anonymity was also enhanced because the repetitive cycle made it more difficult to distinguish the primary sender of the file.

Some have criticized this Winny mechanism, saying relaying is a wasteful system that only raises the volume of traffic, or causes more bandwidth congestion. In response, Kaneko counters that this is a misunderstanding of how Winny works, some data may indicate that Winny and other P2P systems take up the majority of traffic volume on the Internet, but in Kaneko's view, the Winny relaying system was created to utilize network resources efficiently.

Winny can be summarized as a mechanism for simultaneously achieving communication efficiency and anonymity through intentional erroneous delivery.

1-4. Winny V1.0 as a third-generation system

Kaneko made the following three points about Winny V1.0. First, related to factors behind the dissemination of Winny V1.0, Kaneko maintained that "Winny has indeed succeeded in achieving anonymity and efficiency simultaneously, but this was not the primary reason for its release." Search efficiency was always an issue of primary importance for P2P software and Kaneko believes that this issue has already been addressed by other software. He places greater importance on "node maintenance ratio," or the issue of how to get Winny users to remain as nodes serving the Winny network. How to avoid termination of the program was another issue that Kaneko wanted to investigate.

Kaneko's solution was Winny's "automatic downloading function," where Winny users can automatically download a file by searching for a target filename and then letting the program do its work. According to Kaneko, this function did not exist at the outset but was added because file transfer efficiency had initially been so terrible. This function allows Winny to be activated continuously, led to an increase in the number of users and helped improve the node maintenance ratio. Kaneko believes this

is an important reason why Winny has become so popular.

His second point addressed the capability of the Winny network to withstand failures. Pure P2P features are less likely to bring about total system failure than the hybrid type. When a central search server is attacked, the entire network of the hybrid type fails. However, a pure P2P network never fails in its entirety no matter what nodes are attacked. The continuing strength of the Winny network as it was originally designed is a testament to this fact.

Kaneko's third point concerned concepts of "upstream/downstream" and "clustering" technologies, which have been introduced to enhance efficiency. The former is a mechanism that concentrates key files in nodes that are considered "upstream," or those that have high line speeds. The latter mechanism positions users with similar interests near the network by making preferential connections to nodes where the same search keywords have been input. These mechanisms have improved Winny efficiency.

Kaneko also said that on the basis of these new features Winny can be considered to have achieved third-generation P2P file sharing. Napster and other hybrid-type systems

were first generation file-sharing systems. Gnutella and other pure P2P systems are second generation. Kaneko started developing Winny in 2002, and with its cache mechanism he claimed it is worthy being called "next-generation", or improved version over the second-generation, P2P system. He added that many limitations can be found in this software, however, four years have already passed since its development.

2. Winny V2.0 as an anonymous BBS system

Kaneko went on to explain the large-scale anonymous BBS system, Winny V2.0.

"2channel" is assumed to be the prototype for the large-scale anonymous BBS. 2channel is operated by a bulletin board program on the web so load concentration, a drawback of the server-client type system, has been an issue.² Since around 2000, before Kaneko developed Winny, experts had been discussing how to create a large-scale bulletin board based on a P2P system. Kaneko believes that with the release of Winny V1.0 "all technological experiments for file sharing software had been completed," and he thought of applying the large-scale Winny network to other problems. This is how he began developing other devices for the anonymous BBS system. Kaneko stated that his primary inspiration in

developing Winny V1.0 was intellectual curiosity, but he also felt obligated as a software engineer to develop Winny V2.0.

The basic concept of Winny V2.0 is to treat the BBS "thread" as a Winny cache file. The thread refers to a series of written statements on a certain topic (the title of the thread). At the time of writing, up to 1,000 statements can be written on a single 2channel thread. With Winny V2.0, statements are numbered in series from the first to Xth posting.

File synchronization became an issue. File damage and loss due to the widespread distribution of files often occurs with P2P systems. File identity needed to be secured so that they did not undergo changes during communication. Kaneko said that Winny V1.0 solved the synchronization problem for the P2P system with a simple mechanism the called hash value (see Note 1), which maintains synchronicity by making sure that none of the file contents get overwritten. However, it is not possible to apply the same method to BBS systems because when a written message is added in sequence to a discussion thread, the "file contents are over written during communication." The hash value mechanism of Winny V1.0 cannot be applied to Winny V2.0.

A new mechanism where specific nodes

(masters) manage each of the threads (i.e. cache files), was first put into effect with Winny V2.0. Each of the threaded messages (in addition to their file contents) was given to the node of its respective administrator. In this way the master concentrates the latest postings thereby avoiding the issue of file synchronicity. Connections to the master always become congested when messages are written, indicating that the anonymity through multi-stage information relaying brought about with Winny V1.0 would not work. It was for this reason that Winny V2.0 was designed to have messages written from an adjacent node.

Kaneko stated that there was a critical drawback with Winny V2.0 (still in its development stage) in that the mechanism was unable to protect the anonymity of the master. Winny V2.0 specifications are such that a direct connection with the master can be established by repeatedly hitting the thread "read" button. In other words, the node that first established the thread can be identified just by calling to read the file by hitting the button continuously.

This concluded Kaneko's explanation of Winny V2.0 as an anonymous BBS. Winny V2.0 solved the synchronization issue found among other dispersed-type BBS with its mechanism for integral

master management, sacrificing instead the anonymity of the master. Kaneko said he had ideas about how to solve trade-off situation, but could not implement them because Winny V2.0 development was terminated by the court order following his arrest by Kyoto Prefecture Police.

3. Future prospects and issues concerning next-generation P2P systems

Kaneko concluded by mentioning two issues associated with next-generation P2P systems. The first is whether Winny can be made an open system and the second whether P2P systems can be managed.

First, the source code for Winny was never disclosed, it is not open source. Despite the fact that its protocols and encryption systems have been described in books, its source code has never been revealed. Kaneko said that since Winny file encryption does not contribute to anonymity, anonymity cannot be retained unhindered even if the source code is made public.

When asked why Winny was not given an open source license, Kaneko said that it was an issue of efficiency. To begin with, there are a certain number of "free riders" among P2P software users who only downloaded and did not upload files to the network.³ If the source code were made open, he was concerned someone

would modify Winny to permanently disable file uploading. Kaneko said he could not make the source code open due to concern over the possibility of this situation.

He also said the situation was the same today as 2003. Today, P2P systems that differ from Winny have been developed. The open-source BitTorrent, a file-sharing system developed by Bram Cohen since 2001, has taken care of the issue between efficiency and source code disclosure. Users are aware that popular files exhibit extremely high transfer efficiency because they can be downloaded simultaneously from numerous nodes.

Kaneko categorized BitTorrent as a third-generation P2P system and pointed out the "trilemma" involving anonymity, efficiency and being open source (see Figure 1), saying Winny was able to attain both anonymity and efficiency, but it could not be made an open source.

Freenet is open source and helped achieve anonymity, but it was inferior in terms of efficiency. BitTorrent is open source and excels in efficiency but does nothing about anonymity. There is no single system that satisfies all three issues of anonymity, efficiency and being an open source. Kaneko said he believes that such a system is possible and it is

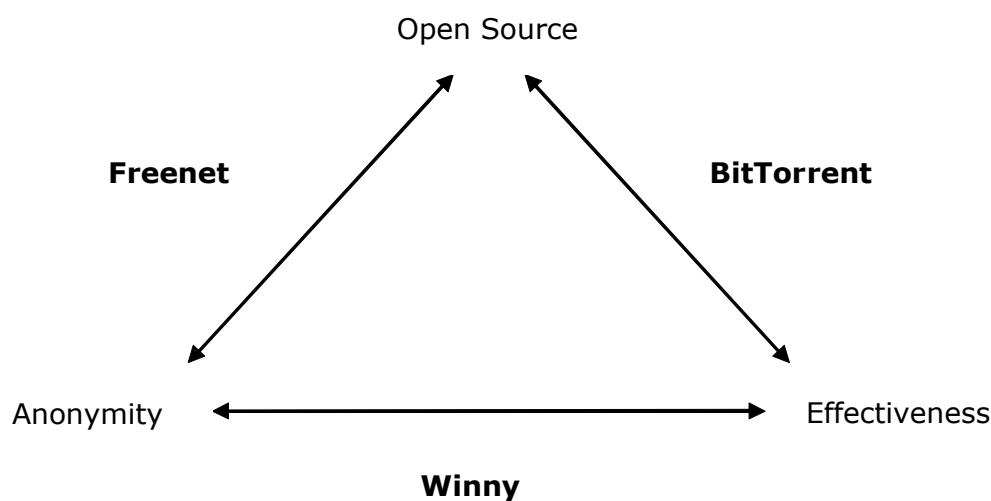


Figure 1: Structure of Trilemma

highly likely that next-generation P2P file software will satisfy all three requirements, and that BitTorrent will be developed further.

Concerning the second issue, relating to the manageability of P2P, it has already been mentioned that no matter what nodes were attacked, the entire network in a pure P2P system would not be affected. Such a system would provide a high resistance to failure. Pure P2P systems, however, offer poor manageability. For example, if a file containing personal information such as a list of names is leaked on Winny, there is no mechanism for an administrator to immediately delete the file. The file continues to be distributed for as long as copies remained in the Winny network. Overseas P2P file-sharing software trials have put most of the blame on users, since there are no pure P2P service administrators and developers and

operating companies cannot be held liable.

Kaneko emphasized that this issue is only a technical defect. He closed his speech by saying that even though he has ideas for creating greater manageability for pure P2P, he cannot start working on improvements for Winny because trials are underway in the courts.⁴

1: According to the dictionary, a hash value is a series of "pseudorandom numbers of fixed length generated from the original text." For example, the hash value function can convert a 40-kilobyte file named "Minutes of meeting.doc" to random numbers of a specified number such as "fb3a0ec36." The hash value would not be overwritten even if the file name changed, unless the file contents were changed. The hash value will be overwritten if even one byte of the

content is changed, even if the file name remains the same. Because of this, it would be possible to confirm whether data had been modified in the course of communication.

2: 2channel faced the threat of a shutting down of its bulletin board in August 2002 because the burden of the servers was excessive. It avoided shutdown when users voluntarily came up with improvements to the bulletin-board program. For more information, refer to "Runaway Internet" Kensuke Suzuki (East Press, 2002).

3: When P2P file-sharing software infringes on copyright it is regarded as an infringement of "the right to make transmittable" under the Japanese Copyright Law. The provision prohibits the act of "making the file available for downloading" as a violation under the copyright law, so uploading is more risky than downloading for the P2P users. Thus, many users become "free riders" who prefer downloading only.

4: Personal data was leaked continuously via Winny following the GLOCOM symposium. Kaneko stated at a meeting of the LSE held in Osaka on March 11, "I hope the Winny network remains sound. Leakage [of personal information] may be prevented by rewriting several lines of the program, but I cannot take any action because of the current situation with the police and prosecutors. I will cooperate if I am asked to do so."

See "Winny Developer Meeting Assertion: Leakage unforeseen and regrettable," MSM Mainichi Interactive, March 11 <<http://www.mainichi-msn.co.jp/keizai/it/24hour/news/20060312k0000m040085000c.html>>.

At a session of the Kaneko's public trial held on March 20, the defense announced that Kaneko is involved with an IT company jointly developing Oztech, a P2P software that uses Winny technology. This software is said to offer new uploading manageability. See "Winny Developer Comes up with Safer Software," MSN Mainichi Interactive, March 20 <<http://www.mainichi-msn.co.jp/today/news/20060321k0000m020066000c.html>>.