2015 IEEE 29th International Conference on Advanced Information Networking and Applications

Study of Filter Sharing Method using Virtual Peers in P2P Networks

Masahiro Sakuma* Yoshihiro Kita* Kentaro Aburada† Mirang Park* Naonobu Okazaki‡
*Kanagawa Institute of Technology
1030 Shimo-Ogino, Atsugi, Kanagawa 243-0292, Japan
†Oita National College of Technology
1666 Oaza-Maki, Oita, 870–0152, Japan
‡University of Miyazaki
1-1 Gakuen-Kibanadai-Nishi, Miyazaki 889-2192, Japan

***Abstract*—Among the problems faced by peer-to-peer (P2P) networks is the malicious content that often spreads unchecked because such contents are transferred from any peer to the other peer directly. While various personal and/or shared countermeasure filtering methods are available, such filters have different configurations and strength levels. Therefore, if malicious content is permitted to penetrate peers with weak filters, they can spread across an entire network. In response, in this paper, we propose a filter sharing method that utilizes virtual peers.**

I. Introduction

Currently, peer-to-peer (P2P) networking is drawing significant amounts of attention and a number of innovative P2P file sharing systems such as Napster[1], Skype[2], Winny[3], and BitTorrent[4] have been introduced. However, P2P networks face problems such as the outflow of personal information and the virus infections that can result when malicious content is uploaded. Furthermore, it is difficult for P2P network users to identify malicious peers and/or content based on the currently used methods[5], [6], [7] that monitor delivered content from received other peers and assigns original reputation values to peers who communicate malicious data. Additionally, since these methods may not be able to detect all of the malicious peers in a network, each peer needs to be capable of identifying and blocking malicious content and peers.

Spam filters[8] remove spam or computer viruses according to given criteria that are based on context, particular words, or suspicious features within content. Such filters use Bayesian filtering[9], which is a collaborative filtering method. Collaborative filters can produce personal recommendations by computing the similarity between the preferences of a given person and those of other people. However, the main bottleneck with existing collaborative filtering systems is the collection of preference information, which means that the systems only become useful after a critical mass of opinions has been collected. This is especially troublesome as most people are not motivated to express their detailed preferences. In P2P networks, where peers are even less likely to provide their preferences, an acceptable collection of preferences cannot be expected. This makes the application of collaborative filtering difficult.

Most P2P file sharing software applications are equipped with filtering capabilities. However, since each peer must set-up and operate those features individually, low-skill-level peers may not want to take the trouble, or may establish unsuitable settings. Thus, it is difficult to achieve harmonized filtering over an entire network.

An existing filter sharing method that has been investigated previously[10] was found to have a weak filter, and was thus unable to protect the network from malicious peers. This method is based on a technique that reinforces personal filters by incorporating a shared filter which is received from other peers. This meant that, even if some peers had no filter-related knowledge, all peers could theoretically achieve the same level of resistance to malicious content. However, because the shared filter settings would still be different for each peer, a portion of such network peers would be unable to block a proportion of the uploaded malicious content. These setting differences would also hinder the network's ability to respond to newly uploaded malicious content because the shared filter upgrade process is different for each peer as well.

In this paper, we propose a filter sharing method which manages creating and upgrading shared filters via a virtualized peer (virtual peer) in an existing P2P network Winny[3] that can be expected to reduce the scattering of shared filter settings for each peer and thus prevent the spread of malicious content.

II. Related Work

A. Filtering in Winny

Winny is one of the most popular P2P network file sharing software applications used in Japan. In this network, when a peer requests content from an upper peer, he or she first must set search criteria. The peer then compares the search criteria with a key that is based on the files held in each peer. The selection key is the recorded values, which can include the file names, file sizes, and identifying values (hash values) observed between files that have the same or similar names.

Winny uses a filtering method built on a filter called an ignore list. The ignore list includes ignore keys for peers who are searching for content. The peer filtering process then uses blacklist methods to block malicious content by excluding any keys that are registered in the ignore list from the search results. However, while using the ignore list in each peer can be expected to prevent the spread of malicious content, it also requires sharing information on malicious content in each peer, which is difficult to achieve because a percentage

1550-445X/15 $31.00 © 2015 IEEE
DOI 10.1109/AINA.2015.259

IEEE computer society

of the network peers will be unable to filter out a portion of the returned malicious content due to differences in their individual ignore lists.

B. Filter sharing method using hash keys

P2P network file sharing applications are equipped with functions that allow users to create filters using lists (blacklists) in which information on blocked contents (filtering information) is described, such as the Winny ignore list. Any peer can create unique filters (personal filters) using such functions. However, those created by users are often weak because both in-depth knowledge on filtering techniques and malicious content are necessary to create effective filters.

A filter sharing method[10] using hash keys has been proposed to solve this problem. This method utilizes a technique whereby a peer receives a shared filter directly from the filter creator, which he or she then uses to reinforce his or her personal filter. To accomplish this, the filter creator first encrypts his or her shared filter, and then sends the hash key (decryption key) to hash key administrator. The hash key administrator is a third party which is independent from both the filter creator and filter requester. Next, the filter requester receives the shared filter from the filter creator, and then receives the hash key from hash key administrator. Finally, the filter requester decrypts the shared filter using the hash key. While this method makes it possible to share strong filters with weak peers and prevent fake shared filters from being received from malicious peers, it has the following problems:

- Diffusion of shared filter effectiveness because of differences in the shared filters produced by different filter creators.
- Filter requesters must query the filter creator for every shared filter upgrade.

In a P2P network, these problems result in countermeasure delays when new malicious contents appear.

C. Content searching using super-node

A content searching method[11] using a super-node has been proposed as an efficient way to search contents in P2P networks. This method utilizes a technique whereby a single peer (called the super-node) manages the network's content index information. In this method, when a search query is received from a peer in the network, the super-node provides information on peers which possess the desired content to the peer originating the search.

The problem with this method is that communication failures are likely to occur because all the search queries are concentrated in the super-node. In addition, anytime the super-node is disconnected, the network loses the entire information index and must then allocate the super-node role to another peer and then rebuild the index information in the new super-node. Super-node virtualization [12] provides a method for solving these problems. This method creates a virtual peer on the network instead of a super-node. Multiple peers then can share and manage a virtual peer, and multiple virtual peers can be created on the network. Furthermore, two beneficial effects can be expected: network load balancing, and the prevention of index information loss when the super-node disconnects. Our method uses super-node virtualization to unify filter sharing.

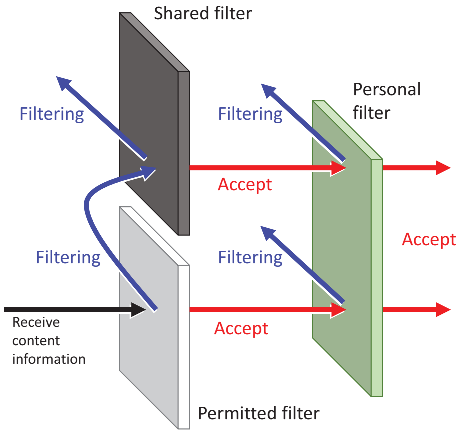

Fig. 1. Content filtering by shared, permitted, and personal filters

III. Proposed Method

A. Our goals

The objective of our method is to ensure that weak peers possessing inadequate personal filter configurations do not receive malicious content. However, if all peers were to share the same shared filter in a P2P network, it would be difficult to create a filter that is universally suitable because requested and hosted contents are different for each peer. Therefore, a cluster is created by collecting network peers which host contents that possess common keywords or genres, after which the virtual peer creates a shared filter specifically for that cluster.

Our goals are as follows:

- All peers will share the same shared filter in the cluster.
- All peers will update the shared filter at the same time.

B. Definition of personal filter and shared filter

In the case of Winny, the user's node makes a key and an encrypted cache file from a file (content) when user uploads the file on network. The information which are included in the key, are as follows[3].

- File name
- File size
- File ID (MD5 hash value includes data of file)
- Publisher information
- Number of reference (and latest access date)
- Bitmap information of cache
- Virsion of key information
- Location information of file

Location information of file includes the file owner's IP adress and port number.

The key is stored the node's memory. When the node receives a query of the file from the other nodes, the key is spreaded to them. The ignore filter information and index of content are used the key in Winny. In our method, the all of filters include the key as filter information.

Personal filters and shared filters are those from which blacklists are formed. Shared filters are independent of personal filters. In order to prevent shared filter tampering, users are prevented from editing the shared filter. Thus, in our method, creating or updating a shared filter must be performed automatically within the virtual peer.

All peers are equipped with a shared filter to ensure that the same filtering can be applied universally. However, since it is likely that the desired content of each peer will be different, it is inevitable that some requested content will be blocked by the shared filter. Furthermore, since our proposed method is directed towards protecting weak and vulnerable peers, a strong filter is required. Conversely, skilled peers with advanced filtering knowledge may be inconvenienced if the settings on the shared filter are too strong. Accordingly, we propose permitting the shared filter to be relaxed under certain conditions.

The permitted filter list (white list) describes contents that match the permitted criteria. Permitted criteria describe content-related information such as file names, file sizes, and hash values, as well as filter criteria. The initial state of a permitted filter is an empty list, and each user can add or remove items to his or her list freely.

Fig. 1 shows content filtering by the permitted, shared, and personal filters. First, received contents are filtered by the permitted filter. If none of the contents are blocked, they are then filtered by the shared filter. Next, the contents which have passed the permitted and shared filters are examined by the personal filter. As a result, the peer only receives contents which have passed the permitted personal, shared, and permitted filters.

C. Virtual peer construction

In Winny, each peer has metrics that are used to set connection priorities to other peers. The peers use these metrics to sift through the peers who are available for connection. Connection priorities are base on the peers who score high when the following conditions are applied[3]:

- Peers which match the search keywords or categories
- Peers which have successfully downloaded sought after contents.
- Peers which are linked to local peers.

Based on such conditions, local peers will maintain links with peers who have high connection priorities and attempt to reach a stable state in which they can connect to peers who have similar tastes and priorities via short hops. As a result, a cluster consisting of a set of peers with the same tastes is formed.

In our method, shared filter management is performed via the virtual peer which is constructed in the cluster. Virtual peers are constructed and managed by groups of two or more peers. In Winny, peers are divided into upper and lower peers based on network speed. Since the upper peers have consistently higher network connection speeds, we felt it would be appropriate that they be tasked with managing the virtual peer in order to ensure the shared filter is updated frequently. Additionally, since we consider it important that stable peers manage the shared filter, the following conditions are used to select the peers which are used to construct the virtual peer:

- Upper peers.
- Peers who have accumulated long active periods in the cluster.

Among the peers available to construct the virtual peer, the peer leader is the one which has accumulated the most time, and the other peers in the clusters are member peers. If there are two or more peers with the same accumulated time, a leader is selected at random from those peers. Member peers are responsible for the following:

- Personal filter collection.
  Member peers collect personal filters from lower (affiliated) peers which are connected directly or indirectly to member peers.
- Sharing the shared filter between member peers.
  After it is created or updated, member peers share the shared filter with other member peers.
- Delivery of the shared filter to affiliated peers.
  Member peers deliver the shared filter to their affiliated peers.
- Delivery of filtering information to leader peer.
  Member peers receive filtering information from affiliated peers and delivery it to their leader peer.

In addition to behaving as a member peer, the leader peer is responsible for the following roles:

- Shared filter creation
  The leader peer collects personal filters from member peers and creates the shared filter.
- Shared filter updating
  The leader peer receives filtering information from member peers and incorporates it into the shared filter.
- Member peer management
  The leader peer manages the number of member peers in order to maintain the stability of the virtual peer. When member peers are disconnected, the leader peer examines the affiliated peers and adds those with the longest accumulated time as new member peers.

When a leader peer disconnects, the member peer which has the longest accumulated time in the cluster becomes the new leader peer.

We define the number of leader peers LP and member peers MP, in order to construct and maintain virtual peer VP as follows. The total number of peers in a cluster is P_{max}, the

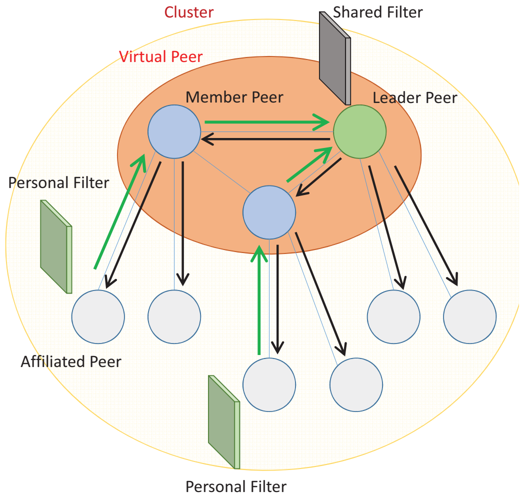

Fig. 2. Creation of a shared filter in a cluster

TABLE I. SYMBOLS WHICH ARE USED IN PROTOCOLS

Symbols	Means
C^k	Clusters
	$(1 \leq k \leq K)$
K	Number of clusters
P_x^k	Peers in C^k
	$(1 \leq x \leq X^k)$
X^k	Number of peers in C^k
LP^k	Leader peer in C^k
MP_n^k	Member peers in C^k
	$(1 \leq n \leq N^k)$
N^k	Number of member peers in C^k
$AP_m^{k_n}$	Affiliated peers join to MP_n^k
	$(1 \leq m \leq M_n^k)$
M_n^k	Number of affiliated peers join to MP_n^k
PsF_x^k	Personal filters are owned by P_x^k
$PsF_x^k(E_t)$	A filter element E_t in PsF_x^k
	$(1 \leq t \leq T_x^k)$
T_x^k	Number of filter elements in PsF_x^k
PmF_x^k	Permitted filters are owned by P_x^k
SF^k	Shared filters of C^k
$SF^k(old)$	Unupdate shared filters
$SF^k(new)$	Updated shared filters
$Agg(PsF^k)_n$	An aggregate of PsF_x^k is collected
	from $AP_m^{k_n}$
$LT(SF^k)$	Life time of SF^k
$A \rightarrow B$	Transmit filters (or contents) from A to B

maximum number of affiliated peers which are connected via a member peer, is AP_{max}.

$$\begin{aligned} VP &= LP + MP \\ LP &= 1 \\ MP &= \frac{P_{max}}{AP_{max}} \end{aligned}$$

D. Proposed filter sharing method

Fig. 2 shows the process of creating a shared filter in a cluster. First, the leader peer and member peers in a cluster set up the virtual peer. Next, the leader peer creates the shared filter and delivers it to member peers and their affiliated peers. We propose the following protocols for shared filter creation and delivery. Table I shows the symbols which are used in the protocols.

1) $AP_m^{k_n} \rightarrow MP_n^k \ : \ PsF_x^k$
Each affiliated peer $AP_m^{k_n}$ sends its own personal filter PsF_x^k to a member peer MP_n^k.
2) $MP_n^k \ : \ Agg(PsF^k)_n = \sum_{i=1}^{M_n^k} PsF_i^k + PsF_n^k$
The member peer MP_n^k summarizes the collected personal filters to a single filter PsF_n^k as $Agg(PsF^k)_n$. Any duplicated filtering information is summarized.
3) $MP_n^k \rightarrow LP^k \ : \ Agg(PsF^k)_n$
MP_n^k sends $Agg(PsF^k)_n$ to the leader peer LP^k.
4) $LP^k \ : \ SF^k = \sum_{i=0}^{N^k} PsF_i^k$
LP^k creates the shared filter SF^k from the personal filters PsF_x^k.
5) $LP^k \ : \ Set\ LT(SF^k)\ to\ SF^k$
LP^k sets the lifetime $LT(SF^k)$ of the shared filter SF^k. $LT(SF^k)$ is explained in section III-E.
6) $LP^k \rightarrow MP_1^k, MP_2^k, ..., MP_n^k \ : \ SF^k$
The leader peer LP^k delivers the shared filter SF^k to all member peers.
7) $MP_j^k \rightarrow AP_1^{k_j}, AP_2^{k_j}, ..., AP_m^{k_j} \ : \ SF^k$
Each member peer MP_j^k delivers the shared filter SF^k to its own affiliated peers.
8) If $LT(SF^k)$ is complete, the process returns to 1).

When a peer searches for content, it affiliates with a cluster which lists common keywords or genres. Simultaneously, the peer affiliates with member peers which have fewer than the maximum number of affiliated peers, and receives the shared cluster filter from a neighboring peer. When the peer is disconnected from the cluster, the shared cluster filter is deleted from that peer.

Therefore, even if the personal filter is weak, content can be delivered safely via the shared cluster filter. Additionally, even if a peer comes and goes frequently, it performs its own shared filter management by installing or deleting the filter when appropriate.

E. Shared filter updates

Because new malicious contents are being generated all the time, shared filters must be updated frequently. Malicious contents which passed through the shared filter are defined as new malicious contents. Fig. 3 shows the shared filter process. The protocols used for shared filter updates are as follows:

1) $P_x^k \rightarrow LP^k : PsF_x^k$

When a peer P_x^k blocks malicious content which passed through the permitted filter and shared filter using its personal filter, the peer then sends the personal filter PsF_x^k to the leader peer LP^k.
2) $LP^k : SF^k(new) = SF^k(old) + PsF_x^k$
The leader peer LP^k then incorporates the received filtering information into its own shared filter $SF^k(old)$, and updates it as $SF^k(new)$.
3) $LP^k \rightarrow MP_1^k, MP_2^k, ..., MP_{N^k}^k \ : \ SF^k(new)$
The leader peer LP^k then delivers the updated shared filter to all member peers in the cluster.

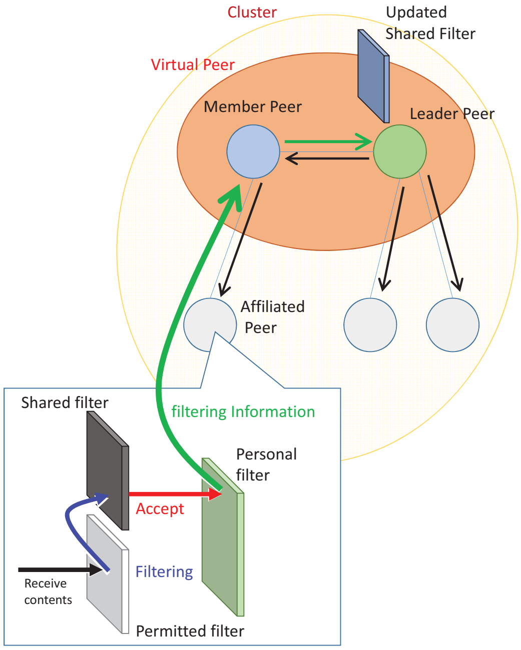

Fig. 3. Shared filter updating process

4) $MP_n^k \rightarrow AP_1^{k_n}, AP_2^{k_n}, ..., AP_{M_n^k}^{k_n} : SF^k(new)$
Each member peer then delivers the updated shared filter to its affiliated peers.

If the leader peer were to continue adding filtering information into its shared filter, the filter would become excessively large. Such enlarged shared filters can result in network congestion. Accordingly, it is necessary to define the lifetimes of shared filters so that they can be removed automatically when their lifetimes expire. At such times, the cluster leader peer collects personal filters from all peers in the cluster, creates a new shared filter, and then distributes it to its peers.

F. Filter sharing and updating of affiliated peers which belong to multiple clusters

Since Winny makes it is possible for peers to belong to multiple clusters, we defined a rule that requires each peer to receive the shared filters created for each cluster it joins.

Fig. 4 shows the process used to create a shared filter in cases where a peer belongs to multiple clusters. In such situations, the affiliated peer sends personal filter information to all member peers it is connected with, and then receives the shared filters of each cluster in return.

Fig. 5 shows the process used to update the shared filter in cases where a peer belongs to multiple clusters. First, contents are delivered between peers belonging to the same cluster. For example in Fig. 5, a peer receives Content X from a peer which belongs to Cluster B. This peer is filtering content X using the Cluster B shared filter. If Content X passes through the shared filter and is blocked by the personal filter, the peer sends the Content X filtering information to the Cluster B virtual peer. Then, the Cluster B leader peer incorporates the received filtering information into its own shared filter, and then delivers the updated shared filter to all Cluster B peers.

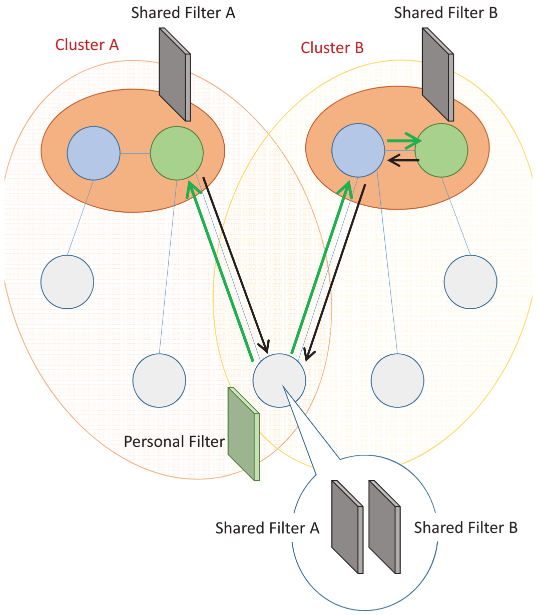

Fig. 4. Shared filter creation in cases where a peer belongs to multiple clusters

TABLE II. DEFINED VALUES IN A SIMULATOR

Items	Values
Number of the all of peer	1,000
Number of member-peers include leader-peer	100
Number of contents which are haved each peer at the initial state	5
Number of the all of content requests	10,000
Number of peer $P(C_x)$ who holds malicious content C_x	1
Number of peer $P(F_x)$ who holds the filter F_x for filtering C_x	10
Number of kinds x of malicious content	10

In this manner, even if a peer belongs to multiple clusters, it can maintain updated shared filters. These updated shared filters permit peers with weak personal filters to handle content safely.

IV. Evaluation and Discussion

A. Simulator-based filter sharing experiment

In order to evaluate the utility of our proposed methods, we conducted experiments on filter sharing methods using a P2P network simulator. The results of these experiments clarified the following two points:

- The utility of preventing malicious content dissemination via shared filter unification.
  In our proposed method, all peers will be able to achieve harmonized filtering because weak peers which can compromise the strength of a shared filter are strengthened by the unification of shared filters in a cluster. Thus, we can expect to prevent the dissemination of malicious content.

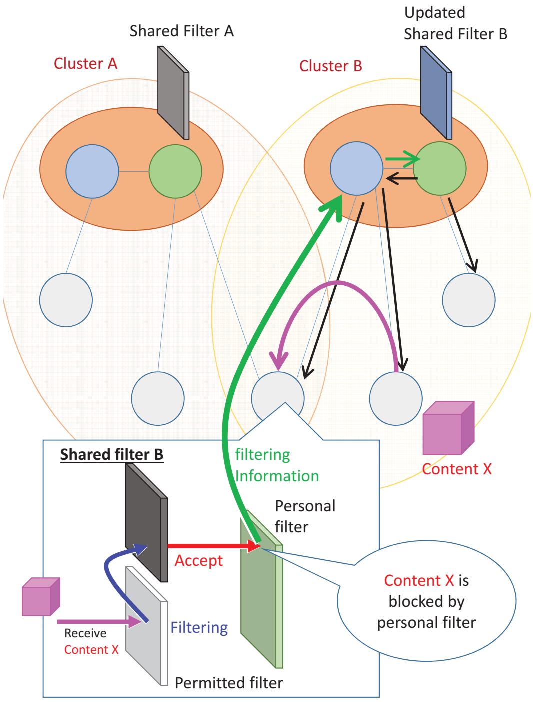

Fig. 5. Shared filter updating in cases where a peer belongs to multiple clusters

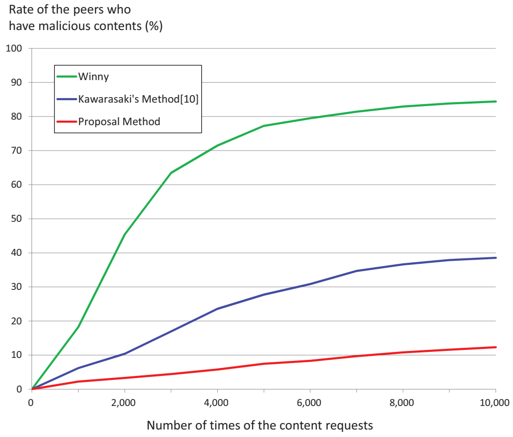

Fig. 6. Number of peers who hold C_x

- The practicality of simultaneous shared filter updating. Shared filters must be updated continuously in order to provide countermeasures to new malicious content. However, when new malicious content is disseminated, all peers will be able to update their shared filters simultaneously when the leader peer delivers the updated shared filter to all cluster peers. Thus, we can expect the quick imposition of countermeasures to prevent the dissemination of new malicious content.

TABLE II shows the defined values in a simulator, which uses the Winny P2P network structure. We have evaluated and compared our proposed method with the two methods: the filter un-sharing method in Winny and an existing filter sharing method (Kawarasaki's Method)[10].

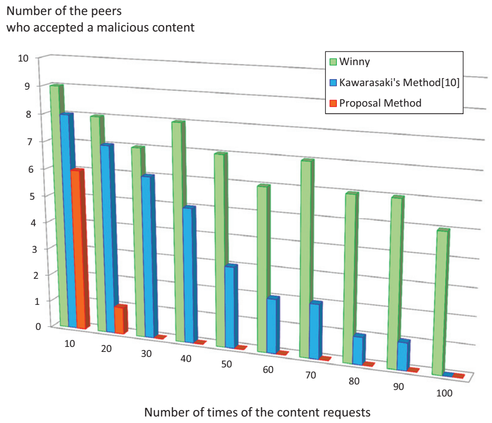

Fig. 7. Number of peers who accepted C_x

The evaluation of all clarified points is as follows:

- The ability to prevent malicious content dissemination via shared filter unification.
  We count the total number of peers who hold malicious contents as well as each content request in a cluster. These peers do not include the peer who was the source of the malicious content.
- The practicality of simultaneous shared filter updating.
  We count the total number of peers who accepted each malicious content C_x request.

Figure6 shows the total number of peers holding C_x. In the case of Winny, it was confirmed that the peers holding C_x are responsible for increasing the numbers of content requests in a cluster, and that almost all peers hold C_x, in addition to the peers using the F_x filter to clean C_x. In the case of Kawarasaki's Method, it could be confirmed that the number of peers holding C_x did not exceed that of Winny, but approximately 40% of those peers continue to hold C_x. In the case of our proposed method, it was possible to confirm that the peers holding C_x can be maintained at a level of approximately 10% in the peers who holds C_x. Therefore, our proposed method is useful method for limiting the spread of malicious contents.

Figure7 shows the number of peers who accepted C_x. In the cases of Winny and Kawarasaki's Method, it can be confirmed that the peer who accepted C_x exists at the request of C_x. In the case of our proposed method, it can be confirmed that approximately 20 peers initially accepted the C_x requests, but that all peers would later refuse to accept C_x requests. Therefore, our proposal method provides a practical countermeasure against the spread of new malicious contents.

B. Discussion

In the existing filter sharing method[10], in which the filter is shared throughout the whole P2P network and is managed by each peer, both its massive scattering and update delays weaken the shared filter. In contrast, our proposed method can be expected to eliminate the need to provide shared filter information to each peer, and to prevent the spread of malicious content by fast, automatic filter updating.

Since attackers can intentionally block the filtering information using their personal filter in attempts to enlarge the shared filter. Shared filter expansion problems must be rapidly solved in order to prevent network congestion.

Our proposed method can minimize shared filter enlargement by setting the life time of filtering information. However, this method does not always provide an adequate solution. Therefore, it is necessary to combine it with keyword similarity and to set filter information limits in order to prevent the shared filters from becoming excessively enlarged. If the maximum amount of filter information is exceeded, the leader peer is informed of the need to reconstruct the shared filter.

It is known that attackers sometimes incorporate malicious content into the permitted filters owned by other peers. In our proposed method, since the personal filter is used to block malicious content that has successfully passed through a permitted filter, peers may suffer damage if the personal filter cannot block the malicious content. To solve this problem, it is advisable to periodically initialize the permitted filter. However, since this can lead to a decrease in usability, it is necessary to promote user understanding with respect to such initialization, and to carefully set the permitted filter initialization frequency.

Furthermore, it is necessary to evaluate filter sharing and the updating of affiliated peers who belong to multiple clusters. This evaluation can confirm that our proposal method can be applied to the filtering of malicious contents for large-scale P2P networks.

V. Conclusion

In this paper, we proposed a method for creating and updating shared filters via virtual peers in order to prevent both shared filter weakening and update delays. We confirmed the effectiveness of our filter sharing method via shared filter experiments on a P2P network simulator. Our results showed that it is possible to suppress the spread of malicious content and simultaneously solve the shared filter weakness problem via proposed method.

In our future work, it will be necessary to address the problems described below:

- Evaluation of the applicability of filter sharing in a peer which belongs to multiple clusters.
- Finding ways to respond to attacks which intentionally increase the amounts of filtering information in a shared filter.
- Finding ways to respond to attacks which include malicious content into permitted filter information.
- Finding ways to respond to attacks which increase the shared filter management load on peers affiliated with multiple clusters.

References

[1] T., Merriden: Irresistible Forces: The Business Legacy of Napster and the Growth of the Underground Internet, Capstone Inc., ISBN:978-1-84112-170-3, 2001.

[2] Skype, Microsoft Inc.,
http://www.skype.com/

[3] I., Kaneko: Technology of Winny, ASCII Media Works Inc., ISBN:4-7561-4548-5, 2005 (in Japanese).

[4] BitTorrent, Bittorrent Inc.,
http://www.bittorrent.com/

[5] L., Mekouar, Y., Iragi, and R., Boutaba: Detecting Malicious Peers in A Reputation-Based Peer-to-Peer System, The 2nd IEEE Consumer Communication and Networking Conference (CCNC), pp.37-42, 2005.

[6] X., Jin and S.H., Garychan: Detecting Malicious Nodes in Peer-to-Peer Streaming by Peer-Based Monitoring, ACM Transactions on Multimedia Computing, Communications and Applications, Vol.6, No.2, Article 9, pp.1-18, 2010.

[7] X., Wei, T., Ahmed, M., Chen, and A.K., Pathan: SMART: A Subspace based Malicious Peers Detection algorithm for P2P Systems, International Journal of Communication Networks and Information Security (IJCNIS), Vol.5, No.1, pp.1-9, 2013.

[8] A., Gray and M., Haahr: Personalised, Collaborative Spam Filtering, 1st Conference on Email and Anti-Spam (CEAS), pp.1-8, 2004.

[9] D., Fox, J., Hightower, L., Liao, D., Schulz, and G., Borriello: Bayesian Filtering for Location Estimation, Journal of IEEE Pervasive Computing, Vol.2, Issue 3, pp.24-33, 2003.

[10] M., Kawarasaki and K., Ibuki: Filter Sharing Method to Suppress Harmful Content Diffusion over P2P Networks, The 3rd International Conference on Latest Advance in Network (ICLAN'2008), pp.63-68, 2008.

[11] L., Xiao, Z., Zhuang, and Y., Liu: Dynamic Layer Management in Superpeer Architectures, IEEE Transactions on Parallel and Distributed Systems, Vol.16, No.11, pp.1078-1091, 2005.

[12] M., Shikano, T., Ueda, K., Abe, H., Ishibashi, and T., Matsuura: Communication Methods for Virtual Peers on musasabi P2P Platform, Technical Report 2, IPSJ DPS-139, 2009 (in Japanese).