RESEARCH ARTICLE OPEN ACCESS

Copyright Protection in Peer To Peer Network

[1]Siraj Ul Muneera S, Yoga Lakshmi M[1], Dr. Ashwini K [2],
[1]Sr. Assistant Professor Department of Information Science and Engineering, New Horizon College of Engineering Bangalore, India.
[2]Department of Information Science and Engineering, New Horizon College of Engineering Bangalore, India

ABSTRACT
Recently, P2P technology has become so popular that a large number of files are being exchanged by millions of users concurrently. However, due to the significant growth of P2P file sharing, even copyrighted files are also actively exchanged so copyright infringement has become a serious issue. In particular, P2P file sharing is recognized as a killer application for the P2P technology, and several P2P file sharing systems have been used by many users from the late 90's to the present, including Napster, Gnutella, Kazaa and Bit Torrent. However, the popularization of such P2P file sharing causes several undesired issues in recent years, such as the illegal sharing of copyrighted contents violating the copyright law and the act as a hotbed of cyber-crimes such as phishing scams and leaks of personal information. In this work, we propose a solution to ensure copy right violation in P2P systems. Keywords- Peer to Peer systems, copyright infringement, file sharing, digital rights management, pollution attack, index poisoning, watermarking and fingerprinting.

I. INTRODUCTION

Peer-to-Peer (P2P) technology has been widely used in many fields including IP-phone, live streaming, and file sharing. In particular, P2P file sharing is recognized as a killer application for the P2P technology, and several P2Pfile sharing systems have been used by many users from the late 90's to the present, including Napster, Gnutella, Kazaa and Bit Torrent. However, the popularization of such P2P file sharing causes several undesired issues in recent years, such as the illegal sharing of copyrighted contents violating the copyright law and the act as a hotbed of cyber-crimes such as phishing scams and leaks of personal information. Thus far, many challenges have been conducted to restrain the download of illegal files by anonymous users. DRM (Digital Rights Management) encodes contents using a specific encoding technique so that it could be decoded merely by using specific software and/or hardware. An example of DRM is Windows Media DRM1, which requests a private key for playing back encoded contents which is individually issued for each paid customer. The detection of illegal file sharing has also been investigated extensively. For example, we can identify a user who illegally leaks paid contents to unauthorized users with the aid of digital watermarking and can identify users who illegally share paid contents by deploying a decoy peer.

1.1 Motivation

Index and content poisoning have been proposed to control the exchange of copyrighted content. Unfortunately, however, such control methods are costly in terms of the amount of control traffic since they apply the control method to all the peers and they generate much control traffic. So the objective of our work is to reduce the cost of index poisoning in terms of number of peers searched and updated

1.2 EXSISTING SYSTEMS AND LIMITATIONS

1.2.1. "Pollution in P2P File Sharing Systems".
Author name: J. Liang, R. Kumar, Y. Xi, and K. W. Ross.
Publication: In Proc. the 24th IEEE International Conference on Computer Communications (INFOCOM), 2005.
Brief: This scheme works by crawling the entire p2p network where it is polluted. We develop an automated procedure to detect whether a given version is polluted or not, and we show that the probabilities of false positives and negatives of the detection procedure are very small.
Limitation: Replacing the entire content in the network is very costly. 1.2.2. "The Index Poisoning Attack in P2P File Sharing Systems".

1.2.2. "The Index Poisoning Attack in P2P File Sharing Systems".
Author name: J. Liang, N. Naoumov, and K. W, Ross.
Publication: In Proc. the 25th IEEE International Conference on Computer Communications (INFOCOM), 2006.
Brief: One of the leading sources of disruptions in the P2P file sharing systems is the index poisoning attacks. This attack seeks to corrupt the indexes

used to reference files available for download in P2P systems with false data. In order to protect the users from these attacks it is important to find solutions to eliminate or mitigate the effects of index poisoning attacks.
Limitation: It is hard understanding where and who have generated the attack.

1.2.3. "An Effective Index Poisoning Algorithm for Controlling Peer-to-Peer Network Applications".
Author name:
P. Putra and A. Nakao
Publication:
In Proc. International Workshop on Modeling, Analysis, and Control of Complex Networks (Cnet '11), 2011.
Brief: Copyright infringement is considered a significant issue in P2P network communications. Index and content poisoning have been proposed to control the exchange of copyrighted content. Unfortunately, however, such control methods are costly in terms of the amount of control traffic since they apply the control method to all the peers and they generate much control traffic. In general, directly applying index poisoning to a peer may indirectly poison the neighboring peers, thus, it is possible to reduce the number of peers to target for the same effect of poisoning.
Limitation: This method can lift the limitation in the number of controllable files in conventional index poisoning since it minimizes the control traffic but still achieves the same effectiveness with the existing work.

1.2.4. "Controlling File Distribution in Winny Network through Index Poisoning".
Author name: M. Yoshida, S. Ohzahata, A. Nakao, and K. Kawashima.
Publication: In Proc. International Conference on Information Networking (ICOIN 2009), 2009.
Brief: Peer-to-peer (P2P) file sharing networks have witnessed dramatic increase in popularity for the past few years. In order to meet the demand of users, most P2P file sharing networks have been primarily focusing on improving transmission efficiency and network scalability. However, these networks do not usually have management mechanisms for distributing files in general. Consequently, copyright infringements in P2P file sharing networks have become prevalent. In order to prevent illegal file distribution, several anti-P2P companies have controlled the file distribution by index poisoning.
Limitation: The result of this method shows that the index poisoning could cause a serious damage on a Winny network, when many files are under our control simultaneously.

1.3 PROPOSED SYSTEM
Here, we propose a cost-effective index poisoning scheme for unstructured P2P file sharing systems. The proposed scheme consists of the following four components:
1) The first component identifies the direction of the flow of indices and virtually partitions the set of peers into three parts as upstream peers, midstream peers, and downstream peers.
2) The second component periodically injects copies of altered indices with short lifetime to the owner of illegal files with time interval T1.
3) The third component periodically injects copies of altered indices with long lifetime to upstream peers with time interval T2.

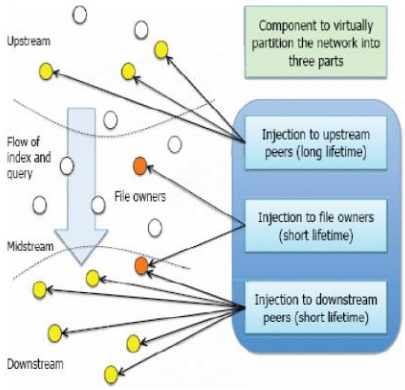


Fig 1.3.1: Overview of the proposed scheme.

4) The fourth component periodically injects copies of altered indices with short lifetime to downstream peers with time interval T3. The reader should note that in the third and the fourth factors described above, while keeping the required cost sufficiently low. As for the metrics for the cost of index poisoning schemes, we focus on the following two factors:
• Network cost which is evaluated by the total number of messages as well as the load of agent to inject altered copies to the network.
• Spatial cost which is evaluated by the number of altered copies existing in the network. To reduce the network cost, we should design a scheme so that altered copy will reach peers holding the correct copy through distributed propagation, and to reduce the spatial cost, we should design a scheme so that the lifetime of each copy is determined in such a way that altered copy disappears after arriving at the target peer.

1.3.1 Benefits of the proposed system
Copy right can be enforced by index poisoning in a cost effective manner. The spatial cost can be reduced to one third and the network cost to a half.

II. SYSTEM REQUIREMENTS AND SPECIFICATIONS

Software requirement Specification is a fundamental document, which forms the foundation of the software development process. It not only lists the requirements of a system but also has a description of its major feature. An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies at a particular point in time (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time. The SRS also functions as a blueprint for completing a project with as little cost growth as possible. The SRS is often referred to as the "parent" document because all subsequent project management documents, such as design specifications, statements of work, software architecture specifications, testing and validation plans, and documentation plans, are related to it. It is important to note that an SRS contains functional and nonfunctional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

2.1 Functional Requirement

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements: -

2.1.1 A user can approach the system to poison their copyrighted content.
2.1.2 The seed of copyrighted content must be located.
2.1.3 Poison the index at seed.
2.1.4 Identify upstream nodes and poison index at upstream.
2.1.5 Identify downstream nodes and poison index at downstream.
2.1.6 Measure network cost.

2.2 Non-functional Requirement

Nonfunctional requirements are the requirements which are not directly concerned with the specific function delivered by the system. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Nonfunctional requirements arise through the user needs, because of budget constraints, organizational policies, the need for interoperability with other software and hardware systems or because of external factors such as:

1. Product Requirements 2. Organizational Requirements 3. User Requirements 4. Basic Operational Requirements

2.2.1 Product Requirements Portability: The system must work for any operating system.
Correctness: The index of copyright content alone must be poisoned.
Ease of Use: The software must be easy to use.
Modularity: The design must be modular.
Robustness: The system should not hang or crash.
Nonfunctional requirements are also called the qualities of a system. These qualities can be divided into execution quality & evolution quality. Execution qualities are security & usability of the system which are observed during run time, whereas evolution quality involves testability, maintainability, extensibility or scalability.

2.2.2 Organizational Requirements

Process Standards: IEEE standards are used to develop the application which is the standard used by the most of the standard software developers all over the world.
Design Methods: Design is one of the important stages in the software engineering process. This stage is the first step in moving from problem to the solution domain. In other words, starting with what is needed design takes us to work how to satisfy the needs. The design of the system is perhaps the most critical factor affecting the quality of the software and has a major impact on the later phases, particularly testing and maintenance. We have to design the product with the standards which has been understood by the developers of the team.

2.2.3 User Requirements

- The copyright logs must be generated and stored.
- Graphical elements for time spent in poisoning must be drawn.

2.2.4 Basic Operational Requirements

The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, will be related to these following points: -
Mission profile or scenario: It describes about the procedures used to accomplish mission objective. It also finds out the effectiveness or efficiency of the system.

Performance and related parameters: It points out the critical system parameters to accomplish the mission **Utilization environments:** It gives a brief outline of system usage. Finds out appropriate environments for effective system operation.
Operational life cycle: It defines the system lifetime.

2.3 Resource Requirement
Netbean IDE 7.2: Netbean is a multi-language software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written primarily in Java and can be used to develop applications in Java and, by means of the various plug-ins, in other languages as well, including C, C++, COBOL, Python, Perl, PHP, and others. Netbean employs plug-ins in order to provide all of its functionality on top of (and including) the runtime system, in contrast to some other applications where functionality is typically hard coded. The Netbean SDK includes the Netbean java development tools (JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace, in this case a set of metadata over a flat file space allowing external file modifications as long as the corresponding workspace "resource" is refreshed afterwards.
Swing: The Java Foundation Classes (JFC) consists of five major parts: AWT, Swing, and Accessibility, Java 2D, and Drag and Drop. Java 2D has become an integral part of AWT, Swing is built on top of AWT, and Accessibility support is built into Swing. The five parts of JFC are certainly not mutually exclusive, and Swing is expected to merge more deeply with AWT in future versions of Java. Swing is a set of classes that provides more powerful and flexible components than are possible with the AWT. In addition to the familiar components, Swing supplies tabbed panes, scroll panes, trees, and tables. It provides a single API capable of supporting multiple look-and feels so that developers and end-users are not locked into a single platform's look-and-feel. The Swing library makes heavy use of the MVC software design pattern, which conceptually decouples the data being viewed from the user interface controls through which it is viewed. Swing possesses several traits such as—
1. Platform independence
2. Extensibility
3. Component-oriented
4. Customizable
5. Configurable
6. Look and feel.

Platform independence both in terms of its expression and its implementation, extensibility which allows for the "plugging" of various custom implementations of specified framework interfaces Users can provide their own custom implementation of these components to override the default implementations. Component-orientation allows responding to a well-known set of commands specific to the component. Specifically, Swing components are Java Beans components, compliant with the Java Beans Component Architecture specifications. Through customizable feature users will programmatically customize a standard Swing component by assigning specific borders, colors, backgrounds, opacities, etc., configurable that allows Swing to respond at runtime to fundamental changes in its settings. Finally look and feel allows one to specialize the look and feel of widgets, by modifying the default via runtime parameters deriving from an existing one, by creating one from scratch, or, beginning with J2SE 5.0, by using the Look and Feel which is configured with an XML property file.

2.4 Hardware Requirements
We need 1 machine with following minimal requirements

CPU	: Intel 2.2 GHZ
Memory	: 2GB
Disk	: 40 GB
Display	:15-inch color

2.5 Software (Tools &Technologies) Requirements
Following are the requirements

Coding	:java
Simulation	: p2p network sim
Platform	: jdk 1.7
OS	: Windows 7 or 8 or 10
Dev Tool	: Netbean IDE 7.2

III. SYSTEM ANALYSIS

Analysis is the process of finding the best solution to the problem. System analysis is the process by which we learn about the existing problems, define objects and requirements and evaluates the solutions. It is the way of thinking about the organization and the problem it involves, a set of technologies that helps in solving these problems. Feasibility study plays an important role in system analysis which gives the target for design and development.

3.1 Feasibility Study
Depending on the results of the initial investigation the survey is now expanded to a more detailed feasibility study. "FEASIBILTY STUDY" is a test

of system proposal according to its workability, impact of the organization, ability to meet needs and effective use of the resources. Eight steps involved in the feasibility analysis are:

- ✓ Form a project team and appoint a project leader.
- ✓ Enumerate potential proposed system.
- ✓ Define and identify characteristics of proposed system.
- ✓ Determine and evaluate performance and cost effective of each proposed system.
- ✓ Weight system performance and cost data.
- ✓ Select the best proposed system.
- ✓ Prepare and report final project directive to management

Three key considerations involved in the feasibility analysis are

- ✓ ECONOMICAL FEASIBILITY
- ✓ TECHNICAL FEASIBILITY
- ✓ SOCIAL FEASIBILITY

3.1.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

3.1.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement; as only minimal or null changes are required for implementing this system.

3.1.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

IV. SYSTEM DESIGN

Design is a creative process; a good design is the key to effective system. The system "Design" is defined as "The process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization". Various design features are followed to develop the system. The design specification describes the features of the system, the components or elements of the system and their appearance to end-users.

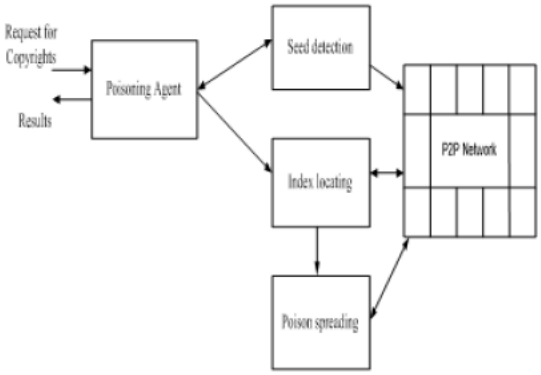
4.1 System Architecture

System architecture is the conceptual design that defines the structure and behavior of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system. The System architecture is shown. The modules are showed in diagram, they are

Poisoning agent: In this module, owner will request for the copyrights so that he can check the copyrights in the various seeds. And display the results whether the network is poisoned or not.

Seed detection: In this module seeds are going to be detected as poison if any copyrights detected in the network. Index locating: In this module the location of index to be put into the P2P network. There network got splitted.

Poison spreading: In this module, if any index is poisoned it spread to the entire P2P network.

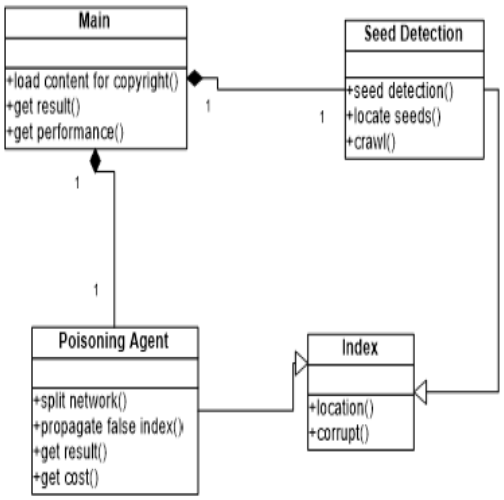


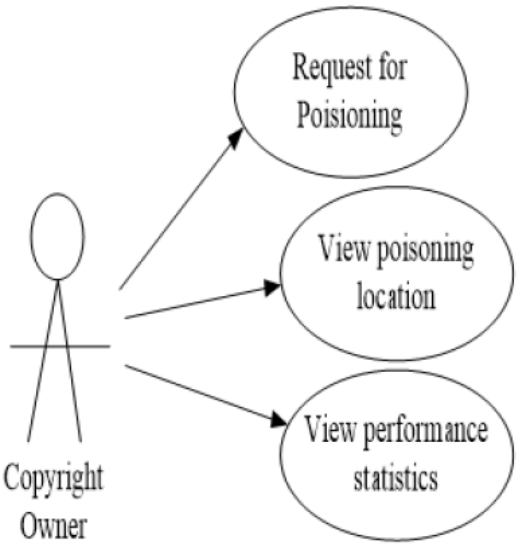
4.2 Classes Designed for the system

A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes.

The class diagram is shown below.

Seed Detection: This class has operations called seed detection, locate seeds and crawl.
Index: This class has operations called location and corrupt.
Poisoning Agent: This class has operations called split network, propagate false index, get result and get cost. **4.3 Use case Diagram of the system**
A use case diagram is a type of behavioural diagram created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. In the below mentioned use case diagram the copyright owner is referred as actor who is responsible for performing various operations such as request for poisoning, view poisoning location and view performance statistics.

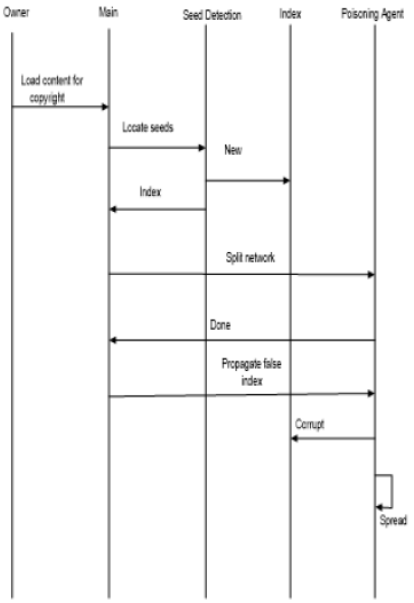




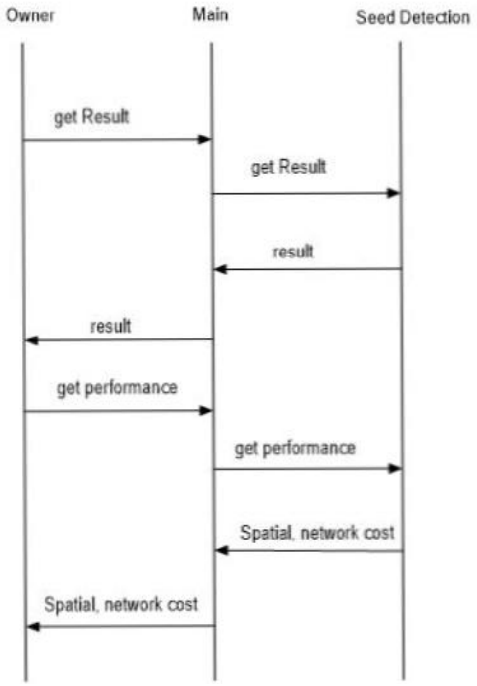
The class diagram has the following classes
Main: This class has operations called load content for copyright, get result and get performance.

4.4 Sequence diagram of system operation
A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart.
The sequence diagrams show below.

4.4.1 Sequence Diagram for poisoning flow
Here Owner, Main, Seed Detection, Index, and Poisoning Agent are objects. Each object interacts with other objects in a sequential order through messages. As shown above.



4.4.2 Sequence Diagram for view result flow
Here Owner, Main and Seed Detection are objects. Each object interacts with other objects in a sequential order through messages. As shown above.

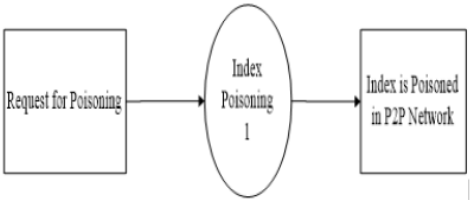


4.5 Data Flow Diagram of the system

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

4.5.1 Level 0 Data flow diagram

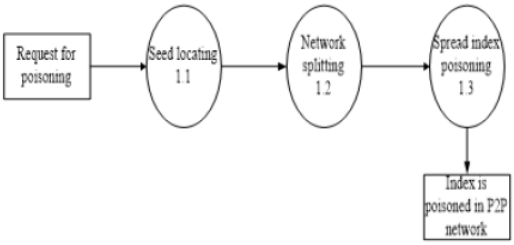
A context-level or level 0 data flow diagram shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modelled purely interms of data flows across the system boundary. The context diagram shows the entire system as a single process, and gives no clues as to its internal organization.



Here in level 0, request for poisoning is taking as input and index is poisoned in P2P network is taken as output. The first main process is Index Poisoning.

4.5.2 Level 1 Data flow diagram

The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. It also identifies internal data stores that must be present in order for the system to do its job, and shows the flow of data between the various parts of the system.



The level 1 worked with the sub processes of first main process. In the above data flow diagram, the sub process is seed locating, network splitting, spread index poisoning. Index is poisoned in P2P network is referred as output.

V. CONCLUSION

First section briefly discussed about the existing system and their limitations. It also discusses about the outcomes of the proposed system. Then the second section gives details of the functional requirements, non-functional requirements, resource requirements, hardware requirements, software requirements etc. Again the non-functional requirements in turn contain product requirements, organizational requirements, user requirements, basic operational requirements etc. Third section is to find out whether the system is feasible enough or not. For these reasons different kinds of analysis, such as performance analysis, technical analysis, economical analysis etc. is performed. Forth section mainly concentrates on system architecture, class diagram, sequence diagram, use-case diagram, data flow diagram etc.

REFERNCES

[1]. R. J. Anderson, “Security Engineering: A Guide to Building Dependable Distributed Systems”, Wiley, 2008.

[2]. J. Kong, W.-D. Cai, and L. Wang, “The evaluation of index poisoning in Bit Torrent”, In Proc. the 2nd International Conference on Communication Software and Networks (ICCSN ’10), 2010, pages 382–386.

[3]. J. Liang, R. Kumar, Y. Xi, and K. W. Ross, “Pollution in P2P File Sharing Systems”, in

Proc. the 24th IEEE International Conference on Computer Communications (INFOCOM), 2005, pages 1174–1185.

[4]. J. Liang, N. Naoumov, and K. W, Ross, "The Index Poisoning Attack in P2P File Sharing Systems", In Proc. the 25th IEEE International Conference on Computer Communications (INFOCOM), 2006, pages 1–12.

[5]. P. Putra and A. Nakao, "An Effective Index Poisoning Algorithm for Controlling Peer-to-Peer Network Applications", In Proc. International Workshop on Modeling, Analysis, and Control of Complex Networks (Cnet '11), 2011, pages 17–22.

[6]. S. Xie, B. Li, G. Y. Keung, and X. Zhang, "Cool streaming: Design, Theory, and Practice", IEEE Trans. on Multimedia, 9(8):1661–1671, May 2007.

[7]. M. Yoshida, S. Ohzahata, A. Nakao, and K. Kawashima, "Controlling File Distribution in Winny Network through Index Poisoning", In Proc. International Conference on Information Networking (ICOIN 2009), 2009, pages 1–5.

[8]. Xiaojun Hei, Chao Liang, Jian Liang, Yong Liu and Keith W. Ross, " A Measurement Study of a Large-Scale P2P IPTV System" , IEEE Communications Magazine (Volume: 49, Issue: 3, March 2011).

[9]. Jiangchuan Liu, Sanjay G. Rao, Bo Li and Hui Zhang, "Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast", Proceedings of the IEEE (Volume: 96, Issue: 1, Jan. 2008).

[10]. Jaeyoung Choi, Jinyoung Han, Eunsang Cho, Ted Kwon and Yanghee Choi, "A Survey on content-oriented networking for efficient content delivery".

[11]. R. Sherwood, B. Bhattacharjee and A. Srinivasan, "A protocol for scalable anonymous communication Security and Privacy", Proceedings 2002 IEEE Symposium.

[12]. Run fang Zhou and Kai Hwang, "Gossip-based Reputation Aggregation for Unstructured Peer-to-Peer Networks Parallel and Distributed Processing", Symposium, 2007, IPDPS 2007 IEEE International.