Kenjiro Cho
Philippe Jacquet (Eds.)

LNCS 3837

Technologies for Advanced Heterogeneous Networks

First Asian Internet Engineering Conference, AINTEC 2005
Bangkok, Thailand, December 2005
Proceedings

Springer

Kenjiro Cho Philippe Jacquet (Eds.)

Technologies for Advanced Heterogeneous Networks

First Asian Internet Engineering Conference, AINTEC 2005
Bangkok, Thailand, December 13-15, 2005
Proceedings

Springer

Volume Editors

Kenjiro Cho
Internet Initiative Japan, Inc.
1-105 Kanda Jinbo-cho, Chiyoda-ku, Tokyo 1010051 Tokyo, Japan
E-mail: kjc@iijlab.net

Philippe Jacquet
INRIA, Campus of Rocquencourt
Domaine de Voluceau, B.P. 105, 78153 Le Chesnay Cedex, France
E-mail: philippe.jacquet@inria.fr

Library of Congress Control Number: 2005936809

CR Subject Classification (1998): C.2.4, C.2, C.3, F.1, F.2.2, K.6

ISSN 0302-9743
ISBN-10 3-540-30884-9 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-30884-3 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11599593 06/3142 5 4 3 2 1 0

Preface

The Asian Internet Engineering Conference (AINTEC) brings together researchers and engineers interested in practical and theoretical problems in Internet technologies. The conference aims at addressing issues pertinent to the Asian region with vast diversities of socio-economic and networking conditions while inviting high-quality and recent research results from the global international research community. The first event was jointly organized by the Internet Education and Research Laboratory of the Asian Institute of Technology (AIT) and the WIDE Project with support from the APAN-TH community.

In response to the recent natural disaster in Asia, AINTEC 2005 solicited papers, among other things, on the survival of the Internet in order to provide alternative means of communication in emergency and chaotic situations. The main topics include: Mobile IP
Mobile Ad Hoc and Emergency Networks
Multimedia or Multi-Services IP-Based Networks
Peer-to-Peer
Measurement and Performance Analysis
Internet over Satellite Communications

There were 52 submissions to the Technical Program, and we selected the 18 papers presented in these proceedings. In addition, we have three invited papers and one invited position paper by leading experts in the field.

Finally, we would like to acknowledge the conference General Chair, Kanchana Kanchanasut of AIT, and the Local Organizers team from AIT, namely, Pensri Arunwatanamongkol, Withmone Tin Latt and Yasuo Tsuchimoto, for organizing and arranging this conference. We are also grateful to the French Ministry of Foreign Affairs through its French Regional Cooperation and the ICT Asia project (STIC-ASIE) for providing travel support.

December 2005 Kenjiro Cho and Philippe Jacquet

Organization

General Chair

Kanchana Kanchanasut (AIT, Thailand)

Program Committee Co-chairs

Kenjiro Cho, WIDE Project
Philippe Jacquet, INRIA, France

Program Committee

Kazi Ahmed (AIT, Thailand)
Patcharee Basu (SOI/ASIA, Japan)
Randy Bush (IIJ, USA)
Thomas Clausen (Polytechnique, France)
Noel Crespi (INT, France)
Tapio Erke (AIT, Thailand)
Thierry Ernst (Keio University, Japan)
Chalermek Intanagowiwat (Chulalongkorn U, Thailand)
Alain Jean-Marie (LIRMM/INRIA, France)
T.J. Kniveton (NOKIA Research Center, USA)
Youngseok Lee (CNU, Korea)
Bill Manning (USC/ISI, USA)
Thomas Noel (University Louis Pasteur, France)
Alexandru Petrescu (Motorola)
Anan Phonphoem (Kasetsart U, Thailand)
Poompat Saengudomlert (AIT, Thailand)
Teerapat Sa-nguankotchakorn (AIT, Thailand)
Kazunori Sugiura (WIDE Project, Japan)
Jun Takei (JCSAT, Japan)
C.W. Tan (USM, Malaysia)
Takamichi Tateoka (University of Electro-Communications, Japan)
Antti Tuominen (Helsinki University of Technology, Finland)
Ryuji Wakikawa (WIDE Project, Japan)

Local Organization

Pensri Arunwatanamongkol (AIT, Thailand)
Withmone Tin Latt (AIT, Thailand)
Yasuo Tsuchimoto (AIT, Thailand)

Table of Contents

Invited Papers

Wireless, Mobility and Emergency Network

Routing in Ad-Hoc Network

Extending MANET

Invited Position Paper

Securing Network

Multi-services in IP-Based Networks

Measurement and Performance Analysis

Efficient Blacklisting and Pollution-Level Estimation in P2P File-Sharing Systems

Jian Liang, Naoum Naoumov, and Keith W. Ross

Department of Computer and Information Science,
Polytechnic University, Brooklyn NY 11201, USA
{jliang, naoum, ross}@poly.edu

Abstract. P2P file-sharing systems are susceptible to pollution attacks, whereby corrupted copies of content are aggressively introduced into the system. Recent research indicates that pollution is extensive in several file sharing systems. In this paper we propose an efficient measurement methodology for identifying the sources of pollution and estimating the levels of polluted content. The methodology can be used to efficiently blacklist polluters, evaluate the success of a pollution campaign, to reduce wasted bandwidth due to the transmission of polluted content, and to remove the noise from content measurement data. The proposed methodology is efficient in that it does not involve the downloading and analysis of binary content, which would be expensive in bandwidth and in computation/human resources. The methodology is based on harvesting metadata from the file sharing system and then processing off-line the harvested meta-data. We apply the technique to the FastTrack/Kazaa file-sharing network. Analyzing the false positives and false negatives, we conclude that the methodology is efficient and accurate.

1 Introduction

By many measures, P2P file sharing is the most important application in the Internet today. There are more than 8 million concurrent users that are connected to either FastTrack/Kazaa, eDonkey and eMule. These users share terabytes of content. In the days of Napster (circa 2000), most of the shared files were MP3 files. Today the content includes MP3 songs, entire albums, television shows, entire movies, documents, images, software, and games. P2P traffic accounts for more than 60% of tier-1 ISP traffic in the USA and more than 80% of tier-1 traffic in Asia [1].

Because of the their decentralized and non-authenticated nature, P2P file sharing systems are highly susceptible to "pollution attacks". In a pollution attack, a polluter first tampers with targeted content, rendering the content unusable. It then deposits the tampered content, or only the metadata for that content, in large volumes in the P2P file sharing system. Unable to distinguish polluted files from unpolluted files, unsuspecting users download the files into their own file-sharing folders, from which other users may then later download the polluted files. In this manner, the polluted copies of a title spread through the

K. Cho and P. Jacquet (Eds.): AINTEC 2005, LNCS 3837, pp. 1–21, 2005.

file-sharing system, and the number copies of the polluted title may eventually exceed the number of clean copies. The goal of the polluter is to trick users into repeatedly downloading polluted copies of the targeted title; users may then become frustrated and abandon trying to obtain the title from the file-sharing system. As a side effect, however, the polluted content becomes a persistent "noise" in the P2P system that interferes with research measurement work. Pollution is currently highly prevalent in file-sharing systems, with as many as 50% to 80% of the copies of popular titles being polluted [2].

In this paper we study mechanisms to measure the effectiveness of a pollution attack. We emphasize, however, that we do not take a side in the P2P file-sharing debate, neither condoning nor condemning the pollution attacks that are commissioned by the music, television and film industries. But given that P2P file sharing traffic is currently the dominant traffic type in the Internet, and that the files being transferred are frequently polluted, a significant fraction of Internet bandwidth is clearly being wasted by transporting large, corrupted files. It is therefore important to gain a deep understanding of the pollution attack and develop effective mechanisms to measure it.

In this paper we explore two techniques for countering the pollution attack:

- **Identifying pollution source IP ranges:** The goal is to identify IP address ranges that are broad and complete enough to cover the hosts providing polluted content, yet narrow enough to exclude the vast majority of ordinary users.
- **Identifying the pollution level of titles:** With knowledge of which titles are being polluted and to what extent users and researchers can use the file sharing system accordingly.

In developing methodologies we have not only aimed for accuracy but also for efficiency. For source identification, one approach would be to download copies of titles from a vast number of IP addresses and then manually check the copies for pollution; the IP addresses that consistently supply polluted content could then be marked. Such an approach would be highly inefficient, requiring enormous bandwidth, computing and human resources, and would also introduce significant "probing" traffic into the Internet.

Our methodologies do not involve the downloading of any files. Instead, they identify polluting IP address ranges and targeted titles by collecting and analyzing metadata from the file sharing system. The metadata is harvested by crawling the nodes in the P2P system and sending tailored queries to each of the crawled nodes. The harvested metadata can then be analyzed to obtain detailed information about the numbers of versions and copies, and the IP subnets containing the versions and copies, for a large number of investigated titles. From this detailed information, our methodology constructs the blacklisted IP ranges and the estimated pollution levels for the targeted titles. The methodology is efficient in that it collects metadata (text) rather than content (which is typically 3MB to several GB per file for music and video) and that a large number of titles and virtually all the file-sharing nodes can be investigated in one crawl.

Our contribution is as follows:

- We developed a methodology for creating a blacklist set. The methodology is based on identifying high-density prefixes, which are prefixes in which the nodes that have a copy of a particular title have, on average, a large number of copies. We provide a heuristic for separating the low density prefixes from the high density prefixes, and a mechanism to merge prefixes that are topologically close. The set of resulting merged prefixes constitutes the pollution source set. We then developed several metrics for measuring the accuracy of the set. The two principal metrics are probability of false positive and false negative. We also examine secondary metrics, including comparing the download times and last-hop RTTs at nodes we have identified and regular nodes.
- We developed a methodology for estimating the pollution level of a title, which is defined as the ratio of polluted copies in the network to the total number of copies in the network. This estimate does not involve the downloading of any files and is solely based on the harvested metadata. We then evaluate our estimate by measuring the actual pollution levels of selected titles.
- We crawled FastTrack for 170 titles, including songs and movies. We then applied the methodologies to the FastTrack metadata harvested during the crawling procedure. Our resulting pollution source set contains 112 prefixes. Our evaluation metrics indicate that the set is accurate, with low probabilities of false positives and false negatives. We also find that the estimates for pollution levels in examined titles is accurate.

This paper is organized as follows. In Section 2 we describe the pollution attack in detail and introduce important terminology. In Section 3 we describe in detail the metholodogies and the evaluation procedures for creating the pollution source set and the pollution-level estimates. In Section 4 we describe the experimental setup, including the crawler and PlanetLab experiments. Section 5 provides the results of our experiment, including evaluation results for the methodologies. Section 6 describes previous work related to this paper. We conclude in Section 7.

2 Overview of Pollution

2.1 File Sharing Terminology

We first provide an overview of a generic P2P file-sharing application. This will allow us to introduce some important terminology that is used throughout the paper. In this paper we are primarily concerned with the sharing of music and video. We shall refer to a specific song or video as a **title**. A given title can have many different **versions** (in fact, tens of thousands). These versions primarily result from a large number of rippers/compressors, each of which can produce slightly different files when created by different users. Modifications of metadata can also create different versions. Users download different versions of titles from

each other, thereby creating multiple **copies** of identical file versions in the P2P file sharing system. At any given time, a P2P file-sharing system may make available thousands of copies of the same version of a particular title.

A file in a P2P file sharing system typically has **metadata** associated with it. There are two types of metadata: metadata that is actually included in in file itself and is often created during the ripping process (e.g. ID3 Tags in mp3 files); and metadata that is stored in the file-sharing system but not within the shared files themselves. This "outside-file" metadata may initially be derived from the "inside-file" metadata, but is often modified by the users of the file-sharing systems. It is the outside-file metadata that is employed during P2P searches. In this paper, when using the term metadata, we are referring to the outside-file metadata. Because different copies of a version of a title may be stored on different user nodes, the different copies can actually have different metadata.

When a user wants to obtain a copy of a specific title, the user performs a keyword search, using keywords that relate to the title (for example, artist name and song title). The keywords are sent within a query into the file-sharing network. The query will visit one or more nodes in the file sharing network, and these nodes will respond if they know of files with metadata that match the keywords. The response will include the metadata for the file, the IP address of the node that is sharing the file, and the username at that node. For many file sharing systems, the response will also include a **hash**, which is taken over the entire version. To download a copy of a version, one sends a request message (often within an HTTP request message) to the sharing user. In this request message, the version is identified by its hash. Many file sharing systems employ parallel downloading, in which case requests for different portions of the version are sent to different users sharing that file.

Many nodes in P2P file sharing systems are behind Network Address Translators (NATs). When crawling a NATed node's private IP address and private port number may be provided rather than its public IP address and public port number. Since the range of private IP address is relatively narrow, different NATed users may have the same private IP address. Thus, from the crawling data, we cannot distinguish between different users solely by their IP addresses. In order to distinguish between different users, including NATed users, we define a **user** as the triple (IP address, port number, username).

2.2 Intentional Pollution

Naturally pollution occurs in P2P systems when users share corrupted versions of some titles. However, the amount of such pollution is negligible. Other users of the system, however, may intentionally introduce a large number of corrupted files. They create numerous versions of their targeted title by tampering with it in one or more ways with the binary content of the file. Then, they connect one or more nodes to the P2P file-sharing system and places the tampered versions into its shared folders on these nodes. Users query for the title and learn about the locations of versions of the title, including the polluted versions and down-

load one or more polluted versions. The P2P software then automatically places the file in the shared folders of those users and the pollutions spreads further. That kind of pollution is prevalent in modern P2P file sharing systems such as FastTrack/Kazaa [2]

Most of the pollution today emanates from "professional" polluters that work on the behalf of copyright owners, including the record labels and the motion-picture companies. From this economic context and from our own testing and usage experience, we conclude that the professional polluters tend to pollute popular content, such as recently-released hit-songs and films. In [2] a random sample of recent, popular songs were shown to be heavily polluted whereas a random sample of songs for the 70s were shown to be mostly clean.

In order to facilitate the spread of the polluted content polluters have high-bandwidth Internet connections ,have high availability, and they are not behind firewalls or NATed routers.

In our methodology for detecting polluted content and blacklisting polluters, we will make the following assumptions about polluters. Many of these assumptions will be corroborated in Section 5, where our measurement results are presented.

- Because polluters share popular titles at attractive file-transfer rates, there is a high demand for their content from unsuspecting users. To meet the demand, the polluter often uses a server farm at one or more polluter sites. The nodes in a server farm are concentrated in a narrow IP address range.
- Whereas regular P2P users run one client instance per host, polluters often run many clients in each of their nodes, with each instance having a different username and sharing its own set of copies and versions for the targeted titles. This is done to improve placement of search results in the users' GUIs.
- A polluter distributes multiple polluted versions of the same title. This also improves the placement of search result in the users' GUIs. As we will show in Section 5, an ordinary user typically has a small number of versions of any title. To compete with all the clean versions in the display of the search results, a polluter needs to provide many different versions (each with a different hash) to increase the chances that its versions are selected from the users' GUIs.

3 Methodology

In this paper we develop methodologies for two tasks. The first task, which we refer to as **blacklisting**, is to find the IP address ranges that include the large majority of the polluters. The second task, referred to as **pollution level estimation**, is to determine the extent of pollution for specified titles. For both of these tasks, the first step is to crawl the file sharing system, as we now discuss.

3.1 Crawling

Crawling a P2P file sharing system is the process of visiting a large number of nodes to gather information about the copies of files being shared in the

system. The crawler might gather, for example, the IP addresses and hashes of all copies of files being shared in the network for a set of specific titles over a given period of time. Several independent research groups have developed crawlers for P2P file sharing systems. A crawler for the original single-tier Gnutella system is described in [3]. A crawler for the current two-tier Gnutella system (with "ultrapeers") is described in [4]. A crawler for eDonkey is described in [15]. A crawler for the FastTrack P2P file sharing is described in [2]. Since P2P networks are dynamic, with nodes frequently joining and leaving, a good crawler needs to rapidly crawl the entire network to obtain an accurate snapshot.

The first step in our methodologies is to crawl the P2P file sharing system and obtain the following information for each title of interest: the number of versions in the file sharing system for the title; the hash values for each of the versions; the number of copies of each version available in the file sharing system; for each copy, the IP address of the node that is sharing it; the port number of the application instance at that node (many modern P2P systems vary the port number across nodes to bypass firewalls); the username at that node; and, for each copy, some copy details (e.g., playtime, file size, description, etc). For each title of interest, the crawler deposits this information in a **crawling database**, which can then be analyzed off-line. We will describe a crawler for the FastTrack network in Section 4.

3.2 Identifying Pollution Sources

Polluters typically control blocks of IP addresses and can easily move their nodes from one IP address to another within the block. Thus, rather than idnetifying individual IP addresses, we should find ranges or IP addresses that are likely to include the polluters in the near future as well as the present. Our methodology has the following steps:

1. Crawl the P2P file sharing system as described above.
2. From the data in the crawling database, identify the /24 prefixes that are likely operated by polluters.
3. Merge groups of /24 prefixes that are topologically close and don't cross BGP prefixes. The set of merged prefixes becomes our blacklist set.

We now describe the second and third steps in more detail.

The second step is to identify /24 prefixes that are likely operated by polluters. A polluter typically leases from a data center a set of server nodes in a narrow IP address range. Data centers do not normally include ordinary P2P users, which typically access the Internet from residences and universities. A /24 prefix is small enough so that both polluters and ordinary users do not operate from within the same prefix; and it is large enough to cover multiple polluting servers in most subnets. In the third step, we search for larger subnets.

Let N denote the number of titles investigated and T_n denote the nth title. For each title T_n, we determine from the crawling database the /24 prefixes that contain at least one copy of title T_n. Suppose there are $I^{(n)}$ such /24 prefixes; denote the set of these prefixes by $\mathcal{P}^{(n)} = \{p_1^{(n)}, p_2^{(n)}, \ldots, p_{I^{(n)}}^{(n)}\}$.

We now introduce the important concept of the "density of a prefix," which will be used repeatedly in this paper. For each such prefix $p_i^{(n)}$, define $x_i^{(n)}$ to be the number of IP addresses in the prefix with at least one copy of the title and $y_i^{(n)}$ to be the number of copies (included repeated copies across nodes) of the title stored in the prefix. Finally, define the **density** of prefix $p_i^{(n)}$ as $d_i^{(n)} = y_i^{(n)}/x_i^{(n)}$.

From our assumptions about how polluters operate (see Section 2), we expect the prefixes with high density values to be operated by polluters and prefixes with low densities to contain only "innocent" users. We consider prefixes with a density higher than a threshold $d_{thresh}^{(n)}$ to be operated by polluters. There are many possible heuristics that can be used to determine this threshold. We now describe a simple heuristic that gives good performance. It is based on the median value of the distinct density values in $\{d_1^{(n)}, d_2^{(n)}, \ldots, d_{I^{(n)}}\}$ denoted by $d_{median}^{(n)}$. Of course, different prefixes have different numbers of users and different densities, so in order to allow for a variance in user behavior we set a threshold to a multiple of the median. Specifically, our heuristic sets the threshold to

$$d_{thresh}^{(n)} = k d_{median}^{(n)} \tag{1}$$

where k is an appropriately chosen scaling factor (see Section 5). We say that a prefix $p_i^{(n)}$ is a **polluting prefix** if $d_i^{(n)} \geq d_{thresh}^{(n)}$. Let $\mathcal{Q}$ be the union of all the polluting /24 prefixes over all N titles.

A polluter may actually operate within a network that is larger than a /24 prefix. The third step of our methodology is to create larger prefixes which encompass neighboring /24 prefixes in $\mathcal{Q}$. For this, we merge adjacent prefixes in the IP space. We also merge some non-adjacent prefixes. To this end, we perform a traceroute from each of 20 PlanetLab nodes to one IP address in each of the prefixes in $\mathcal{Q}$. Prefixes which have the same last router become candidates for merging. In doing this we need to account for the possibility that some of the traceroutes passing through the same last router may actually pass through the router via different interfaces (and thus IP addresses) [9]. Suppose there are J groups of prefixes, with each prefix in a group sharing the same last router. (Some groups may contain a single prefix.) Let $\mathcal{G}_j$, $j = 1, \ldots, J$, denote the groups. For each group of prefixes $\mathcal{G}_j$, denote p_j as the longest prefix that covers all the prefixes in $\mathcal{G}_j$. For each such p_j we verify that it does not cross prefixes found in a BGP table. If it does, we decompose p_j back into its original /24 prefixes. Let $\mathcal{P}$ be the resulting set of prefixes. $\mathcal{P}$ is our final pollutions source set, and consists of all the p_j's that pass the BGP test and all of the decomposed /24 prefixes as just described.

Note that this methodology for creating a pollution source set does not involve the downloading of content. Indeed, any download-based methodology would require the downloading of an excessively large number of files as well as an automated procedure to determine whether a downloaded file is polluted. Our approach is instead based on the metadata that is gathered by the crawler. This approach is efficient in that crawling a large-scale P2P file-sharing system can be quickly done with modest resources.

3.3 Evaluation Procedure for Pollution Source Sets

The pollution source set $\mathcal{P}$ may not be completely accurate in that it may not contain all polluting nodes (false negatives) and it may contain some active nodes that are innocent users (false positives). We evaluate a blacklisting methodology by estimating the probability of false positives and false negatives.

To this end, we need a procedure to determine whether a downloaded version of any given title is polluted. This can be done by downloading the version and manually watching or listening to it. Such a manual procedure would require an excessive amount of human resources. Instead we use a simple automated procedure which has been shown to give accurate results [2]. Specifically, we download the version into RAM and declare the version to be clean (unpolluted) if the following three criteria are met:

1. Rehashing the file results in the same hash value as the one that was used to request the file;
2. The title is decodable according to the media format specifications; for example, an mp3 file fully decodes as a valid mp3 file [24].
3. The title's playback duration is within 10% of the one specified in release information for that title.

In any one of the three criteria is violated, we consider the version to be polluted. We refer to this procedure as the **automated version-checking procedure**.

Having described our procedure to determine whether a downloaded version is polluted, we can now state our false-negative and false-positive evaluation procedure. To evaluate the false-negative probability, we randomly select 1,000 users having IPs outside of $\mathcal{P}$ and having a copy of at least one of the investigated titles. For each randomly selected node, we randomly download 5 versions stored at that node. We declare a node to be a **false negative** if all of the following conditions are satisfied: (*i*) it has at least 5 versions of any one of the investigated titles; (*ii*) its upload throughput is greater than a given threshold. (In Section 4 we describe how we estimate a node's upload throughput); (*iii*) its Last Hop RTT is less than a threshold (we define Last Hop RTT in Section 4) ; and (*iv*) all of the randomly selected versions are polluted. Thus a randomly selected node is declared a false negative if that node has the main characteristics of a polluting node. (See Section 2.) The false-negative probability is simply the the number of randomly selected nodes declared to be false negatives divided by the total number of randomly selected nodes.

A false positive occurs when an "innocent" non-polluting node is blacklisted as a polluter by our methodology. This can happen when a /24 prefix is labelled a polluting prefix but contains non-polluting users, or when innocent users are added to $\mathcal{P}$ during the merging process. To evaluate the false-positive probability, we randomly select 1,000 nodes in $\mathcal{P}$ containing a copy of at least one of the titles. For each randomly selected node, we randomly download five versions stored at that node. We declare a randomly selected node to be a **false positive** if any of the following criteria are satisfied: (*i*) its throughput is smaller than the threshold; (*ii*) its last hop RTT is larger than the threshold; and (*iii*) at least

one of the randomly selected versions is clean. The false-positive probability is simply the the number of randomly selected nodes declared to be false positives divided by the total number of randomly selected nodes.

3.4 Estimating Content Pollution Levels

In this subsection we provide our methodology for estimating the pollution level of any arbitrary title T_n. The methodology builds on the blacklisting methodology. We define the **pollution level** of a title as the fraction of copies of the title available in the P2P file sharing system that are polluted. The pollution level of a title can be estimated by randomly selecting a large number of copies of the title, downloading each of the copies, and then testing the copies for pollution (either by listening to them or through some automated procedure). This requires an exorbitant amount of resources, particularly if we wish to accurately determine the pollution levels of many titles. We instead estimate the pollution levels of titles directly from the metadata available in the crawling database. To this end, we make the following assumptions:

1. All copies of the title that are stored in a blacklisted node (that is, in a node in $\mathcal{P}$) are polluted.
2. For each node outside of $\mathcal{P}$ with at least one copy of T_n, all copies stored at that node are polluted except for one copy.

With these assumptions, we now derive an expression for $E^{(n)}$, the **estimated pollution level** of title T_n. For a title Recall that $y^{(n)}$ is the total number of copies of the title available in the crawling database. Also let $z^{(n)}$ be the number of nodes outside of the blacklist set $\mathcal{P}$ that have at least one copy of T_n. The above two assumptions imply that the number of copies of T_n that are polluted is $y^{(n)} - z^{(n)}$; thus, our estimate of the pollution level for title T_n is

$$E^{(n)} = \frac{y^{(n)} - z^{(n)}}{y^{(n)}}. \tag{2}$$

3.5 Evaluation Procedure for Pollution-Level Estimation

$E^{(n)}$ is an estimate of the pollution level of title T_n, derived solely from the metadata in the crawling database. To evaluate the accuracy of this estimate, we compare it with a measured value, which is obtained by actually downloading content. Specifically, for a given title T_n we do the following:

1. We download the most popular versions of the title. The number of versions downloaded is such that the downloaded versions covers at least 80% of all copies of the title in the file-sharing system. For title T_n, let J_n be the number of versions that meet this 80% criterion.
2. For each of these versions, we determine if the version is polluted or not using the automated version checking procedure described in Section 3.3. Let $\delta_i^{(n)}$ be equal to 1 if version i is determined polluted and be equal to 0 otherwise.