applied sciences

MDPI

Article

Teletraffic Analysis of DoS and Malware Cyber Attacks on P2P Networks under Exponential Assumptions

Natalia Sánchez-Patiño [1], Gina Gallegos-Garcia [2,*] and Mario E. Rivero-Angeles [2]

[1] Facultad de Informática de Barcelona, Universidad Politécnica de Cataluña, Carrer de Jordi Girona, 31, 08034 Barcelona, Spain; natalia.de.los.angeles.sanchez@estudiantat.upc.edu
[2] Centro de Investigación en Computación, Instituto Politécnico Nacional, Av. Juan de Dios Bátiz S/N, Nueva Industrial Vallejo, Gustavo A. Madero, Ciudad de México 07738, Mexico; mriveroa@ipn.mx
* Correspondence: ggallegosg@ipn.mx or ggallegos@cic.ipn.mx; Tel.: +52-55-57296000 (ext. 56509)

Abstract: Peer-to-peer (P2P) networks are distributed systems with a communication model in which no central authority governs the behavior of individual peers. These networks currently account for a considerable percentage of all bandwidth worldwide. However, this communication model also has a clear disadvantage: it has a multitude of vulnerabilities and security threats. The nature of the P2P philosophy itself means that there is no centralized server responsible for uploading, storing, and verifying the authenticity of the shared files and packets. A direct consequence of this is that P2P networks are a good choice for hackers for the spread of malicious software or malware in general since there is no mechanism to control what content is shared. In this paper, we present a mathematical model for P2P networks to study the effect of two different attacks on these systems, namely, malware and denial of service. To analyze the behavior of the cyber attacks and identify important weaknesses, we develop different Markov chains that reflect the main dynamics of the system and the attacks. Specifically, our model considers the case in which a certain number of nodes are infected with a cyber worm that is spread throughout the network as the file is shared among peers. This allows observation of the final number of infected peers when an initial number (we evaluate the system for from 1 to 14 initial nodes) of malicious nodes infect the system. For the DoS attack, our model considers the portion of peers that are unable to communicate and the average attack duration to study the performance degradation of such an attack. A two-pronged approach was used to study the impact of the attacks on P2P networks; the first focused only on the P2P network, and the second focused on the attacks and the network.

Keywords: teletraffic; peer-to-peer networks; cyber attacks

check for updates

Citation: Sánchez-Patiño, N.; Gallegos-Garcia, G.; Rivero-Angeles, M.E. Teletraffic Analysis of DoS and Malware Cyber Attacks on P2P Networks under Exponential Assumptions. *Appl. Sci.* **2023**, *13*, 4625. https://doi.org/10.3390/app13074625

Academic Editors: Ireneusz Kubiak, Tadeusz Wieckowski and Yevhen Yashchyshyn

Received: 2 March 2023
Revised: 25 March 2023
Accepted: 28 March 2023
Published: 6 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Peer-to-peer networks are computer networks in which all or some aspects function without fixed clients or servers [1]. They have elementary operating principles; their maintenance cost is low because they do not have centralized servers, and, therefore, thy are very popular, especially in data transmission and file-sharing services of any kind. Moreover, they avoid bottlenecks at the server since high traffic loads imply that many users are active in the system, and each of these contributes resources (memory, processing, and communication bandwidth) to the overall operation of the network. They are organized in a flat structure that allows a direct exchange of information between clients. That is why they are ideal for file dissemination to many computers. Figure 1 shows the operation of this kind of network.

Indeed, a *peer* acts as a server and as a client in a self-organized and dynamically coordinating structure, since the number of nodes can be increased or decreased at any time. Peers share the file with each other, which is divided into chunks that are distributed throughout the system. Then, a given peer uploads a set of chunks to other peers that

do not have them, while it downloads missing chunks from other peers. The respective peer is responsible for finding the correct information. If the data are common, it is easy to find them, but it is difficult to find rare data from other peers. Peers with the complete file usually are called *Seeds*, while peers without the complete file are called *Leechers* [2]. Hence, seeds can share the complete file with any given leecher. In this regard, it is important for seeds to remain in the system after they have downloaded the file to increase the capacity of the system.

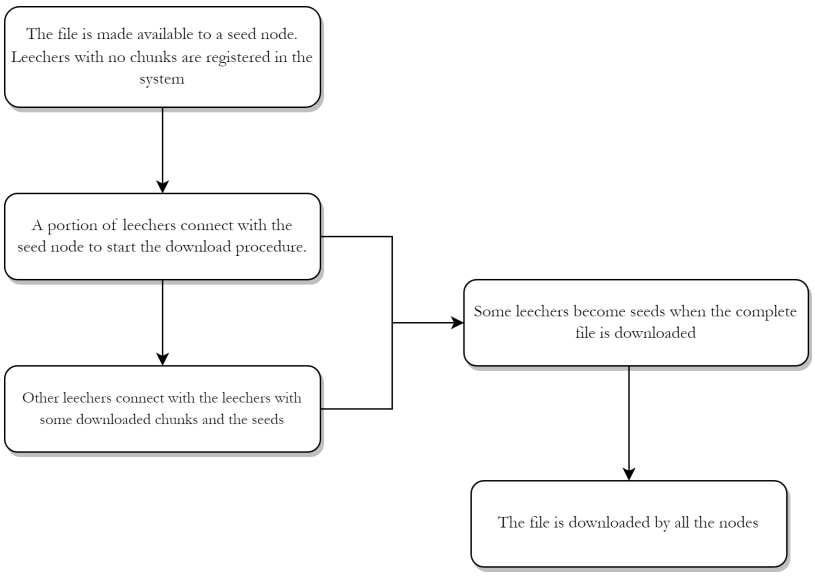


Figure 1. Flow chart for operation of the P2P network.

P2P networks can be divided into three categories: unstructured P2P, structured P2P, and hybrid P2P networks [3–5]. An unstructured P2P network does not tell the exact location of the data, so it looks for the requested data in its network. This type of network represents a degree distribution of power law and is dominant in real-life deployment. The disadvantage of this type of network is that it does not seem to know if it can access the data. A structured P2P network uses a globally consistent protocol for efficient searching. It uses a hash table called a distributed hash table (DHT). This table provides information about where a particular file comes from based on several important factors. A hybrid P2P network combines the topology of a structured and an unstructured one. This type of network defines so-called *Superpeers*, which act as servers to a small portion of the network, and each of them has a list of file information. From this description, it is clear that, for any P2P network, nodes share the chunks directly among themselves without the need for previous verification from a centralized authority. Hence, it would be relatively easy for a malicious node to modify a given chunk to propagate malware throughout the network. Furthermore, since the P2P network entirely relies on users sharing their communication resources, a simple denial of service attack (DoS), which impedes neighbor nodes' communication, can drastically hinder the operation and performance of the system [6].

Taking BitTorrent as an example, when a file is downloaded, part of the file is downloaded, and the other part is received at the same time by different peers who may have downloaded a chunk or the whole file [7]. This download mechanism in peer-to-peer networks improves the download speed, avoiding bottlenecks at the server and taking advantage of the user's resources to increase the system bandwidth. However, the adequate operation of the system heavily relies on nodes trusting each other. Hence, in the case of malware inserted in one or several chunks, it can be spread and affect many users. To the best of our knowledge, the malware propagation dynamics in a P2P network have been largely overlooked. Based on the above, in this paper, we develop a mathematical model based on Markov chains that captures the main dynamics of the system, including peer

arrival and exit, file downloading, and infected chunks being disseminated. In this sense, the main contributions of our work are as follows:

- We propose an analytical framework to study the behavior of the malware attack that allows identification of the rate at which the malware is dispersed and the number of infected nodes.
- We develop a mathematical analysis to study the impact of jamming the communications of a group of nodes under a DoS attack.
- Based on the insight gained from the mathematical description of the attacks, we recommend the use of countermeasures to investigate the dynamics of the attack when cybersecurity measures are enabled.
- We obtain the performance metrics of the system under the different attacks and different conditions of the attacks.

The proposed teletraffic model is an approximation analysis for different cyber attacks on P2P systems when events occur in exponentially distributed times. Expressly, for the DoS type of attack, we assume that the attacker impairs or limits the communication capabilities of the network during exponentially distributed random times. If the attack has a random duration with distributions with a coefficient of variation (CoV) different from one (as in the case of the exponential distribution), this model is no longer accurate. However, it can be easily extended by using phase-type distributions to account for a CoV < 1 using an Erlang distribution or for a CoV > 1 using a hyper-exponential distribution, in both cases using the developed Markov chains in this work as the basic models. For the malware attack, we consider that there is an initial number of malicious peers, N_I, that is registered in the system and that will begin the worm dissemination when the file distribution begins. These malicious nodes have the same characteristics in terms of their communication and processing capabilities. As such, the upload and download bandwidths are not different from the rest of the nodes. Whenever the system operation begins, the malicious nodes will spread the malware to any other peer in the system, and these infected peers will continue the malware propagation to other peers that connect to them, and so on. As such, the only variable of the attack is N_I. Note that the malicious nodes cannot behave differently than regular nodes in terms of the upload and download rates because the P2P system would not be compatible, and they would not be able to communicate. If countermeasures are considered, we assume that these measures will prevent some nodes from being infected, while others would not be able to avoid infection (with the probability P_I in our model) since there are no completely secure cyber systems. In the case of the DoS attack, there is no malware attack, but rather some peers are unable to communicate. Hence, the model considers the portion of nodes that cannot download nor upload any chunks, δ, and the average duration of the attack, $1/\triangle$. We believe that these are the only parameters of the attack from the standpoint of the effect on the P2P communication capabilities.

In addition, in this work, we are not concerned with the computational effects of cyber attacks, i.e., the effect on electronics or computer-related processes. Indeed, our model and analysis cannot reflect these issues, as pointed out by the reviewer. Our model can only reflect the impact on the communication capabilities under the malware and DoS attacks. However, our analysis can consider different characteristics of these attacks, such as the number of initially infected nodes, the mean duration of the attacks, the efficiency of the countermeasures, and the number of nodes that cannot use their communication capabilities. In this regard, our model cannot measure the impact of other variables related to these attacks, such as the time of infection (nodes are not infected instantly by the malware or DoS attack), the use of firewalls, the capabilities of the computer used by the attacker, and the capabilities of the nodes under attack, among many other parameters that can have an important impact on the development of the attack. However, as is often the case in analytical studies, there is a compromise between the tractability of the analysis and the details of the model. If we were to introduce these parameters, the model would be much more complex and probably could not be studied using these mathematical tools.

The advantage of using an approximation based on this simplified model is that it can obtain general results before the implementation of the system.

The rest of the paper is organized as follows. We give the background for this paper in Section 2. We survey related work in Section 3. Furthermore, we present the main assumptions and considerations of the problem in Section 4. Then, in Section 5, we introduce the mathematical model approach using Markov chains to represent the interactions of the attacks in the topology of a P2P network. In Section 7, we present the results of our three different scenarios, analyzing the impact of crucial variables on the model. Finally, Section 8 concludes this paper and points out future research directions.

2. Background

Mathematical models developed to study the propagation of infectious diseases have been adapted to the case of worm propagation [8]. Worms are programs that spread themselves across a network by exploiting security or policy flaws in commonly used services [9]. In this regard, several studies of worm propagation have been conducted in which the main mathematical tool used is structural equation modeling (SEM), which is derived from epidemic models in which the host can only transfer from susceptible to infectious. The susceptible, infected, and recovered model improves on the SEM model by taking the removed state and the transition from infectious to removed into account [10]. The SIS model is another evolution of the SEM model; it assumes that infectious hosts can get back to a susceptible state with a certain probability [11]. Unlike these works, we assume that nodes are infected by the distribution of the chunks shared among peers, which is directly related to resource sharing and communication procedures. Moreover, we model a jamming attack based on the same dynamics of the P2P network. In the network security context, both deterministic and stochastic transmission models of worms, based on their respective equivalents in epidemiology, have been proposed. Deterministic propagation models of worms may be further classified into two categories: continuous-time and discrete-time. Since the propagation of worms is a discrete event process, discrete-time propagation models of worms are more accurate than their continuous-time counterparts in the deterministic regime. Some relevant works on deterministic propagation models of worms can be found in [8,11–13]. All of the previous models consider a continuous-time propagation base except the last one, which considers discrete-time.

Stochastic propagation models of active worms are based on the notion of stochastic processes. All of them are discrete-time in nature. Two prominent instances of stochastic propagation models of worms can be found in [12,14–16].

Attacks on P2P networks can be classified into two types, namely, general and specific attacks [17]. From the general network attacks perspective, this classification provides the most damaging attacks that threaten the network since they aim to disable the complete operation of the system. Malware, DoS, and DDoS (denial of service) fall under this category [18–22].

From the specific attacks on P2P networks perspective, they can also be classified into network and application levels. At the network level, an adversary may try to break the routing system, block access to information by impeding the routing process, or obtain some particular identifiers. At the application level, an adversary can attempt to corrupt or delete data stored in the system. Some papers on network attack simulations can be found in [18,23–25].

3. Related Work

Different models have been proposed from different perspectives. We mention the most relevant and use Table 1 to summarizes introduced methodology. In [26], a model is presented that predicts how a P2P-based virus propagates through a network. Ref. [27] presents a density-dependent Markov jump process model for worms. In [28], a model based on an event-driven simulator is developed. Works focused on propagation can be

found in [29–35]. Ref. [36] suggests a model for proactive worm prevention based on P2P networking technologies. In [37], a unified model is developed. Ref. [38] presents a model based on complex network theory. Ref. [39] submits a customized form of susceptible–exposed–infected. A new multi-node coordinated attack model is proposed in [8].

In the context of the Internet of Things, different approaches have been proposed [40–43]. However, many of these works are focused on IoT systems, which have many key differences from P2P systems. Specifically, in P2P networks, it is desirable for all nodes to cooperate and communicate with the rest of the nodes to increase the system bandwidth. In contrast, in IoT systems, nodes usually have no clear advantage in communicating with any other node in the system. Rather, they communicate with the sink node or some specific nodes that may be used as relay nodes to reach the sink. In this regard, a malware attack would have an entirely unique behavior and performance as in the P2P network. Furthermore, for the DoS attack, if certain key nodes in an IoT system are impaired to communicate, the effect on the system performance would be greatly affected if the packet cannot reach the sink. This is especially true in P2P networks. Indeed, only a portion of nodes would not be able to download nor upload chunks, but the rest of the nodes would continue to operate normally, but with a decreased bandwidth.

Table 1. Related work and introduced methodology.

Reference	Introduced Methodology
[26]	This model is a modified version of the susceptible–exposed–infected model from the field of epidemiology.
[27]	Computationally simple hybrid deterministic/stochastic model for the observed scanning behavior on a local network.
[28]	Event-driven simulator.
[29]	Discrete simulations that provide some verification.
[30]	Non-linear differential equations.
[31]	Deep analysis of the features of file sharing and virus propagation.
[32]	0.01 Files in the network as subjects instead of the computers, as is traditional.
[33]	Epidemiological modeling that predicts.
[34]	In-depth analysis of the active malicious code characteristics.
[35]	Based on the worm propagation characteristics and mechanism of a conditional triggered worm attack.
[37]	Attacks of various pollution, including file-targeted attacks and index-targeted attacks.
[38]	Complex network theory.
[39]	A customized form of susceptible–exposed–infected model based on the study of epidemiology.
[8]	Deterministic models of the propagation of computer viruses in a heterogeneous network.

As we have mentioned, our model focuses on malware and DoS attacks on the P2P system since these are general attacks focused on disrupting the normal operation of the complete system and, therefore, are the most damaging attacks. In this regard, we provide a mathematical framework to quantitatively measure the effectiveness of such attacks. We model both the infection procedure and the case in which countermeasures are used to limit the propagation of malware in infected chunks. Furthermore, we consider a jamming

attack in which a malicious node transmits an interference signal that is capable of disabling certain nodes, thereby preventing them from sharing their chunks with other peers.

4. Malware and DoS Attacks in the P2P Network

In this section, we describe in detail the attacks that we mathematically model using Markov chains. It is important to note that our work theoretically models both the DoS and worm-based attacks similarly to how the Erlang B formula and the basic P2P system (also derived from continuous-time Markov chains) model the blocking probability in telephone systems, i.e., assuming exponentially distributed times (interarrival and service duration). In this regard, our analysis only considers the communication links available or disabled under these types of attacks, as well as the rate of propagation of worms in P2P networks under different conditions. As in the Erlang B model and P2P basic system, there is no consideration of the electronics, computing capabilities, interference, noise, or many other components and variables involved in these systems. The main reason for this is that a mathematical model that considers all of these details may be very hard to produce and even more computationally difficult to solve.

Considering the basic P2P system presented in [44,45], in which exponentially distributed times were considered as well, we aim to extend this model by introducing these two types of cyber attacks, which, given the distributed nature of P2P networks, are very harmful. These attacks are modeled using exponential assumptions, which may differ from practical attacks but give a first approximation to the theoretical performance of these networks. The results are similar to those of the basic model, in which the user interarrivals or dwelling times may not follow an exponential distribution, but the theoretical performance presents a good approximation to real systems (see the results in [45]). Following this, we assume that the DoS attack follows an exponential distribution. Specifically, we assume that the period between attacks and the duration of each attack are random variables with exponential distributions. This may differ from a real attack pattern where botnets are generating such disruptions in the system. In this regard, the exponential model presented in this work has the advantage of presenting a base model that can be easily extended to cover different attack patterns. Indeed, the exponential distribution assumption implies that both the attack interarrival times and attack duration have a coefficient of variation equal to one ($CoV = \sigma_x / E[x]$, where σ_x is the standard deviation, and $E[x]$ is the mean of these times). In the case that the attack pattern has a different CoV, it can be modeled using an Erlang distribution $(if CoV < 1)$, a hyper-exponential distribution $(if CoV > 1)$, or even a Markov modulated poisson process (MMPP) in the case that there are bursts of such attacks. All of these different attack patterns use the basic exponential distribution chain that we developed in this work as the base.

4.1. Malware

Malicious code or malware is the generic term used to designate any informatic program created deliberately to carry out an unauthorized activity that, in numerous instances, is harmful to the system in which it has been lodged. It is a complex piece of software that is capable of complicated attacks, such as collecting all kinds of information. The unauthorized activity of malware (payload) may range from a simple erasure of files to the retrieval and later use of private and/or confidential information (websites visited, contact lists, passwords, account numbers, etc.). Occasionally, the activity performed by the malware may provide some benefit to its creator or disseminator [46]. The flow chart for this attack is depicted in Figure 2.

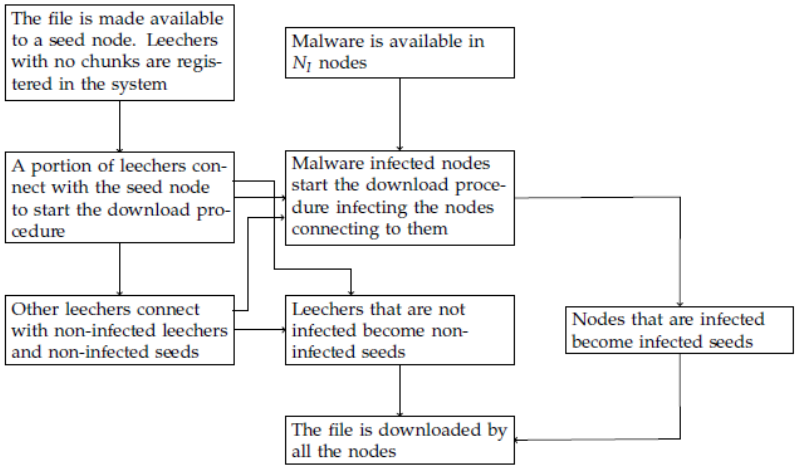


Figure 2. Flow chart for the operation of the malware attack in the P2P network.

Malware is currently one of the main threats to information security. Far from decreasing, this threat (and the effects thereof) will expand considerably in the coming years, mainly because of improvements in its techniques and goals. In the case of malware distribution in P2P networks, the threat and impact are even higher since the intrinsic operation of the system relies on nodes sharing data among peers that are not even known or identified to many nodes, and this file is shared among all users of the system. Hence, it is of paramount importance to quantify the impact of such attacks.

In this work, we consider that the malicious nodes are present from the beginning of the download process, with N_i nodes attacking the network. The rationale for this is that, for these types of attacks, there is no need to maintain an *active* attack since the P2P system is very efficient at propagating the malware given a sufficiently high value of N_i. In this way, the attacker is less likely to get detected since it only provided an initial number of infected seeds, and no new malicious nodes are introduced afterward. This number remains fixed, and no new malicious nodes are added during the operation of the system. We also consider that when a peer enters into contact (i.e., downloads one or more chunks) with an infected peer, it automatically becomes infected. Given the developed analysis, we believe this represents an important baseline model that allows the relaxation of these assumptions in future work if required.

A technique [47] for efficient and effective malware detection is to build models of the malicious samples offline and then verify at run-time if the behavior of a suspicious application adheres to a known model. Ref. [48] used hierarchical behavior specifications to build a model of a malicious program. As the number of malicious samples continues to grow, efficiency is essential, not only for detectors, but also for automatic malware analysis systems. To address this problem, [49] proposed a technique that allows detection of whether a binary is a polymorphic variant of a malware sample that has already been analyzed in the past.

Malicious software has a wide range of analysis avoidance techniques that it can employ to hinder forensic analysis. A review of the literature [50] on malware analysis methodologies found that the most effective methodologies take the presence of analysis avoidance techniques into account [51]. Ref. [52] presented an incremental, static, and dynamic spiral analysis methodology for analyzing malware that additionally molds the analysis environment as the understanding of the malware is attained.

4.2. Denial of Service and Distributed Denial of Service

A denial of service (DoS) attack attempts to make a node or network and its resources unavailable to its intended users by overloading [6]. Figure 3 shows the flow chart for this kind of attack. A distributed denial of service (DDoS) attack usually means that a group of network nodes launches DoS attacks against the same victim. They both cause the

service to stop working by using reasonable service requests to exhaust the resources of the target host. The host sends riddles to its clients before continuing with the requested computation, thus ensuring that the client performs an equally costly computation. DoS and DDoS attacks are very likely to happen in P2P systems. Indeed, in a P2P network, there are numerous participants, and the traffic generated by them is considerably large. Therefore, it is very difficult to predict the traffic between nodes. Hence, attackers can make use of this kind of behavior to overload the network and then disrupt or disable the P2P network. The impact of such an attack is more important when there are millions of concurrently active peers because there is the risk that it could serve as a DDoS engine for attacks against a targeted host. Since any node can act as a router in P2P systems, DDoS attacks are difficult to detect [53–57].

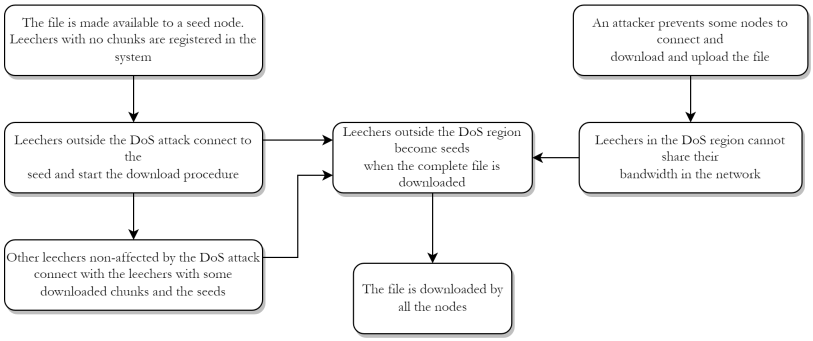


Figure 3. Flow chart for the operation of the DoS attack in the P2P network.

The query flooding a P2P network can be easily attacked by sending a massive number of queries to peers. Furthermore, this attack can be performed by malicious nodes transmitting a jamming signal that causes enough interference in the surrounding nodes, effectively impairing their communication capabilities. Attacks can also use the P2P network as an agent to attack other targets, such as websites. Peers in the network request files from a target and overwhelm the victim with enormous bandwidth usage. Such attacks intend to exhaust resources, paralyzing the capacity of the target. The exhausted resources include the target's CPU processing, downstream bandwidth, upstream bandwidth, etc.

5. Mathematical Model

In this section, we mathematically model both malware and denial of service attacks using continuous-time Markov chains that describe the main dynamics of the system, namely, peer arrivals and departures, file downloading procedure, and infection rates for the malware attack and saturated nodes with impaired communication from the DoS attack. We consider the main and shared characteristics of several attacks to be malware, worms, and poisoning since their behavior and form of attack are very similar. In this way, the parameters are simple, and the findings obtained can be extended to more than one attack in this type of network.

To this end, we use the basic P2P network model presented in [53] that we now describe in detail.

5.1. Basic P2P Model

This model assumes that there are two types of users: leechers and seeds. The Markov chain that describes this system is composed of two states, (x, y), where x is the number of leechers, and y is the number of seeds. Note that there has to be at least one seed in the system to share a file, which is, in fact, the only node that has the complete file at the beginning of the downloading procedure. Hence, this represents an irreducible continuous-time Markov chain with valid state space $(\Omega_{x,y})$ where the state system space (x, y) is with $x = 0, 1, 2, \ldots$, and $y = 1, 2, 3, \ldots$. This chain is irreducible because any state can be accessed by any other state, and all states communicate with each other.

Peers arrive in the system with no chunks (i.e., arriving as leechers) at the rate λ and leave the system before finishing the complete download of the file at the rate θ. Hence, the model considers that nodes become impatient or simply have to leave the downloading procedure for a random time with an exponential distribution and the mean $1/\theta$. Once a leecher downloads the complete file, it becomes a seed at the rate τ. Seeds dwell in the system for a random exponentially distributed time with the mean $1/\gamma$. This passage from leecher to seed can be further explained as follows: Two scenarios of conditions are considered. The first one is *penury*, in which there is a scarcity of users that, in turn, is reflected as a scarcity of resources since there are very few peers sharing the file, and the system's bandwidth is denoted by $\mu[(\eta \cdot x) + y]$, where μ is the upload rate, and η is the parameter that reflects the efficiency of the file sharing among leechers. Indeed, not all leechers that connect among these can interchange chunks of the file because they may have duplicated chunks. This is especially true when there are few leechers in the system or when the chunks are not uniformly distributed by the seeds, i.e., there are chunks that are widely distributed, while other chunks are rare. However, as the number of leechers increases, and the system uses good chunk distribution policies, the efficiency increases, reaching a value close to 0.9. Assuming that nodes can download the file at a maximum rate c, in penury conditions, the available bandwidth is not sufficient for the nodes to download at this rate. Hence, $\mu(\eta \cdot x + y) < (c \cdot x)$. Indeed, the total download rate of the system is $c \cdot x$ because only leechers are downloading the file, and seeds have already completed the download. The second scenario is *abundance*, in which the number of users is sufficient to allow leechers to download at the maximum rate, $(c \cdot x)$. Hence, the leecher-to-seed transition rate can be expressed as:

$$\tau = \min[\mu(\eta \cdot x) + y), cx] \tag{1}$$

We numerically solve the aforementioned chain to obtain the average number of seeds and leechers in the system. Moreover, this Markov chain can be represented as depicted in Figure 4.

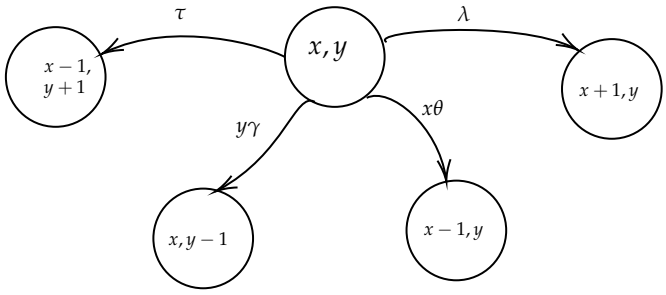


Figure 4. Markov chain of the basic P2P system.

5.2. P2P Infected by Malware

Based on this basic model, we now develop the Markov chain that describes the peer infection procedure. In this model, we assume that there are N_i malicious nodes at the beginning of the file distribution attacking the system. Furthermore, new nodes that arrive, at the rate λ, are not infected, but, if they connect to an infected node, they become infected. For this infection model, we assume that nodes cannot be uninfected, but this restriction will be relaxed in further sections. Hence, the Markov chain that models these dynamics can be formed by four states (X_n, X_i, Y_n, Y_i), where X_n : uninfected leechers, X_n : infected leechers, Y_n : non-infected seeds, Y_i : infected seeds, and the initial state is $(0, 0, 1, N_i)$. Leechers (seeds), whether infected or not, can leave the system at the rate θ (γ).

A non-infected leecher that never downloaded a chunk from an infected peer becomes a non-infected seed at the rate τ_{nn}, given by:

$$\tau_{nn} = \min[cP_nX_n, \mu(\eta \cdot X_n + Y_n)] \tag{2}$$

where P_n is the probability that a peer does not get infected. Note that in order for a leecher to not get infected, all of the available uplink bandwidth, i.e., all of the resources shared with this leecher, have to be free from infected peers. Conversely, an infected leecher that downloaded chunks from infected peers at some point in the file exchange procedure becomes a seed at the rate τ_i

$$\tau_i = \min\left[cX_i, \mu(\eta \cdot X + Y)\frac{X_i}{X}\right] \tag{3}$$

where the total number of leechers is given by $X = X_n + X_i$, and the total number of seeds is $Y = Y_n + Y_i$. For this rate, note that cX_i is the total download rate from infected leechers downloading in abundance conditions, and, in penury, the portion of infected leechers is X_i/X, and only infected peers are uploading to these leechers. If not, i.e., if uninfected peers were exchanging chunks, these uninfected peers would become infected. As such, the only bandwidth considered for an infected leecher to become an infected seed is: $\mu(\eta \cdot X + Y)\frac{X_i}{X}$. Moreover, an uninfected leecher can become an infected seed at the rate:

$$\tau_{ni} = \min[c(1 - P_n)X_n, \mu(\eta \cdot X + Y)(1 - P_n)] \tag{4}$$

In this case, the leecher managed to remain uninfected during the file downloading only until the final chunks when it becomes a seed and downloads the last chunks from an infected source. Then, the uninfected nodes (X_n) are the peers downloading, but they get infected in this process with the probability $(1 - P_n)$. Furthermore, unlike the case of τ_{nn}, all peers can upload the file to these leechers as long as they get infected at this point, explaining the right part of this expression.

To calculate P_n, we have to consider that the total bandwidth in the system is given by $\mu(X\eta + Y)$, while the bandwidth provided by non-infected peers is $\mu(X_n\eta + Y_n)$, and the bandwidth provided by infected peers is $\mu(X_i\eta + Y_i)$. Then, considering that any peer can be connected to any other peer in the system, the probability that only the non-infected bandwidth is used can be expressed as:

$$P_n = \frac{X_n\eta + Y_n}{X\eta + Y} \tag{5}$$

From this, the rate at which leechers become infected during the file-sharing process before becoming seeds, i.e., considering that they can be infected by downloading any possible chunk in the file, can be written as:

$$\tau_c = \min[ck(1 - P_n)X_n, \mu k(\eta \cdot X + Y)(1 - P_n)] \tag{6}$$

where k is the number of chunks, each of size B bytes, forming a file of size F bytes. Then, the number of chunks in a file is given by $k = F/B$. From this, we can see that the chunk download (upload) rate is ck (μk), which is k times faster than the file download (upload) rate.

From this description, when the system is in the state (X_n, X_i, Y_n, Y_i), the valid transitions to any other state are listed as follows and depicted in Figure 5:

- $(X_n + 1, X_i, Y_n, Y_i)$ when a new non-infected leecher arrives at the rate λ.
- $(X_n - 1, X_i, Y_n, Y_i)$ at the rate $X_n\theta$ when a non-infected leecher leaves the system.
- $(X_n, X_i - 1, Y_n, Y_i)$ at the rate $X_i\theta$ when an infected leecher leaves the system.
- $(X_n, X_i, Y_n - 1, Y_i)$ at the rate $Y_n\gamma$ when a non-infected seed leaves the system.
- $(X_n, X_i, Y_n, Y_i - 1)$ at the rate $Y_i\gamma$ when an infected seed leaves the system.

- $(X_n - 1, X_i + 1, Y_n, Y_i)$ when a leecher downloads an infected chunk at the rate τ_c.
- $(X_n - 1, X_i, Y_n + 1, Y_i)$ at the rate τ_{nn} when a non-infected leecher becomes a non-infected seed.
- $(X_n, X_i - 1, Y_n, Y_i + 1)$ at the rate τ_i when an infected leecher becomes an infected seed.
- $(X_n - 1, X_i, Y_n, Y_i + 1)$ at the rate τ_{ni} when a non-infected leecher becomes an infected seed.

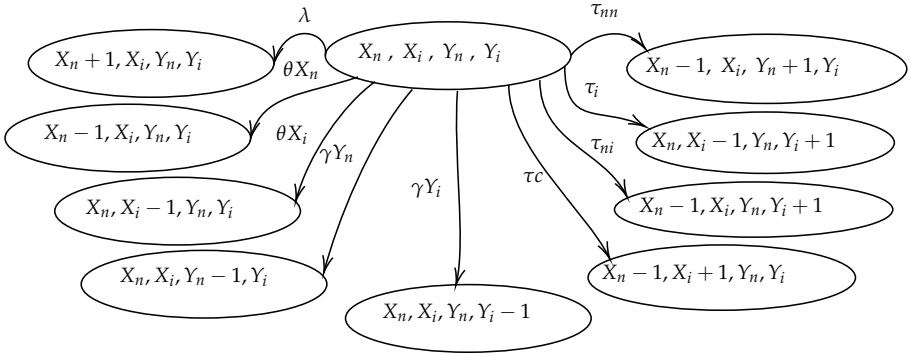


Figure 5. Markov chain of the infection process in a P2P system.

5.3. P2P System Infected by Malware with Countermeasure

In this case, we assume that every time a leecher downloads a chunk, it passes through a revision process, looking for malware. An example of this kind of measure could be antivirus protocols for verifying the origin of the files or quarantine protocols for files that are not known to be trusted. However, not all malware can be detected, and not all peers may have the latest version of the antivirus software, which causes some malware to remain undetected and infect leechers, as in the previous case.

Then, in this chain, the model accounts for the malware revision procedure in the sense that the nodes do not get infected just by downloading an infected chunk. Rather, they become infected only if these countermeasures fail. Then, P_I is the probability that a non-infected node downloading from an Infected node does become contaminated. The closer this parameter is to one, the higher the probability of contagion, and, when it is closer to zero, this indicates that the assumed measures are more effective. Building on this, the Markov chain, depicted in Figure 6, describes the use of countermeasures in a malware attack as very similar to the previously described chain, except that whenever there is a risk of getting infected, the security software avoids infection with the probability P_I. Then, only transitions to the following states are modified:

- $(X_n - 1, X_i + 1, Y_n, Y_i)$ when a leecher downloads an infected chunk and cannot avoid infection at the rate $P_I \tau_c$
- $(X_n - 1, X_i, Y_n, Y_i + 1)$ at the rate $P_I \tau_{ni}$ when a non-infected leecher cannot prevent infection and becomes an infected seed.

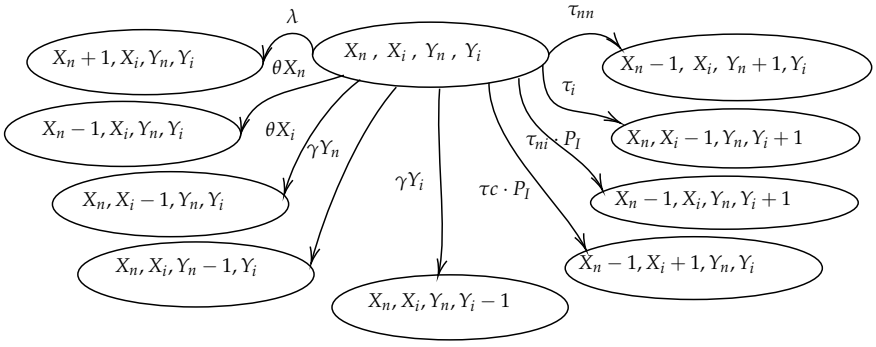


Figure 6. Markov chain of the infection process in a P2P system with countermeasures.

5.4. Markov Chain with a DoS Attack

In this attack, nodes are unable to communicate, preventing resources from sharing with other peers. Building on this, the Markov chain that captures the main dynamics of this system is given the following valid state space (X_n, X_d, Y_n, Y_d) for $(\Omega_{X_n,X_d,Y_n,Y_d} \mid X_n, X_d, Y_n \leq 0, Y_n \leq 1)$ where X_n and Y_n (X_d and Y_d) are the leechers and seeds without an attack (under the DoS attack). From this, when the system is in the state (X_n, X_d, Y_n, Y_d), the possible transitions are shown in Figure 7 and are described as follows:

- To the state $(X_n + 1, X_d, Y_n, Y_d)$ at the rate λ when a new leecher arrives. We assume that new peers are not under the DoS attack.
- To the state $(X_n - 1, X_d, Y_n, Y_d)$ when a leecher without an attack leaves the system at the rate $X_n\theta$.
- To the state $(X_n, X_d - 1, Y_n, Y_d)$ when a leecher under attack leaves the system at the rate $X_n\theta_d$. In this case, we assume that a leecher that realizes that it cannot communicate may choose to leave the system as soon as it detects a possible attack or perceives a malfunction by maybe trying to reset the communication device or leaving the system to re-enter at a future time. As such, the node may leave the system earlier than usual; then, $\theta_d > \theta$.
- To the state $(X_n - 1, X_d, Y_n + 1, Y_d)$ when a leecher without an attack downloads the complete file and becomes a seed. Indeed, if this peer is not under attack while downloading the file, it is very likely that it will remain in the same conditions when it finishes its download process. This occurs at the rate $\tau = \min[CX_n, \mu(\eta X_n + Y_n)]$. It is important to note that only peers without an attack can share their resources, while nodes under the DoS are not considered in the communication bandwidth. Additionally, only the leechers without an attack can become seeds, which explains the left side of this rate.
- To the state $(X_n, X_d, Y_n - 1, Y_d)$ at the rate $Y_n\gamma$ when a seed without an attack leaves the system.
- To the state $(X_n, X_d, Y_n, Y_d - 1)$ at the rate $Y_d\gamma$ when a seed that is being attacked leaves the system. In this case, since the peer has already downloaded the file, it may not detect an ongoing attack. As such, the departure rate remains the same as seeds without any attacks.
- To the state $(X_n - [X_n\delta^l], X_d + [X_n\delta^l], Y_n - [Y_n\delta^s], Y_d + [Y_n\delta^s])$ when a portion of new peers get attacked. Hence, in the case that a malicious node generates a spurious signal in a given area, some nodes inside this region can be leechers (in this case δ^l), and some other nodes can be seeded (δ^s in this model).
- To the state $(X_n + [X_n\delta^l], X_d - [X_n\delta^l], Y_n + [Y_n\delta^s], Y_d - [Y_n\delta^s])$ when the attacker desists its attack in some region. We consider that the average attack time is $1/\Delta$. Then, this occurs at the rate $I(X_d, Y_d)\Delta$ where $I(X_d, Y_d)$ is an indicator function (the end of an attack can only occur if an attack is present in the system) given as:

$$I(X_d, Y_d) = \begin{cases} 0 \text{ if} & X_d \text{ or } Y_d = 0 \\ 1 \text{ if} & X_d \text{ or } Y_d \neq 0 \end{cases}$$

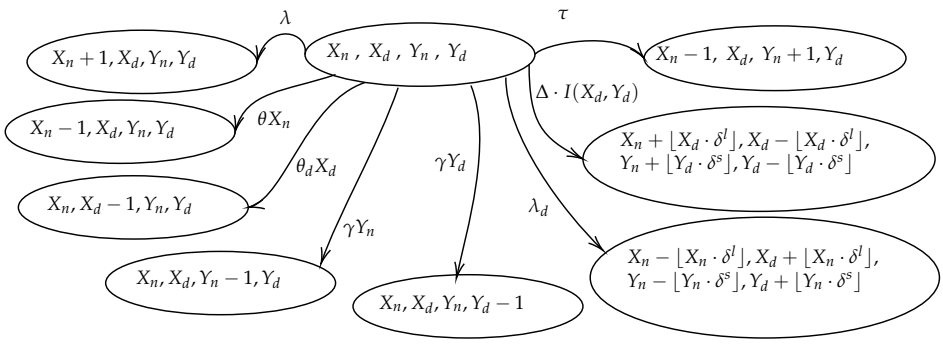


Figure 7. Markov chain of the P2P network under a DoS attack.

6. Numerical Solution of the Markov Chains

The numerical results presented in this section were obtained by numerically solving the previous Markov chains and using Python with the Anaconda package manager and the Jupyter Notebook interface. Other packages within Python that are important for the simulation are random, math, NumPy, Matplotlib, and Pandas.

6.1. Simple Markov Chain

For the basic Markov chain, i.e., when no attack is present, the parameters used for the solutions are presented in Table 2. Algorithm A1 represents the states that were presented in Figure 4. When the system is in the state (x, y), the solution consists of calculating all of the exponential random times associated with the output rates from this state and choosing the minimum time. Then, the next state is the one associated with this minimum time. We store all the times in each state to calculate the stable state probabilities $\pi_{(x,y)}$.

Table 2. Parameters used in the simple chain.

Parameter	Value	Description	Units
c	0.02	Download file rate	files/s
λ	1	Peer arrival rate	users/s
μ	0.00125	Upload file rate	files/s
θ	0.01	Leecher connection time	s
γ	0.01	Seed connection time	s
η	0.85	Download efficiency coefficient	

6.2. Markov Chain with Initially Infected Nodes

For the Markov chain with infected initial nodes, the parameters of the simple chain plus the parameters explained in Table 3 were taken. The basic model corresponds to the normal and conventional operation of the P2P system. However, when an active attack is present, the system would be degraded according to the intensity of the attack. For the *malware* attack, the intensity of the attack is reflected in the initial number of malicious nodes, N_i, while for the DoS attack, the intensity of the attack is reflected in the time of the attack and number of nodes affected by it, given by $1/\delta$, δ^l, and δ^s, correspondingly.

Table 3. Added parameters to the chain with infected nodes.

Parameter	Value	Description	Units
K	1/10	File chunks	size/chunks
X_i	-	Infected leechers	peers
Y_i	-	Infected seeds	peers
N_i	10	Initial number of infected seeds	peers

Algorithm A2 represents the numerical solution of the Markov chain that was presented in Figure 5 . For this string, we also varied three different parameters over a certain range. These parameters are γ, which represents the seed connection time; μ, which indicates the download rate; and λ, which is the arrival rate of new users connecting to the network per second. Three different experiments were performed to vary each of the above parameters against the number of initially infected nodes (N_i), which represents the intensity of the attack, to study the performance of the system in terms of the number of healthy or infected peers and determine which value or range of values favored or harmed the growth of the infected nodes. The value of K was set to 0.1, although this value can change depending on the specific conditions of the system.

6.3. Markov Chain with Infected Initial Nodes and Countermeasures

In this case, similar to the previous Markov chain, a comparison was made between the number of initially infected nodes and the probability P_I that a healthy node downloading files from an infected node does not get contaminated or that the contaminated chunks are passed through the security software to disinfect it. Note that the model does not consider the case in which the infected chunks are discarded. Indeed, they are disinfected, with the probability P_I or not detected, and then, the leecher gets infected with the probability $1 - P_I$, which implies that the security software could not detect the malware, to observe how it affects the behavior of seeds and leechers, whether healthy or infected. In this experiment, the range of initially infected nodes was extended, starting from 1 to 14 nodes, and the infection rate was carried out within the range of 0 to 1 with steps of 0.01.

In Algorithm A3, we can see that P_I is multiplied by the rates associated with states 6 and 7, as these states affect the growth of the infected pairs.

7. Numerical Results

In this section, we show the most representative results that could be used in future P2P networks that suffer cyber attacks. First, we show well-known results for the basic model that shows the normal operation of the system to clearly see the effects of the different attacks on the network.

7.1. Basic P2P Network

These results were obtained using the parameters and values presented earlier. Specifically, c = 0.02 (files/s), λ = 1 (users/s), μ = 0.00125 (files/s), θ = 0.01 (leecher departures per sec), γ = 0.01 (seed departures per sec), and η = 0.85. In Figure 8, we can see the number of peers (both leechers and seeds) who go through an initialization phase during which the system is found in a transitory mode where there are few peers, especially seeds, and the file sharing is very inefficient due to the lack of resources. As time passes, the system enters a stable state where the number of peers varies around a clear average value. This is because the number of resources is sufficiently high, and most leechers find either a leecher that can share the missing chunks or seeds that remain in the system. Note that the number of peers continues to vary throughout the realization of the experiment, but it clearly does so around a fixed value.

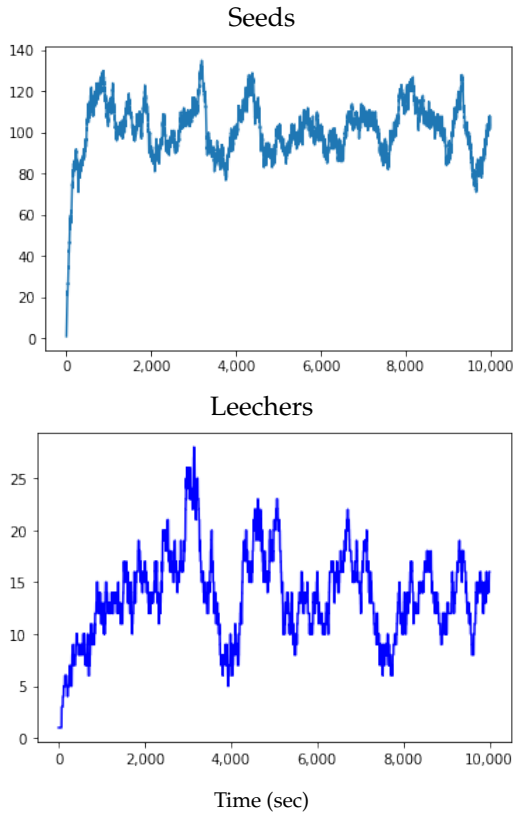


Figure 8. Evolution in time of one realization of the number of leechers and seeds in the system.

For this experiment, the average number of leechers obtained was 100, and the average number of seeds was 16, as shown in Table 4.

Table 4. Averages and simulation time of the simple Markov chain for each type of peer.

Average in 100,000 Iterations	
Leechers	**Seeds**
100 ± 15	16 ± 3
Approximate running time: 25 min	

7.2. P2P System with a Malware Attack

For these results, we used the following set of parameters: $c = 0.002$ (files/s), $\lambda = 1$ (arrival rate, users/s), $\mu = 0.05$ (files/s), $\theta = 0.005$ (leecher departures per sec), $\gamma = 0.3$ (seed departures per sec), $\eta = 0.85$, and $K = 1/10$ (chunk length), and now, we also use an initial number of infected seeds of $N_i = 10$. In this case, the evaluation time was significantly increased due to the increased complexity of the system, as shown in Table 5. The results shown in Figure 9 clearly show that the system also reaches a stable state after a transitory phase. This includes the number of infected peers. Indeed, for this initial number of malicious nodes sharing the malware in their chunks, not all of the system gets infected; it is much less than half of the leechers and half of the number of seeds. These are essential results because, using this model, the system administrator can estimate the number of malicious nodes attacking the system by only observing the resulting number of infected peers in time.

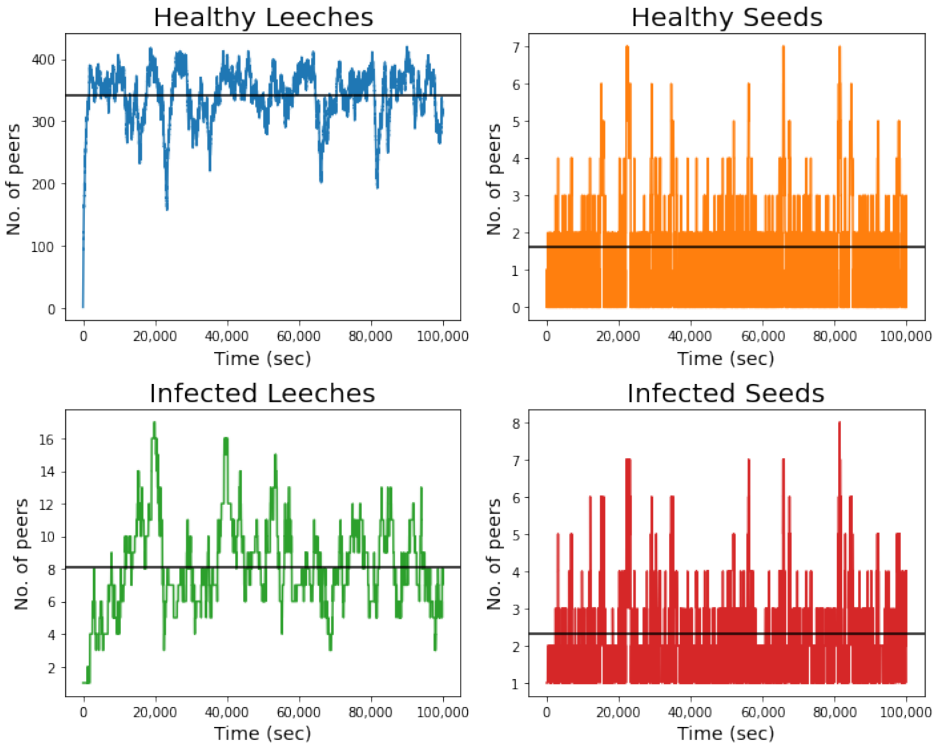
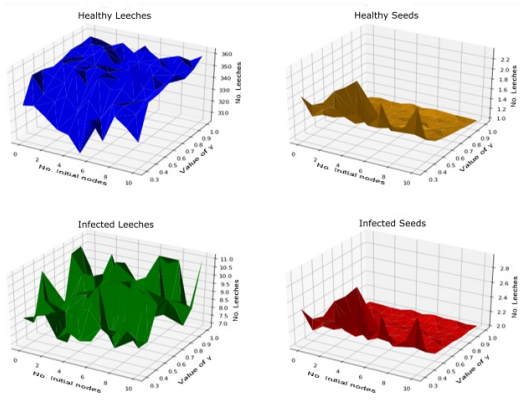


Figure 9. Evaluation in time of one realization of the number of healthy and infected leechers and seeds.

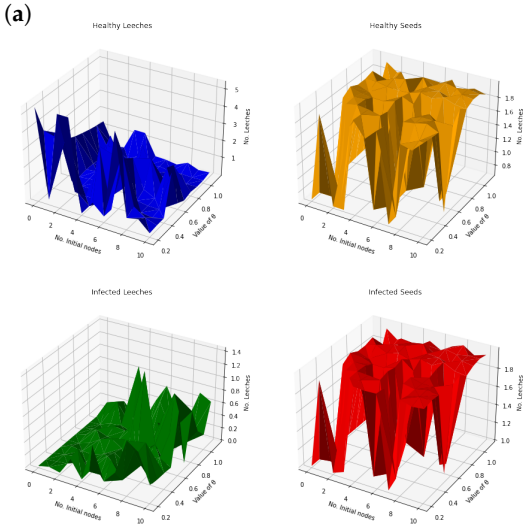
Table 5. Average results for each type of peer in the network together with the simulation time of the Markov chain with malicious nodes.

Average in 100,000 Iterations			
Healthy Leechers	**Healthy Seeds**	**Infected Leechers**	**Infected Seeds**
33 ± 8	2 ± 1	10 ± 5	2 ± 2
Approximate time: 37 min			

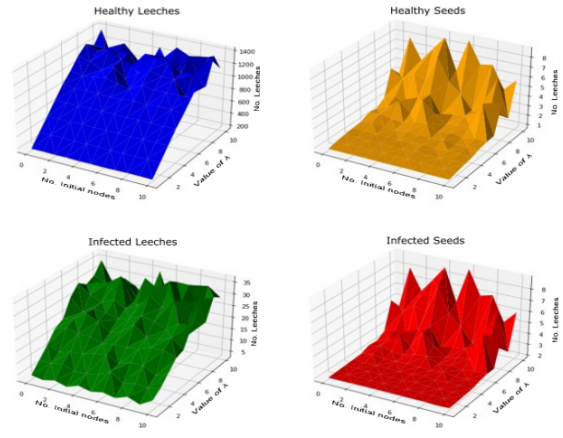
To further study the system performance of the system under a malware attack, we vary many of the system parameters, such as γ, θ, λ, and N_i see as Figure 10. From these results, we can see that the value of γ has no important impact on the malware process since both healthy and infected peers vary in the same proportion for any value of the initial numerous malicious nodes. However, for the case of θ, we can see that for variations in the departure rate of leechers, the system becomes very sensitive to the number of initial malicious nodes, i.e., an increasing N_i entails a higher number of infected nodes, which did not occur when varying γ. In the case of the arrival rate, we case see an expected increase in the number of infected peers as the leecher arrival rate increases. This occurs simply because there are more peers in the system sharing their resources, which makes it easier for nodes to get infected.



(**a**)



(**b**)



(**c**)

Figure 10. Average number of healthy and infected peers under a malware attack for different values of γ, η, and λ. (**a**) Variation of infected nodes and γ. (**b**) Variation of infected nodes and θ. (**c**) Variation of infected nodes and λ.

7.3. P2P System with Countermeasures for the Malware Attack

In this case, the system has incorporated different countermeasures to avoid infections of the malware. However, no countermeasures are completely effective, and some peers still get infected. This vulnerability of cybersecurity countermeasures is modeled with the probability P_I. Hence, low (high) values of P_I, correspond to the case in which either

the countermeasures are very effective (inefficient) or the malware attack is weak (strong) and cannot (can) infect nodes. This effect is clearly seen in Figure 11 where infected peers drastically diminish as P_I approaches the value of 0, avoiding most of the potential infections in the system. These results were obtained using $\gamma = 1$, $\eta = 0.85$, and the new user arrival rate $\lambda = 1$. The running time was 1 hour and 33 minutes, which exceeded the previous execution times, accounting for the added complexity of the system. Note that the number of initial nodes no longer has any effect on the number of infected nodes. As such, the introduction of countermeasures is an effective way to control the spread of malware. From the practical side, the system administrator can clearly see from the number of infected nodes the effectiveness of the countermeasures to produce more effective tools if required. Moreover, as time passes, some countermeasures that were efficient in the beginning may become obsolete as time passes by, which would be reflected in a higher portion of infected nodes.

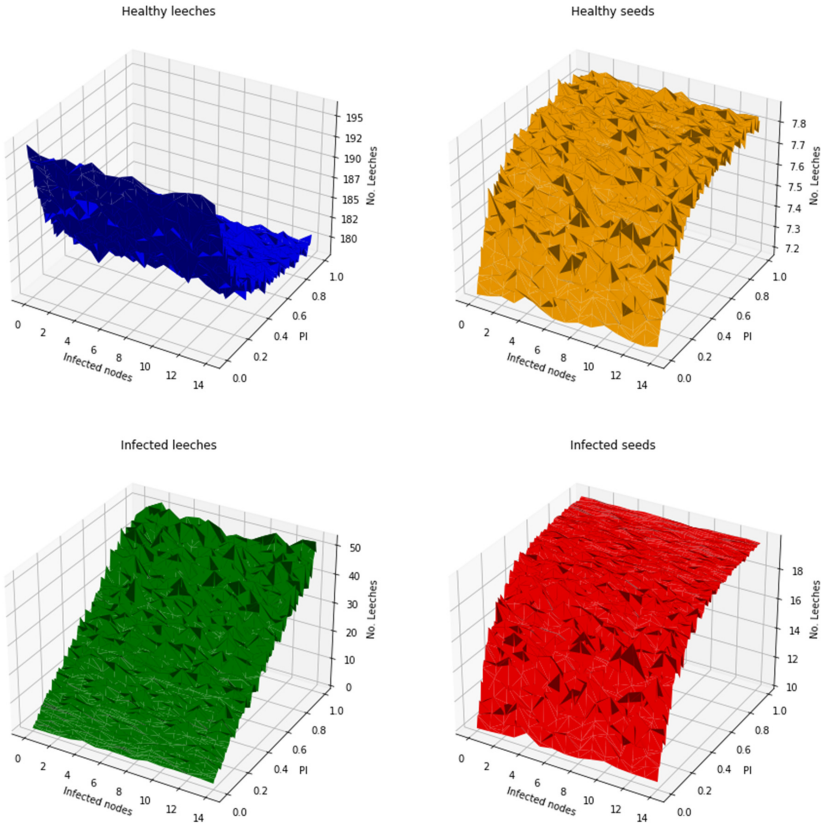


Figure 11. Average number of healthy and infected nodes for different numbers of initial malicious nodes, Ni, and efficiency of the countermeasures, PI.

7.4. P2P System under a Denial of Service Attack

For this attack, we no longer have infected nodes, but rather, nodes that cannot communicate since they are under a DoS attack. Hence, a malicious node jams the communication capabilities of a certain number of nodes, given by δs and δl in the previously presented model. For a high-level attack, i.e., a node that has the capability of attacking many nodes simultaneously, these parameters will also be high. For a stealthy attack, the attacker may choose a low value for these parameters. In Figure 12, we present the results for a value of $\delta_s = \delta_l = 5$, for which, when an attack is present, there are five nodes on average jammed by the attacker.

We now investigate the system's performance for different conditions of the attack. We first present the evolution in time of the P2P system under this attack, as depicted in Figure 12. For these results, we consider that there is an initial number of five jammed

peers, and every time the attack is active there are five peers jammed. We can clearly see that, at the beginning of the attack, there are only five seeds jammed in the system that cannot communicate with the rest of their peers. As time passes by, this initial number of seeds decreases since they eventually leave the system. Thereafter, the attack mainly affects the leechers due to the proportion of seeds in the system, i.e., there are much more leechers, and the probability that the node under attack is a leecher is much greater than that it is a seed.

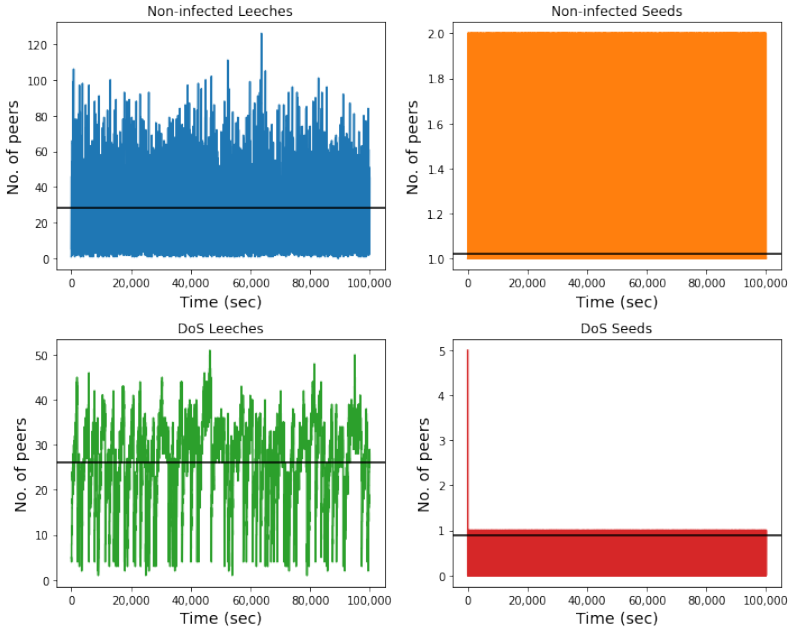


Figure 12. Time evolution of the P2P system with $\delta = 5$ nodes affected by the DoS attack.

We now present the average number of leechers and seeds with and without the jamming attack for different attack intensities. Note that the intensity of the attack directly depends on the number of nodes that are jammed when an attack begins (for the leechers, it is δ_l, and, for the seeds, it is δ_s) and the average time of such an attack, $1/\Delta$. Our model is sufficiently flexible to allow the number of leechers and seeds being jammed to be different. This is the case in a more sophisticated attack where the hacker has the capacity to recognize that a node is either a leecher or seed and jam its radio communications for a specific time. For instance, he may choose to jam only seeds, in which case $\delta_l = 0$ and $\delta_s > 0$, or only leechers where $\delta_l > 0$ and $\delta_s = 0$. Table 6 shows the case in which the leechers and seeds are affected in the same proportion. It also shows the case in which δ_s $\{\delta_l\}$ remains constant and the number of leechers {seeds} under attack varies.

These different attack scenarios produce very interesting results, such as the ones presented in Table 6, in which both the leechers and seeds are jammed in the same proportion. We can see that for some peers under the DoS attack, the number of infected peers increases, but this entails a low number of communicating peers, i.e., the number of peers that can actually share their data gets lower and lower until the point at which no communication is possible among peers, and the population is almost zero. These results prove that there is an attacking intensity where the attack can be relatively stealthy, in which peers can still communicate, and the network is still operational. This occurs at the values of δ_s and δ_l and affects around 40% of the peers. After this point, the system is severely altered, and the attack would be detected rapidly.

Another important result is presented in Table 6, where we can see that if only one population of the system is attacked, either leechers or seeds, only that population decreases to zero. Specifically, for the case in which leechers are attacked, but not seeds, arriving leechers, which are not under attack, can find many resources available since the attacked

leechers are not consuming any bandwidth. This allows an expedited file download since these few leechers are all connected to the seeds with all the chunks, and so, we still have successful file downloads. In the case in which only seeds are impacted, the file download process is severely hindered after a value of δ_s since only leechers cannot find all of the chunks of the file, and no more leechers convert into seeds. We can see that, in both of these last cases, the average attack time of $1/\Delta$ is clear in the sense that if the attack has a long duration (low values of Δ), the number of seeds or leechers also goes to zero, even if they are not directly affected. However, the most important parameter for predicting the impact of the attack is the proportion of peers that are jammed when an attack occurs.

Table 6. System performance of a P2P system under a DoS attack for different attack intensity varying δ_l, δ_s, and Δ.

Δ	δs	Healthy Leechers	Healthy Seeds	DoS Leechers	DoS Seeds
1	0.00	9.155	53.013	7.811	2.924
1	0.25	9.784	4.406	9.884	3.967
1	0.50	10.576	1.614	11.208	1.658
1	0.75	9.410	1.167	10.214	1.123
40	0.00	9.727	52.927	151.998	2.659
40	0.25	7.349	3.669	133.747	15.693
40	0.50	4.725	1.177	208.656	11.323
40	0.75	12.693	1.817	116.988	4.143
80	0.00	9.033	52.525	146.615	2.544
80	0.25	7.172	3.556	154.939	16.503
80	0.50	5.685	1.238	223.982	10.481
80	0.75	6.836	1.253	180.951	7.281
Δ	**δl**	**Healthy Leechers**	**Healthy Seeds**	**DoS Leechers**	**DoS Seeds**
1	0.00	429.252	2.289	4.119	2.099
1	0.25	41.452	1.946	54.870	2.082
1	0.50	8.488	1.451	7.729	1.401
1	0.75	4.944	1.519	2.153	1.394
40	0.00	431.542	1.883	1.713	17.139
40	0.25	19.523	1.814	285.543	8.148
40	0.50	7.048	1.369	117.530	9.321
40	0.75	4.846	1.390	39.069	8.573
80	0.00	427.611	1.962	2.633	20.930
80	0.25	15.665	1.461	329.619	15.223
80	0.50	7.597	1.432	170.413	11.722
80	0.75	6.602	1.603	36.358	10.091
Δ	**δs and δl**	**Healthy Leechers**	**Healthy Seeds**	**DoS Leechers**	**DoS Seeds**
1	0.00	439.798	97.388	4.694	2.283
1	0.25	38.829	5.531	39.261	4.513
1	0.50	9.744	1.517	9.152	1.405
1	0.75	4.192	1.221	2.164	1.118
40	0.00	404.865	112.855	2.692	2.889
40	0.25	18.157	3.590	277.356	14.842
40	0.50	6.161	1.357	159.849	9.440
40	0.75	3.824	1.156	28.004	3.046
80	0.00	423.479	117.059	3.290	3.241
80	0.25	12.866	3.532	307.616	17.266
80	0.50	5.735	1.301	163.830	11.180
80	0.75	4.727	1.262	80.808	8.086

With our results, it is essential to highlight that this work is not intended to detect cyber attacks but rather to give theoretical results on the performance effects of both worms and DoS attacks in P2P networks when these attacks follow exponential distributions. After examining the numerical results, we observed that, especially for the DoS attacks, it is possible to see a clear deterioration in the communication capabilities, reflected in the number of leechers that can download the file and also in the number of seeds in the network. (This is more evident when the duration of the attack and the number of nodes under attack is high, as presented in Table 6 of the manuscript.) However, this is clearly not an accurate method for detecting DoS attacks since this effect can also be a product

of high levels of interference or noise that occur for reasons (such as other systems in the area or traffic peaks in the zone) other than attacks. For the worm dissemination attack, the number of leechers and seeds is not affected by the sharing procedure of the worm. Hence, in that case, the attack has to be detected by cybersecurity software installed in the downloading nodes. Therefore, this analytical model can only be used to know in advance the propagation rate of the worm with and without cybersecurity countermeasures.

8. Conclusions

In this work, we mathematically model, study, and analyze the effect of two common attacks on P2P networks, namely, worms and DoS attacks. Different parameters of the attacks are considered in our model to clearly see the effectiveness of such attacks, which may provide different counterattack mechanisms for future network deployments or at least provide effective tools for detecting such attacks. The effectiveness of the attacks is visible in terms of the number of infected and healthy peers and also in the proportion of leechers successfully downloading the file. More specifically, for the worm attack, we are interested in evaluating the rate at which the malicious software is spread, while, in the DoS attack, we explore the impact on the communicating capacities of the system.

To this end, we developed and numerically solved different Markov chains that abstract the main dynamics of the P2P networks that are under attack in different scenarios and intensities. Hence, these models and results can be extended to other types of attacks or different scenarios of the same attacks considered in our work by adjusting the involved variables and developing effective countermeasures before a real attack occurs.

Author Contributions: Conceptualization, M.E.R.-A.; Methodology, M.E.R.-A.; Investigation, N.S.-P.; Data curation, N.S.-P.; Writing—original draft, N.S.-P. and G.G.-G.; Writing—review & editing, M.E.R.-A.; Supervision, G.G.-G.; Funding acquisition, G.G.-G. and M.E.R.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by CONACyT grant numbers 321068, SIP-20230710, and SIP-20231239.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Algorithms

Algorithm A1: Simple Chain

```
x=0;
y=1;
for i=1 to 100,000 do
    tau = min(M·(N·x+y),C·x);
    T1 = -(1/L) · ln(1-u());
    T2 = -(1/(H·x)) · ln(1-u());
    T3 = -(1/(G·y)) · ln(1-u());
    T4 = -(1/tau) · ln(1-u());
    T = min(T1,T2,T3,T4);
    ;
    if T=T1 then
        x+=1 ;
    else if T=T2 then
        x-=1 ;
    else if T=T3 then
        x-=1 ;
        y-=1 ;
    else
        x+=1 ;
        y+=1 ;
    end
```

Result: Number of x and y in total and at each step.

Algorithm A2: Chain with infected nodes

```
xn=0;
yn=1;
xi=0;
yi=1;
for i=1 to 100,000 do
    Pn = (xn·M+yn)/(xn+(yn+yi));
    tauni = min(C·(1-Pn)·xi,M·(N·xn+yn)·(1-Pn));
    taunn = min(C·Pn·xn,M·(N*xn+yn));
    taui = min(C·xi,((xn+xi)·N+(yn+yi)·(xi/(xn+xi))));
    tauC = min(C·K·(1-Pn)·xn,M·K·(M·(xn+xi)+(yn+yi))·(1-Pn));
    T1 = -(1/L) · ln(1-u());
    T2 = -(1/(H·xn)) · ln(1-u());
    T3 = -(1/(H·xi)) · ln(1-u());
    T4 = -(1/G·yn) · ln(1-u());
    T5 = -(1/(G·yi)) · ln(1-u());
    T6 = -(1/(tauC)) · ln(1-u());
    T7 = -(1/(tauni)) · ln(1-u());
    T8 = -(1/taui) · ln(1-u());
    T9 = -(1/taunn) · ln(1-u());
    T = min(T1,T2,T3,T4,T5,T6,T7,T8,T9);
    if T=T1 then
        xn += 1 ;
    else if T=T2 then
        xn -= 1 ;
    else if T=T3 then
        xi -= 1 ;
    else if T=T4 then
        yn -= 1 ;
    else if T=T5 then
        yi -= 1 ;
    else if T=T6 then
        xn -= 1 ;
        xi += 1 ;
    else if T=T7 then
        xn -= 1 ;
        yi += 1 ;
    else if T=T8 then
        xi -= 1 ;
        yi += 1 ;
    else
        xn -= 1 ;
        yn += 1 ;
    end
```

Result: Number of *xn*, *xi*, *yn*, and *yi* in total and at each step.

Algorithm A3: Chain adding a parameter as a countermeasure

```
for Ni=1 to 15 do
    for PI=0 to 1 do
        for i=1 to 100,000 do
            Pn = (xn·M+yn)/(xn+(yn+yi));
            tauni = min(C·(1-Pn)·xi,M·(N·xn+yn)·(1-Pn));
            taunn = min(C·Pn·xn,M·(N*xn+yn));
            taui = min(C·xi,((xn+xi)·N+(yn+yi)·(xi/(xn+xi))));
            tauC = min(C·K·(1-Pn)·xn,M·K·(M·(xn+xi)+(yn+yi))·(1-Pn));
            T1 = -(1/L) · ln(1-u());
            T2 = -(1/(H·xn)) · ln(1-u());
            T3 = -(1/(H·xi)) · ln(1-u());
            T4 = -(1/G·yn) · ln(1-u());
            T5 = -(1/(G·yi)) · ln(1-u());
            T6 = -(1/(tauC·PI)) · ln(1-u());
            T7 = -(1/(tauni·PI)) · ln(1-u());
            T8 = -(1/taui) · ln(1-u());
            T9 = -(1/taunn) · ln(1-u());
            T = min(T1,T2,T3,T4,T5,T6,T7,T8,T9);
            if T=T1 then
                xn += 1 ;
            else if T=T2 then
                xn -= 1 ;
            else if T=T3 then
                xi -= 1 ;
            else if T=T4 then
                yn -= 1 ;
            else if T=T5 then
                yi -= 1 ;
            else if T=T6 then
                xn -= 1 ;
                xi += 1 ;
            else if T=T7 then
                xn -= 1 ;
                yi += 1 ;
            else if T=T8 then
                xi -= 1 ;
                yi += 1 ;
            else
                xn -= 1 ;
                yn += 1 ;
            end
```

Result: Number of *xn*, *xi*, *yn*, and *yi* in total and at each step.

References

1. Quang Hieu, V.; Mihai Lupu, B.C.O. *Peer-to-Peer Computing: Principles and Applications*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 16.
2. Esquivel, E.E.B.; Rivero-Angeles, M.E.; Fernandez-Vazquez, A. Priority scheme for mobile nodes in P2P bit-torrent based networks. In Proceedings of the 2014 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 13–16 May 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 469–473.
3. Deng, L.; He, Y.; Xu, Z. Combating index poisoning in P2P file sharing. *Adv. Inf. Secur. Assur.* **2009**, *5576*, 358–367. [CrossRef]
4. Wang, W.; Zhao, W. Model the P2P Attack in Computer Networks. In Proceedings of the International Conference on Logistics, Engineering, Management and Computer Science, Shenyang, China, 29–31 July 2015; pp. 1303–1307. [CrossRef]

5. Jin, X.; Chan, S.H.G. Unstructured peer-to-peer network architectures. In *Handbook of Peer-to-Peer Networking*; Springer: Boston, MA, USA 2010; pp. 117–142.
6. Chaganti, R.; Boppana, R.V.; Ravi, V.; Munir, K.; Almutairi, M.; Rustam, F.; Lee, E.; Ashraf, I. A Comprehensive Review of Denial of Service Attacks in Blockchain Ecosystem and Open Challenges. *IEEE Access* **2022**, *10*, 96538–96555. [CrossRef]
7. Wararkar, P.; Kapil, N.; Rehani, V.; Mehra, Y.; Bhatnagar, Y. Resolving Problems Based on Peer to Peer Network Security Issue's. In Proceedings of the Procedia Computer Science, Nagpur, India, 11–12 December 2015; Elsevier B.V.: Amsterdam, The Netherlands, 2016; Volume 78, pp. 652–659. [CrossRef]
8. Chumachenko, D.; Chumachenko, K.; Yakovlev, S. Intelligent simulation of network worm propagation using the code red as an example. *Telecommun. Radio Eng.* **2019**, *78*, 443–464. [CrossRef]
9. Rajesh, B.; Reddy, Y.J.; Reddy, B.D.K. A survey paper on malicious computer worms. *Int. J. Adv. Res. Comput. Sci. Technol.* **2015**, *3*, 161–167.
10. Frauenthal, J.C. *Mathematical Modeling in Epidemiology*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
11. Gourieroux, C.; Jasiak, J. Analysis of Virus Propagation: A Transition Model Representation of Stochastic Epidemiological Models. *arXiv* **2020**, arXiv:2006.10265.
12. Rasheed, M.M.; Norwawi, N.M.; Ghazali, O.; Faaeq, M.K. Detection algorithm for internet worms scanning that used user datagram protocol. *Int. J. Inf. Comput. Secur.* **2019**, *11*, 17–32. [CrossRef]
13. Awasthi, S.; Kumar, N.; Srivastava, P.K. A study of epidemic approach for worm propagation in wireless sensor network. In *Advances in Intelligent Computing in Engineering*. Book Series AISC Vol. 1125. Springer: Singapore, 2020; pp. 315–326.
14. Andersson, H.; Britton, T. *Stochastic Epidemic Models and Their Statistical Analysis*; Springer Science & Business Media 2000: New York, NY, USA, 2012; Volume 151.
15. Zhang, Z.; Kundu, S.; Wei, R. A delayed epidemic model for propagation of malicious codes in wireless sensor network. *Mathematics* **2019**, *7*, 396. [CrossRef]
16. Ahmad, M.A.; Woodhead, S. Containment of fast scanning computer network worms. In Proceedings of the International Conference on Internet and Distributed Computing Systems, Windsor, UK, 2–4 September 2015; Springer: Cham, Switzerland, 2015; pp. 235–247.
17. Malatras, A. State-of-the-art survey on P2P overlay networks in pervasive computing environments. *J. Netw. Comput. Appl.* **2015**, *55*, 1–23. [CrossRef]
18. Washbourne, L. A survey of P2P Network security. *arXiv* **2015**, arXiv:1504.01358.
19. Tang, H.; Liu, Y.; Huang, J. A worm counter-measurement strategy in P2P networks: Modeling and analysis. In Proceedings of the 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, Hangzhou, China, 30 October–1 November 2012; IEEE: Piscataway, NJ, USA, 2012; Volume 2, pp. 706–710.
20. Jaideep, G.; Battula, B.P. Survey on the present state-of-the-art of P2P networks, their security issues and counter measures. *Int. J. Appl. Eng. Res.* **2016**, *11*, 616–620. [CrossRef]
21. Gopalakrishnan, K. Security vulnerabilities and issues of traditional wireless sensors networks in IoT. In *Principles of Internet of Things (IoT) Ecosystem: Insight Paradigm*; Intelligent Systems Reference Library; Springer: Cham, Switzerland, 2020; Volume 174, pp. 519–549.
22. Ferdous, S.; Chowdhury, F.; Moniruzzaman, M. A Taxonomy of Attack Methods on Peer-to-Peer Network. In Proceedings of the 1st Indian Conference on Computational Intelligence and information Security, 2007; Volume 2007, pp. 132–138.
23. Bhise, A.M.; Kamble, S.D. Detection and mitigation of Sybil attack in peer-to-peer network. *Int. J. Comput. Netw. Inf. Secur.* **2016**, *8*, 56. [CrossRef]
24. Cerri, D.; Ghioni, A.; Paraboschi, S.; Tiraboschi, S. ID mapping attacks in P2P networks. In Proceedings of the GLOBECOM '05, IEEE Global Telecommunications Conference, Rio de Janeiro, Brazil, 4–8 December 2005; Volume 3. [CrossRef]
25. Trifa, Z.; Khemakhem, M. Taxonomy of structured P2P overlay networks security attacks. *Int. J. Comput. Inf. Eng.* **2012**, *6*, 470–476.
26. Thommes, R.; Coates, M. Modeling Virus Propagation in Peer-to-Peer Networks. In Proceedings of the 2005 5th International Conference on Information Communications & Signal Processing Bangkok, Thailand, 6–9 December 2005; pp. 981–985. [CrossRef]
27. Rohloff, K.; Basar, T. Stochastic behavior of random constant scanning worms. In Proceedings of the 14th International Conference on Computer Communications and Networks, San Diego, CA, USA, 17–19 October 2005; pp. 339–344. [CrossRef]
28. Benevenuto, F.; Costa, C.; Vasconcelos, M.; Almeida, V.; Almeida, J.; Mowbray, M. Impact of Peer Incentives on the Dissemination of Polluted Content. In Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon, France, 23–27 April 2006; Association for Computing Machinery: New York, NY, USA, 2006; pp. 1875–1879. [CrossRef]
29. Thommes, R.; Coates, M. Epidemiological Modelling of Peer-to-Peer Viruses and Pollution. In Proceedings of the IEEE INFOCOM 2006, 25th IEEE International Conference on Computer Communications, Barcelona, Spain, 23–29 April 2006. [CrossRef]
30. Kumar, R.; Yao, D.D.; Bagchi, A.; Ross, K.W.; Rubenstein, D. Fluid Modeling of Pollution Proliferation in P2P Networks. *ACM SIGMETRICS Perform. Eval. Rev.* **2006**, *34*, 335–346. [CrossRef]
31. Feng, C.S.; Qin, Z.G.; Cuthbet, L.; Tokarchuk, L. Propagation modeling and analysis of viruses in P2P networks. In Proceedings of the 2008 International Conference on Machine Learning and Cybernetics, Kunming, China, 12–17 July 2008; Volume 7, pp. 3635–3640. [CrossRef]

32. Liu, M.; Han, L.; Hong, F.; Zou, M. A Computer Virus Propagation Model in P2P Networks. In Proceedings of the 2009 First International Workshop on Education Technology and Computer Science, Washington, DC, USA, 7–8 March 2009; Volume 1, pp. 578–581. [CrossRef]
33. Ebrahim, M.; Talha, S.M.U.; Ahmad, J. Modeling Virus Propagation in Pure Peer-to-Peer Networks. In Proceedings of the 8th International Conference on Frontiers of Information Technology, Islamabad, Pakistan, 21–23 December 2010; Association for Computing Machinery: New York, NY, USA, 2010. [CrossRef]
34. Xu X.; Geng, Y.; Xiong, J. Model of Proactive Malicious Code Propagation and Vaccine Contradiction in P2P Network. *Trans. Beijing Inst. Technol.* **2013**, *33*, 605.
35. Liu, X.; Liu, J. Novel non-linear dynamics P2P network worm propagation and immune model. *IET Inf. Secur.* **2020**, *14*, 175–184. [CrossRef]
36. Zhou, Y.; Wu, Z.; Ye, C.; Zhong, J.; Wang, H.; Feng, Y. Proactive worm prevention based on P2P networks. In Proceedings of the 2006 IET International Conference on Wireless, Mobile and Multimedia Networks, Hangzhou, China, 6–9 November 2006; pp. 1–4.
37. Shi, C.; Han, D.; Hu, X.; Yu, Y. A unified model of pollution in P2P networks. In Proceedings of the 2008 IEEE International Symposium on Parallel and Distributed Processing, Miami, FL, USA, 14–18 April 2008; pp. 1–12. [CrossRef]
38. Peng, H.; Lu, S.; Zhao, D.; Zhang, A.; Li, J. An anti-attack model based on complex network theory in P2P networks. *Phys. A Stat. Mech. Its Appl.* **2012**, *391*, 2788–2793. [CrossRef]
39. Faheem Rasheed, M. Modelling Virus Propagation in P2P Networks. *Int. J. Comput. Sci.* **2012**, *9*, 580–587.
40. Kumar, P.; Gupta, G.; Tripathi, R. A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 9555–9572. [CrossRef]
41. Kumar, P.; Kumar, R.; Gupta, G.P.; Tripathi, R.; Srivastava, G. P2TIF: A Blockchain and Deep Learning Framework for Privacy-Preserved Threat Intelligence in Industrial IoT. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6358–6367. [CrossRef]
42. Kumar, P.; Gupta, G.; Tripathi, R. Toward Design of an Intelligent Cyber Attack Detection System using Hybrid Feature Reduced Approach for IoT Networks. *Arab. J. Sci. Eng.* **2021**, *46*, 3749–3778. [CrossRef]
43. Kumar, P.; Gupta, G.; Tripathi, R. Design of Anomaly-Based Intrusion Detection System Using Fog Computing for IoT Network. *Autom. Control Comput. Sci.* **2021**, *55*, 137–147. [CrossRef]
44. Veciana, G.; Yang, X. Fairness, incentives and performance in peer-to-peer networks. *Seeds* **2003**, *250*, 350.
45. Qiu, D.; Srikant, R. Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks. *ACM SIGCOMM Comput. Commun. Rev.* **2004**, *34*, 367–378. [CrossRef]
46. del Rey, A.M. Mathematical modeling of the propagation of malware. *Secur. Commun. Netw.* **2015**, *5*, 422–437. [CrossRef]
47. Kolbitsch, C.; Comparetti, P.M.; Kruegel, C.; Kirda, E.; Zhou, X.y.; Wang, X. Effective and efficient malware detection at the end host. In Proceedings of the USENIX Security Symposium, Montreal, QC, Canada, 12–14 August 2009; Volume 4, pp. 351–366.
48. Martignoni, L.; Stinson, E.; Fredrikson, M.; Jha, S.; Mitchell, J.C. A layered architecture for detecting malicious behaviors. In Proceedings of the International Workshop on Recent Advances in Intrusion Detection, Cambridge, MA, USA, 15–17 September 2008; Springer: Berlin/Heidelberg, Germany, 2008; pp. 78–97.
49. Kharraz, A.; Kirda, E. Redemption: Real-time protection against ransomware at end-hosts. In Proceedings of the International Symposium on Research in Attacks, Intrusions, and Defenses, Atlanta, GA, USA, 18–20 September 2017; Springer: Cham, Switzerland, 2017; pp. 98–119.
50. Verma, A.; Rao, M.; Gupta, A.; Jeberson, W.; Singh, V. A literature review on malware and its analysis. *Int. J. Curr. Res. Rev.* **2013**, *5*, 71.
51. Namanya, A.P.; Cullen, A.; Awan, I.U.; Disso, J.P. The world of Malware: An overview. In Proceedings of the 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, Spain, 6–8 August 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 420–427.
52. Zeltser, L. *Reverse Engineering Malware: Tools and Techniques Hands-On*; SANS Institute: North Bethesda, MD, USA 2007.
53. Qi, M. P2P network-targeted DDoS attacks. In Proceedings of the 2009 Second International Conference on the Applications of Digital Information and Web Technologies, Washington, DC, USA, 13–14 December 2009; IEEE: Piscataway, NJ, USA, 2009, pp. 843–845.
54. R. Aruna, M. Study of P2P Networks Protection Opposing Malicious Attacks. *Int. J. Comput. Sci. Inf. Technol. Res.* **2014**, *2*, 217–223.
55. Jaideep, G.; Battula, B. Detection of DDOS attacks in distributed peer to peer networks. *Int. J. Eng. Technol.* **2018**, *7*, 1051. [CrossRef]
56. Naoumov, N.; Ross, K. Exploiting P2P systems for DDoS attacks. In Proceedings of the 1st International Conference on Scalable Information Systems, Hong Kong, China, 30 May–1 June 2006; p. 47. [CrossRef]
57. B.M. Yakubu, M.I. Khan, A.K.F.J.G.J. Blockchain-based DDoS attack mitigation protocol for device-to-device interaction in smart home. *Digit. Commun. Netw.* **2023**, 1–15. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.