Epidemiological Modelling of Peer-to-Peer Viruses and Pollution

Richard Thommes and Mark Coates

Department of Electrical and Computer Engineering
McGill University
3480 University St
Montreal, QC, Canada H3A 2A7
Email:{rthomm,coates}@tsp.ece.mcgill.ca

***Abstract*— The popularity of peer-to-peer (P2P) networks makes them an attractive target to the creators of viruses and other malicious code. Recently a number of viruses designed specifically to spread via P2P networks have emerged. Pollution has also become increasingly prevalent as copyright holders inject multiple decoy versions in order to impede item distribution. In this paper we derive deterministic epidemiological models for the propagation of a P2P virus through a P2P network and the dissemination of pollution. We report on discrete simulations that provide some verification that the models remain sufficiently accurate despite variations in individual peer conduct to provide insight into the behaviour of the system. The paper examines the steady-state behaviour and illustrates how the models may be used to estimate in a computationally efficient manner how effective object reputation schemes will be in mitigating the impact of viruses and preventing the spread of pollution.**

I. Introduction

Peer-to-peer (P2P) networks have become increasingly vulnerable to malicious behaviour, including the dissemination of *polluted* versions of files and the release of P2P viruses. Early P2P networks such as Napster focussed exclusively on media files, so propagation of viruses was difficult to achieve [1]. Contemporary P2P networks such as Kazaa / Fastrack [2] and eDonkey2000 [3] can be used to disseminate executable files and are hence much more susceptible, particularly as the mainstream adoption of P2P file exchange—the eDonkey2000 network alone typically has over 2 million users connected at any given time [4]—means that a significant fraction of users lack the technical knowledge to detect suspicious files or scan for viruses.

The phenomenon of pollution, the presence of corrupted (or "bad") versions of items (songs, movies or multimedia files) in P2P networks, has become increasingly prevalent. Some of these versions are made available by accident, as users make errors in file generation. But the dominant cause is deliberate dissemination of decoy files, termed *item poisoning* in [5], a technological mechanism employed by copyright holders and their agents to impede the distribution of content. These decoy files have names and metadata matching those of the genuine item, but contain corrupted, unreadable or inappropriate data. Whether accidental or deliberate, pollution has rendered a substantial portion of the files on popular P2P networks unusable.

In this paper we examine the behaviour of viruses and pollution in P2P networks. We adopt an epidemiological approach, developing dynamic models to describe the evolution of infection/pollution. We consider the stochastic nature of the system during our development of the models, but our models are deterministic and focus on the expected behaviour of the system. We illustrate that these deterministic models are sufficiently accurate to capture the behaviour of P2P networks, by comparison with more realistic simulations that model individual peers.

Our initial purpose is to model the impact of malicious code on a P2P network, but a primary motivation is to examine how effective the introduction of mitigation techniques might be. In particular, we focus on *object reputation* schemes (such as Credence [6]) and methods that increase the rate of elimination of infected files. Our model provides an analytical method for determining (at least approximately) how widespread the adoption of such schemes must be, and how effective they must be, in order that specific targets of residual pollution or infection be achieved. We validate these specifications through more accurate simulation of the networks.

The paper is structured as follows. In the remainder of the introduction, we highlight the salient features of P2P networks, viruses and pollution, and discuss related work. Section II presents a model for the expected evolution of a virus in the system. In Section III, we analyze the steady-state behaviour of our P2P virus model. Section IV presents an epidemiological model for the proliferation of pollution. Section V examines the impact of object reputation schemes. Section VI reports on an empirical study of the e-Donkey network, which we conducted to identify suitable parameters for the examination of our models. Section VII reports on discrete-time simulations of the P2P network, which provide a validation that the deterministic models capture the primary characteristics of system evolution despite ignoring the variability in behaviour of individual peers. Finally, Section VIII draws conclusions based on our analysis and results.

A. Peer-to-peer networks, viruses and pollution

This section highlights the key features shared by popular P2P Networks, including Kazaa, eDonkey2000, and Gnutella [7]. Every peer connected to the network has a *shared folder* containing all the files the user wishes to make publicly available for download by others on the network. When a user wants to download a file, he begins by sending out a search request. In response he receives a list of files matching the search criteria. The specific manner in which this list is generated varies among the various P2P networks, but in all cases the query response is the result of the examination of the shared folders of a subset of all peers connected to the network. Once the user elects to download one of the files from the list, his client attempts to set up a connection to a peer sharing the file and begins receiving the file. Depending on the specific network, the client may attempt to simultaneously download different parts of the file from a number of peers in order to expedite the operation. P2P clients typically save new downloaded files in the shared folder – making them immediately available to other users.

A number of worms and viruses that exploit P2P networks have already surfaced. The majority of these behave in a similar fashion. Specifically, when a user downloads a file containing the virus and executes it, a number of new files containing the virus are created and placed in the client's shared directory. Some types of viruses, including Achar [8] and Gotorm [9], generate a fixed list of filenames when executed. More advanced viruses, such as Bare [10] and Krepper [11], randomly pick the list of filenames from a large pool of candidates.

Pollution is a more widespread phenomenon, as indicated by the empirical study performed in [12]. The study indicated that the number of versions of relatively popular items is generally substantial (on the order of tens or hundreds). It was also observed that the pollution level (the fraction of bad versions) for a specific item remained approximately constant over time.

B. Related Work

The advent of mathematical Epidemiology – the field of biology which models how diseases spread in a population – is generally credited to McKendrick and his seminal 1926 paper [13]. Previous work in applying epidemiology to modeling how computer viruses and other malware spreads between machines dates back to the late 1980s/early 1990s [14], [15]. More recently, several authors have utilized epidemiological models to study the spread of worms and e-mail viruses in the Internet [16]–[20].

There have been a number of recent papers which model file propagation in P2P networks [21]–[24]. Dumitriu et al. [5] model the spread of polluted files in P2P networks, and Liang et al. report on an empirical study of pollution in P2P networks in [12]. The behaviour of object reputation mechanisms has been discussed in [6].

Contribution: We believe that our paper is the first to develop a epidemiological model for peer-to-peer viruses. Although these viruses share similarities with Internet worms and e-mail viruses, there are sufficient differences in their spreading mechanics to necessitate the development of a new model. The dynamic pollution model developed in [5] is closely related to our epidemiological pollution model, and produces similar behaviour. Phrasing the model in an epidemiological framework provides an alternative understanding of system behaviour. The deterministic models are reasonably accurate even with substantial variation in individual peer behaviour, and we illustrate how they can be used to estimate in a computationally efficient manner the impact of an object reputation scheme in mitigating P2P viruses and pollution. Conversely, the models can be used to determine how widespread the usage of a reputation scheme must be and how much it must dampen the probability of downloading an infected or polluted file in order to achieve a target level of pollution/infection.

II. P2P Virus Model

The intent of our model is to predict the expected behaviour of a virus which spreads through a P2P network in the form of malicious code embedded in executable files shared by peers. We make the simplifying assumption that all users download files to their shared folder. We are not concerned with the transfer of media files which cannot contain malicious code, and do not model them. Note that we use the term *user* in this paper to refer to a person using a P2P *client* program. The term *peer* is used to collectively refer to a P2P client and the user directing its behaviour.

This model classifies all peers as falling into one of three classes: *Susceptible, Exposed*, or *Infected*:

Susceptible – Peers that are not sharing any infected files, but are at risk of downloading infected files. The number of peers in this category at time t is denoted by $S(t)$.

Exposed – Peers that have downloaded one or more infected files, but have not executed them. The number of peers in this category at time t is denoted by $E(t)$. The Exposed category is included in the model to allow for a delay between download of an infected file and execution.

Infected – Peers that have executed an infected file. Upon execution, a total of of c infected files reside in the peer's shared folder. The number of peers in this category at time t is denoted by $I(t)$.

An Infected client may be detected by the user, who will then proceed to remove all the infected files, thereby returning the state of the peer to Susceptible. At all times, every one of the N peers making up the network falls into one of the three categories. Thus, for all values of t, $N = S(t) + E(t) + I(t)$.

We assume that the total number of uninfected files in the network is fixed at M. The total number of infected files at time t is given by $K(t)$. The expected proportion of infected files in the network, $q(t)$, is therefore $q(t) = \frac{K(t)}{K(t)+M}$.

Event	*Variables Affected*
File downloaded	$q(t)$, $S(t)$, $E(t)$
File executed	$q(t)$, $E(t)$, $I(t)$
Peer recovers	$q(t)$, $I(t)$, $R(t)$

TABLE I

P2P VIRUS MODEL VARIABLES THAT ARE POTENTIALLY AFFECTED BY EACH POSSIBLE EVENT IN THE NETWORK.

When a user downloads a file, we assume the probability of choosing an infected file will be dependent on the prevalence of infected files in the network. The probability will vary to some degree for different peers, according to whether the peer has updated virus-detection software or is aware of the common characteristics of virus files (such files are often much smaller than genuine versions of the item). In our model, we are interested in the average probability of choosing an infected file, and we denote this probability by $h(t)$. In Section III, where we examine steady-state behaviour, we set $h(t) = \alpha q(t)$, for some constant α, to reflect the fact that the probability is closely tied to virus prevalence and to simplify our analysis.

There are three distinct events that may occur in the network which affect one or more of the time-varying variables described above. These events include a peer downloading a file from another, a peer executing a shared file, and an Infected peer recovering. Although individual peers conduct these activities at (potentially very) different rates, we develop our model based on average behaviour. Our simulation results in Section VII indicate that this modelling choice does not produce substantially erroneous behaviour. The average rates at which each of these events occurs are governed by three parameters:

λ_S: Average rate, in files per minute, at which each peer downloads new files (this includes time spent searching and setting up the connection to another peer).

λ_E: Average rate, in files per minute, at which each peer executes shared files. We assume that a peer executes files in the order in which they are downloaded.

λ_R: Average rate, in "recoveries per minute", at which Infected peers recover. A recovery occurs when all infected files are removed, returning the peer state to Susceptible.

A. Model Equations

Table I summarizes which time-varying variables are affected by each of the three events that may occur in the network. The state progression for all peers in our model is $S \rightarrow E \rightarrow I \rightarrow S$.... We now derive the differential equations that govern the evolution of our P2P model.

Rate at which the number of Infected peers changes: When an Infected peer recovers, the number of Infected peers decreases by one. Recoveries occur at rate $\lambda_R I(t)$. When an Exposed peer executes an infected file, the number of Infected peers increases by one. Since files are executed in order of download, the file executed by an Exposed peer will always be the infected file which it had downloaded to become Exposed. This occurs at a rate of $\lambda_E E(t)$. Therefore,

$$\frac{dI(t)}{dt} = -\lambda_R I(t) + \lambda_E E(t) \tag{1}$$

Rate at which the number of Exposed peers changes: The rate at which the number of Exposed peers decreases due to infection is given by the negative of the second term in (1). The rate at which previously Susceptible peers become Exposed is dependent on the aggregate rate at which they download files, $\lambda_S S(t)$, multiplied by the probability that a downloaded file is infected, $h(t)$. The overall rate is therefore:

$$\frac{dE(t)}{dt} = -\lambda_E E(t) + \lambda_S S(t) h(t) \tag{2}$$

Rate at which the number of Susceptible peers changes: Since N is fixed, it always holds that $\frac{dS(t)}{dt} + \frac{dE(t)}{dt} + \frac{dI(t)}{dt} = 0$. Therefore, $\frac{dS(t)}{dt}$ is the negative sum of (1) and (2):

$$\frac{dS(t)}{dt} = -\lambda_S S(t) h(t) + \lambda_R I(t) \tag{3}$$

Rate at which the number of infected files in the network changes: There are three events which result in a change in the number of infected files in the network: a peer downloads an infected file, an Exposed peer becomes Infected, and an Infected peer recovers. We assume that all downloaded files are executed, and that a peer does not download any additional files prior to executing the most recently downloaded file.

Peers cannot share more than one copy of a file with the same name. If the number of unique infected filenames is limited to c, only Susceptible peers can download infected files. Exposed peers do not download any additional files before becoming Infected, and Infected peers are sharing all c possible infected files. Thus, the rate of change due to downloads is $S(t)\lambda_S h(t)$.

An Exposed peer always has one infected file before becoming Infected, meaning in all cases $c-1$ new infected files are created when an Exposed peer becomes Infected. The rate of change is thus $E(t)\lambda_S(c-1)$.

An Infected peer will always share c files, so a recovery results in a reduction of c infected files. The rate is therefore $-I(t)\lambda_R c$. The overall rate of change of K is therefore:

$$\frac{dK(t)}{dt} = S(t)\lambda_S h(t) + E(t)\lambda_E(c-1) - I(t)\lambda_R c \tag{4}$$

We note that if the names of generated files are chosen from a pool of names much larger than c, Infected peers can continue to download infected files and the above equation does not hold. The model and analysis in this case becomes more involved. See [25] for a discussion on this and other variations of the model, including cases where not all downloaded files are executed and where multiple downloads are possible prior to execution.

B. Model Extensions

1) Modeling On-line/Off-line Behaviour: In a real P2P network, individual peers are only on-line for limited durations. In order to capture this behavior, we present an extension of our model that includes both on-line and off-line users. Each of the three variables specifying how many peers are in each category – S, E, I – is partitioned into two variables to account for how many peers in the category are on and off-line. So, for instance, $I(t) = I_N(t) + I_F(t)$, where $I_N(t)$ is the number of Infected peers on-line, and $I_F(t)$ is the number of Infected peers offline. Peers that are off-line go on-line at a certain rate λ_N , and on-line peers go off-line at rate λ_F . The differential equation governing the change in the number of on-line Infected peers at time t is:

$$\frac{dI_N(t)}{dt} = I_F(t)\lambda_N - I_N(t)\lambda_F \tag{5}$$

The equations governing the rates of change in $S_N(t)$ and $E_N(t)$ are analogous. We assume here that peers go on and off-line at the same rate regardless of their state. It would also be simple to expand the model to include different rates for each state.

To complete the specification of the extended model, all the previously defined differential equations are changed as follows: every instance of $S(t)$, $E(t)$, and $I(t)$ is replaced, respectively, by $S_N(t)$, $E_N(t)$, and $I_N(t)$.

2) Modeling Peers that Remain Infected: One can argue that a certain proportion of P2P users, when their client becomes Infected, will never detect that this has occurred and not take any action to remedy this problem. In order to include this behaviour in our model, we classify all peers as "aware" or "oblivious". Aware peers behave as those in our basic model described in II-A, while oblivious peers progress $S \rightarrow I$ and then remain Infected. The number of peers in each group is fixed: $N = N_A + N_O$ where N_A is the number of aware peers, and N_O is the number of oblivious peers.

As in Section II-B.1, the number of peers falling into each of the four categories at time t is partitioned into two groups; in this case the number of aware users in category X at time t where $X \in \{S, E, I\}$, is denoted by $X_A(t)$ and the number of oblivious users in each category is denoted by $X_O(t)$. The behaviour of aware users is determined by equations (1), (2), and (3), with $X_A(t)$ replacing $X(t)$ for all $X \in \{S, E, I\}$. Oblivious users are governed by (1), (2), and (3), with $X_0(t)$ replacing $X(t)$, and λ_R set to zero (reflecting the fact that oblivious peers never recover). Finally, $\frac{dK(t)}{dt}$ is governed by a modified version of (4), with $S(t)$ replaced by $S_A(t)+S_O(t)$, $E(t)$ replaced by $E_A(t)+E_O(t)$, and $I(t)$ replaced by $I_A(t)$.

III. Analysis - Stability Results

If the P2P network reaches a steady-state equilibrium by some time $t = T$, then $\frac{dE(T)}{dt} = \frac{dI(T)}{dt} = \frac{dS(T)}{dt} = 0$. In this section, we assume that the probability of downloading an infected file is a function of the proportion of infected files, i.e., $h(t) = f(q(t))$. Defining $\tilde{E}$, $\tilde{I}$, $\tilde{S}$, as the steady-state values of, respectively, $E(t)$, $I(t)$, and $S(t)$, Equation (1) implies that:

$$\tilde{I} = \tilde{E}\frac{\lambda_E}{\lambda_R} \tag{6}$$

If we define τ and μ as, respectively, the expected number of infected files each Exposed and Infected peer is sharing in steady-state, then $\tilde{q}$, the proportion of infected files in steady-state may be expressed as:

$$\tilde{q} = \frac{\tilde{E}\tau + \tilde{I}\mu}{M + \tilde{E}\tau + \tilde{I}\mu} \tag{7}$$

Substituting (6) into (7) provides:

$$\tilde{q} = \frac{\tilde{E}(\tau\lambda_R + \mu\lambda_E)}{M\lambda_R + \tilde{E}(\tau\lambda_R + \mu\lambda_E)} \tag{8}$$

If $f(\tilde{q}) > 0$, equation (2) implies that, in steady state:

$$\tilde{S} = \tilde{E}\frac{\lambda_E}{\lambda_S f(\tilde{q})} \tag{9}$$

Since $\tilde{S} = N - \tilde{I} - \tilde{E}$, equation (6) can be utilized to express N as:

$$\tilde{S} = N - \tilde{E}(1 + \frac{\lambda_E}{\lambda_R}) \tag{10}$$

If $h(t)$ is proportional to $q(t)$, $h(t) = \alpha q(t)$, we can obtain a closed-form expression for $\tilde{E}$ by substituting (8) into (9), equating with (10), and solving for $\tilde{E}$:

$$\tilde{E} = \frac{\lambda_R\alpha(N\lambda_S(\mu\lambda_E + \tau\lambda_R) - M\lambda_E\lambda_R)}{(\tau\lambda_R + \mu\lambda_E)(\lambda_S\alpha(\lambda_R + \lambda_E) + \lambda_E\lambda_R)} \; ; \tilde{q} > 0 \tag{11}$$

The expression for $\tilde{I}$ follows trivially from (11) and (6):

$$\tilde{I} = \frac{\lambda_E\alpha(N\lambda_S(\mu\lambda_E + \tau\lambda_R) - M\lambda_E\lambda_R)}{(\tau\lambda_R + \mu\lambda_E)(\lambda_S\alpha(\lambda_R + \lambda_E) + \lambda_E\lambda_R)} \; ; \tilde{q} > 0 \tag{12}$$

If $\tilde{q} = 0$, it follows from (7) that $\tilde{E} = \tilde{I} = 0$. It is of interest to consider Equation (12) as it approaches 0. In the limiting case, approached from above, we have the equality

$$N\lambda_S(\mu\lambda_E + \tau\lambda_R) = M\lambda_E\lambda_R \tag{13}$$

Since we assume that all downloaded files are eventually executed, it follows that it is reasonable to equate the rates of download and execution, $\lambda_E = \lambda_S$. Under this assumption, (13) provides the following minimum average recover rate, λ_R^{min} in order for all infected files to eventually be removed from a P2P network:

$$\lambda_R^{min} = \frac{N\mu\lambda_E}{M - N\tau} \; ; M > N\tau \tag{14}$$

This equation indicates that, if $h(t) = \alpha q(t)$, then λ_R^{min} is a linearly increasing function of λ_E.

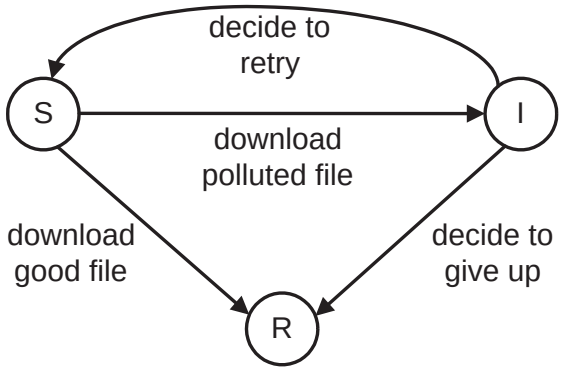


Fig. 1. The transition diagram for peers indicating the actions that trigger movement between the three classes of susceptible (S), infected (I) and recovered (R)

IV. P2P Pollution Model

We assume that M_i peers are interested in item i, and that there are a multitude of versions of the item, classified as "good" or "bad". Initially the P2P network is seeded with $N_g(0)$ good files and $N_b(0)$ bad files. The peers who provided these seed files do not number among the M_i peers we consider in our model. We model the peers as belonging to three classes: *Susceptible*, *Infected*, and *Recovered*. $S(t)$ is the number of susceptible peers at time t; this class includes all peers who will attempt to download another version of the file in the future. Initially $S(0) = M_i$, as all interested peers are susceptible. $I(0) = 0$ and $R(0) = 0$, because no files have been downloaded from the seeds.

A peer transitions between the three states as depicted in the transition diagram in Figure 1. Each peer is susceptible when it intends to download a file. When a susceptible peer downloads a file, it joins the Infected class if the file is bad and the Recovered class if the file is good. A peer may leave the Infected class by testing the downloaded file and electing to retry at some stage in the future. In this case, the peer rejoins the Susceptible class. Alternatively, an infected peer may decide to give up and join the Recovered class, despite not being successful in acquiring a good version of the item. A peer may dwell in the infected state for some period of time before choosing to give up or to retry. This represents the period of time before an infected peer tests a downloaded file.

Eventually all peers will belong to the Recovered class. We label this class "recovered" primarily to highlight the parallels with standard epidemiological models. In our model the distinguishing feature of a recovered peer is that it is no longer actively seeking the item of interest. Note that in our model, any susceptible or infected peer may be sharing none or several polluted files, but cannot be sharing a good file. A recovered peer may share at most one good file and may share several polluted files.

The number of good shared versions of the item varies over time, as does the number of bad. When a peer transitions from the susceptible to recovered state by downloading a good version, it shares the file with probability p_{sg}. When a peer transitions from the susceptible to infected state by downloading a bad file, it shares the file with probability p_{sb}. When a peer transitions from the infected to susceptible state or recovered state, it removes the polluted file with probability p_{db}. We model the probability of downloading a polluted file at time t, $p_b(t)$, as being equal to the fraction of polluted files. This probability is the same for a peer irrespective of how many times it has been infected. This is a reasonable approximation because the number of versions of an item is anticipated to be much larger than the number of re-tries.

We model the expected behaviour of a large group of peers. At time t, a fraction of the susceptible peers λ_s download a file. This is effectively the download rate. A fraction λ_r of the infected peers decide to retry and hence rejoin the susceptible pool. A fraction λ_x of the infected peers choose to give up and enter the recovered state. We make the simplifying assumption that the download rate, and the rates of trying again and giving up (λ_r and λ_x) do not vary over time. A constant value of λ_s produces the approximately exponential decay in the number of downloads of an item as time elapses and its popularity declines. It is reasonable to assume that the variation of the rates of trying again or giving up do not change substantially over time.

With these modelling choices, we arrive at the following set of equations that describe the evolution of pollution in the system.

$$p_b(t) = \frac{N_b(t)}{N_b(t) + N_g(t)} \tag{15}$$

$$\frac{dS(t)}{dt} = -\lambda_s S(t) + \lambda_r I(t) \tag{16}$$

$$\frac{dI(t)}{dt} = p_b(t)\,\lambda_s I(t) - (\lambda_r + \lambda_x) I(t) \tag{17}$$

$$\frac{dR(t)}{dt} = (1 - p_b(t))\lambda_s S(t) + \lambda_x I(t) \tag{18}$$

$$\frac{dN_b(t)}{dt} = \lambda_s\, p_b(t)\, p_{sb}\, S(t) - (\lambda_r + \lambda_x)\, p_{db}\, p_{sb} I(t) \tag{19}$$

$$\frac{dN_g(t)}{dt} = \lambda_s (1 - p_b(t))\, p_{sg}\, S(t) \tag{20}$$

As with the P2P virus model, these equations are derived under the assumption that all peers have common behaviour; variability in individual behaviour means that this will not be a completely accurate model of the system. In addition, the model does not address any notion of memory in user behaviour; it is probable that a peer's downloading behaviour would change substantially if it has already received several bad versions of an item. In simulations in Section VII, we account for variability in peer behaviour and a limited notion of memory; our results indicate that the deterministic model described above, despite its limitations and assumptions, provides a good indication of the evolution of the extent of pollution in the P2P network (for a specific item).

V. The Impact of Object Reputation Schemes

The possibility of downloading an infected or polluted file may be reduced through the use of an *object reputation* scheme which allows P2P users to rate individual files and share this information with others in the network. The standard Kazaa client [2] includes such a feature, allowing users to assign one of four possible rankings to each file. However, this simplistic implementation has been ineffective in combatting the number of polluted files in the network [26]. A recently introduced object-reputation scheme for the Gnutella network named Credence [6] appears promising because of its robustness in the face of malicious peers which intentionally give high ratings to polluted or Infected files. In this section we model the effect that an effective object-reputation scheme such as Credence has on virus propagation in a P2P network.

A. Effect on P2P Virus Propagation

As in Section II-B.2, peers are divided into two groups, "smart" peers which utilize an object-reputation system, and "regular" peers which do not. The number of regular peers falling in a category X at time t, is denoted by $X_R(t)$ and the number of smart users in each category is denoted by $X_S(t)$. Regular peer behaviour is governed by equations (1), (2), and (3). Smart peer behaviour is determined by equation (1) and modified versions of equations (2) and (3) with $h(t)$ replaced by $g(t)$. In order to reflect the fact that smart users are less likely to download infected files, we require that $g(t) \leq h(t)\ \forall t$. In the case of a perfect object-reputation system, in which smart peers never download infected files, $g(t) = 0\ \forall t$ and hence $S_S(t) = N_S\ \forall t$. Finally, equation (4) is replaced by

$$\frac{dK(t)}{dt} = S_R(t)\lambda_S h_t + S_S(t)\lambda_S g_t + (E_R(t) + E_S(t))\lambda_E(c-1) - (I_R(t) + I_S(t))\lambda_R c \tag{21}$$

B. Effect on Pollution Dissemination

We model the effect on pollution dissemination in a similar fashion, decomposing the set of interest peers into the two groups of "smart" and "regular" peers. The object reputation scheme is assumed to reduce the probability of downloading a bad version of a file by a fixed proportion. Smart peers now download a bad version with probability

$$p_{b,S}(t) = \frac{\beta N_b(t)}{N_b(t) + N_g(t)} \tag{22}$$

for some constant $\beta < 1$. Regular peers download bad versions with the same probability as before (proportional to the extent of pollution). The modified epidemiological model now keeps track of the number of smart and regular peers in each class and can hence determine the rates of change of the number of good and bad files in the network. We have:

$$\frac{dN_b(t)}{dt} = \lambda_s\, p_{sb}(p_{b,S}(t)\, S_S(t) + p_{b,R}(t)\, S_R(t)) - (\lambda_r + \lambda_x)\, p_{db}\, p_{sb}\, I(t) \tag{23}$$

$$\frac{dN_g(t)}{dt} = \lambda_s\, p_{sg}((1 - p_{b,S}(t))\, S_S(t) + (1 - p_{b,R}(t))S_R(t)). \tag{24}$$

VI. P2P Measurements

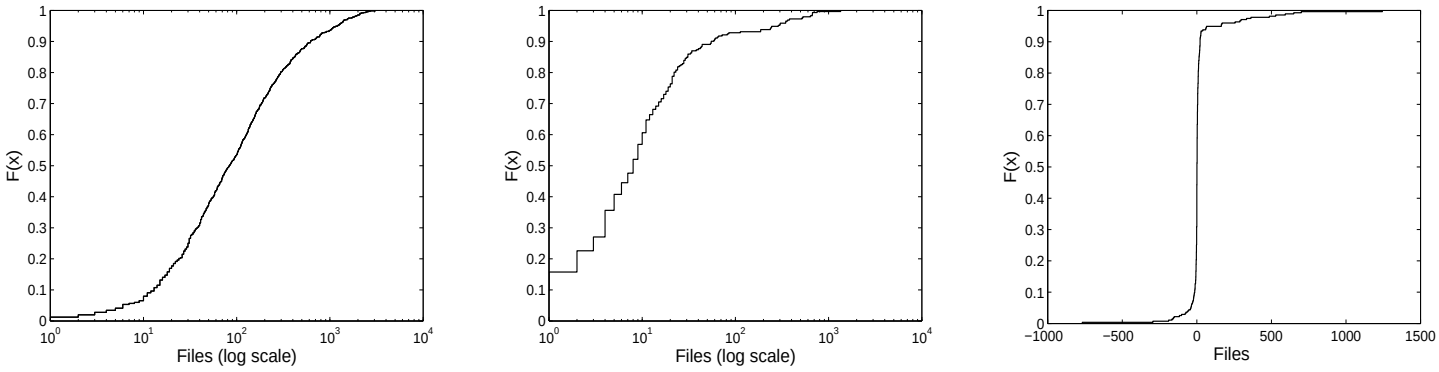
In order to choose a realistic value of λ_S for simulation experiments with our model, we sought to acquire appropriate measurement data from an actual P2P network. A number of previous empirical studies have explored the behaviour of the Gnutella Network [27]–[30] and the Kazaa Network [26], [30], and the eDonkey network [31]. The statistics presented have included the number of files shared by peers, latency between peers, the amount of time spent on and off-line, the degree of peer connectivity, and mean bandwidth usage. However, we are not aware of any previous work directly analyzing the rate at which peers download files.

We chose to conduct our measurements on the eDonkey2000 network because of its popularity and the apparently limited amount of research conducted on the network. BayTSP [32], a company which monitors Internet file-trading, indicates that as of September, 2004 the eDonkey2000 network has, on average, the most users of any P2P network [33].

The eDonkey2000 network is comprised of a number of servers [4] to which a peer can connect. Each server keeps a list of all the files shared by connected peers, and uses this information to respond to keyword-based search queries. The search results returned by the server include a 16-byte MD4 hash [34] value for each file in order to uniquely identify it. When the user elects to download a specific file, his client sends the hash value of the desired file to the server, and the server responds with a list of IP addresses and ports of peers sharing the file.

Our experiment consisted of two phases. In the first part, we collected a list of eDonkey2000 peer IP addresses/ports. We achieved this by first conducting searches for keywords likely to return a significant number of results, for example: ".exe", and ".iso", and then initiating the download of files shared by a large number of peers. Next, we made use of the Ethereal network protocol analyzer [35] to capture and analyze the packets returned by the server containing the peer IP addresses. We initiated the download of approximately 500 files to harvest over 20 thousand peer addresses. For the next phase of the experiment, we developed a scanner program which attempts to connect to every peer and retrieve its list of shared files. We made use of previous work carried out to reverse-engineer the eDonkey2000 protocol [36], and conducted further analysis using Ethereal.

Users of eDonkey2000 have the option of configuring their clients to block requests by other peers to view their list of shared files. Our work was complicated by the fact that approximately 95% of peers to which we attempted to connect did not permit viewing of their shared files. There are



(a) CDF of No. files shared per peer.

(b) CDF of No. downloads per peer per 48 hour interval

(c) CDF of net change No. shared files per peer per 48 hour interval.

Fig. 2. Empirical CDFs based on eDonkey measurement data.

two obvious factors that contribute to this high percentage: eMule [37], the most popular eDonkey2000 client, has the blocking option enabled by default, and the advent of RIAA (Recording Industry Association of America) lawsuits directed against P2P users [38] based on the scanning of shared directories has likely motivated many users to actively disallow the viewing of their files. Nevertheless, we managed to connect to one thousand peers and retrieve their lists of shared files. We repeated this procedure three more times, in 48-hour intervals. Each scan required approximately two hours to carry out. In order to deduce the rate at which users were downloading files, we tracked the addition of any new shared files every time the scanner connected to a peer. We assume that any new file is the result of a download. Admittedly, the possibility exists that a new shared file was not downloaded, but instead added to the shared directory by the user from a source outside the eDonkey2000 network. However, we are unable to distinguish such files and therefore our calculated download rate may be a slight over-estimate. Table II provides the results of our measurements. The overall average download rate is 37.7 files per 48-hour period. Figure 2 provides the empirical cumulative density functions (CDFs) of the number of files shared per peer, the number of downloads per peer per 48 hour interval, and the net change in number of files each peer changes per 48 hour interval. All three plots suggest heavy-tailed distributions, indicating that there are a small percentage of "power-peers", which are much more active and share many more files. This phenomenon has been observed in other empirical studies conducted on P2P networks [30], [31].

We calculated the rate at which peers removed files from their shared folder, by counting all files peers had made available during a given run of our scanner program which were no longer present during a subsequent scan. The average removal rate is 29.1. Although this does not entirely validate our Section II assumption of a zero net increase in the total number of files, it indicates that files are removed from the network at a similar rate to which new ones are downloaded. Furthermore, a website [4] tracking eDonkey2000 server statistics over one-month intervals indicates that, while there are significant daily fluctuations in the number of files available, the month-long trend is fairly constant.

Interval	*% of Peers with 0 new Files*	*% of Peers with 1-10 new Files*	*% of Peers with 10-100 new Files*	*% of Peers with $\geq$ 101 new Files*	*Average Download Rate (Files/ 48 hrs.)*
1	11	50	33	6	41.2
2	12	47	33	8	35.8
3	7	39	48	6	36.0

TABLE II
OBSERVED EDONKEY2000 PEER DOWNLOAD BEHAVIOUR OVER THREE DISJOINT 48-HOUR INTERVALS.

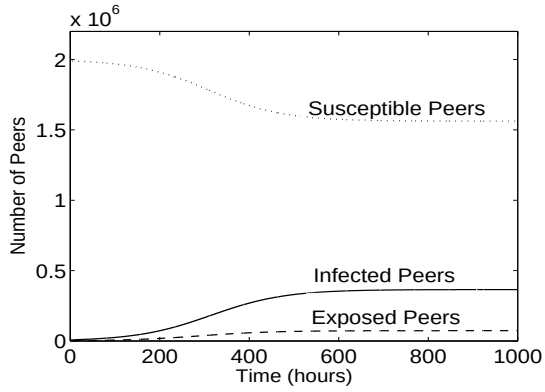
As stated in Section II, we are only concerned about modeling executable files in P2P networks. To estimate the proportion of these files in the eDonkey2000 network, we analyzed the aggregate list of approximately 230 thousand files initially shared by the one thousand peers we tracked. From this list, we removed all files with extensions known to indicate a media file, e.g. ".mp3" and ".avi".

This left just over 55 thousand files that were likely to be executable. Therefore, we estimate that the proportion of files on the eDonkey2000 network that can potentially contain malicious code lies at 24%. We note that this value may be a slight over-estimate, due to the fact that some of the shared files were compressed (".zip" or ".rar"), and therefore we could not identify them as executable with total certainty.

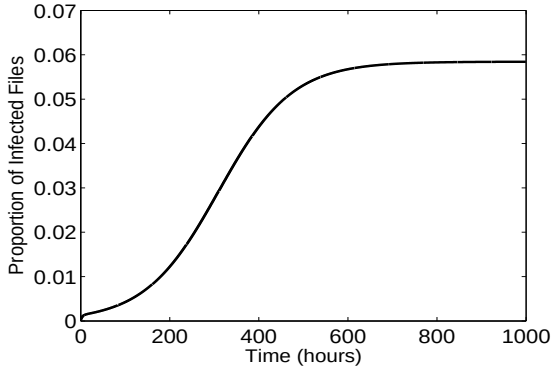
VII. SIMULATION RESULTS

A. Virus Model Behaviour

In this section we provide some examples of virus behaviour in a P2P network as predicted by our model. Figure 3 illustrates how the number of peers falling into each of the three categories evolve over time, and eventually reach a steady state. In this case, $\lambda_E = \lambda_S = 3.47 \times 10^{-3}$ files per minute,



(a) The number of peers in each group



(b) The proportion of infected files

Fig. 3. Example of the dynamic behaviour of a P2P network exposed to a virus (with model parameters set to the values described in Section VII-A). The network reaches steady-state after about 600 hours, at which point approximately twenty percent of the peers are infected.

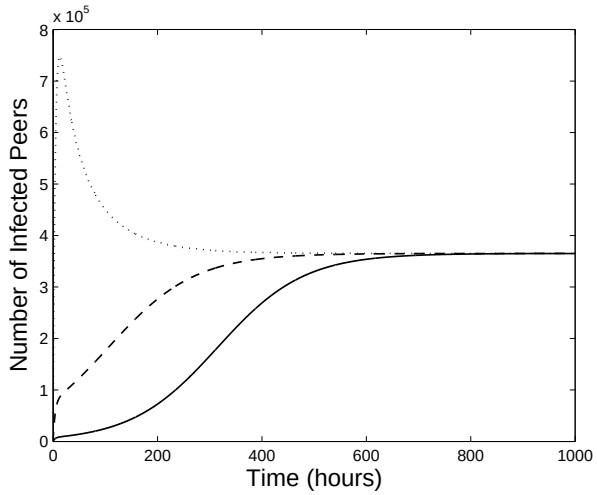


Fig. 4. The effect of the initial infection on the evolution of the number of infected peers. The solid line corresponds to 10 000 infected files initially in the network, the dashed line: 100 000 initial infected files, the dotted line: 1000000 initial infected files.

which corresponds to 5 downloads/executions per day. The average time for a peer to recover is 24 hours, meaning λ_R is 6.94×10^{-4}. The number of peers, N, is 2 million and there are 60 million clean files M. This example makes use of the model in which the number of unique infected files is limited to c, and c is 10. Finally, $h(t) = 0.5q(t)$. Initially, there are 10 000 Exposed peers, each sharing one infected file.

In Figure 4 we examine the effect of varying the initial extent of infection on the evolution of the number of infected peers in the network. For high initial infection (1 million files), there is an initial overshoot in the number of infected peers beyond the steady state. The medium initial infection case converges most quickly to the steady state value, since, out of the three cases, the number of initially infected peers is closest to the eventual steady state value. After about 700 hours, the three networks reach the same steady-state. This is also the behaviour implicitly predicted by equation 12, since it is independent of any initial condition (as long as at least

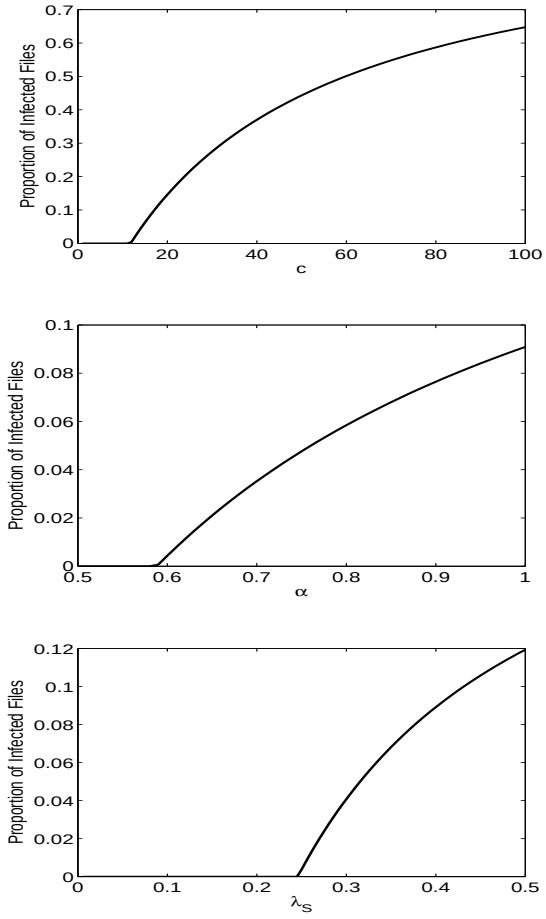


Fig. 5. The effect of varying model parameters on the analytical steady-state proportion of infected files. (a) The effect of varying c, the number of virus files created upon infection. (b) The effect of varying α, the constant determining the probability of downloading an infected file. (c) The effect of varying the download rate λ_S.

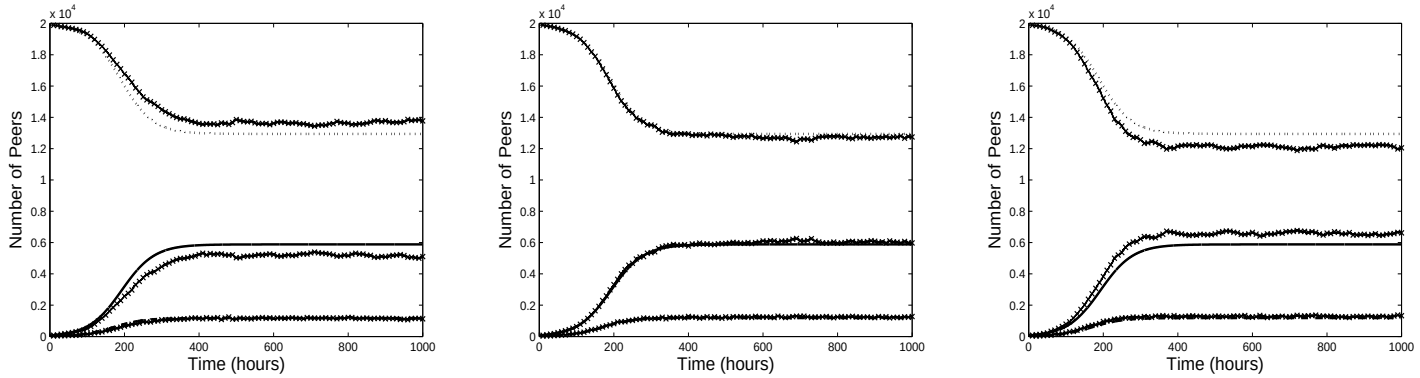


Fig. 6. The impact of variability in individual peer download rates. The solid, dotted and dashed lines show the predicted behaviour according to the dynamic model of infected, susceptible and exposed peers, respectively. The hashed lines show the results achieved in discrete-time simulations. (a) Download rate drawn from a uniform distribution; (b) download rate drawn from a normal distribution; (c) interval between downloads drawn from a normal distribution.

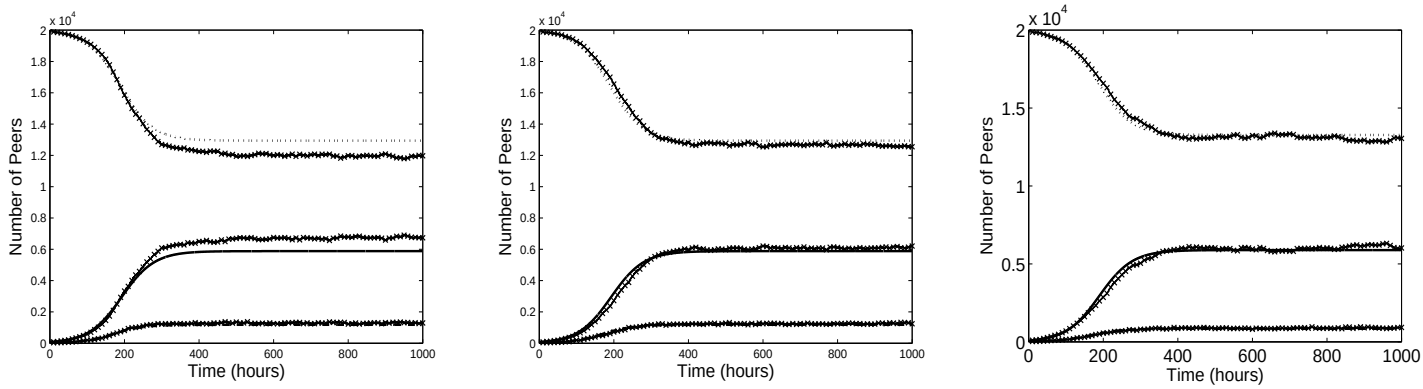


Fig. 7. The impact of variability in individual peer recovery rates. The solid, dotted and dashed lines show the predicted behaviour according to the dynamic model of infected, susceptible and exposed peers, respectively. The hashed lines show the results achieved in discrete-time simulations. (a) Recovery rate drawn from a normal distribution; (b) interval between recoveries drawn from a normal distribution; and (c) Recovery rate, download rate and susceptibility to infection (α) drawn from normal distributions.

one infected file initially exists in the network).

Figure 5 examines how the steady-state proportion of infected files is affected as model parameters are varied. The panels in the figure display the effect of changing (i) c, the number of virus files inserted in the shared directory upon infection; (ii) α, the constant that governs the probability of downloading an infected file; and (iii) λ_S, the download rate of peers in the network. These plots indicate that an increasing α and download rate have a limited effect on the infection level of the network, whereas an increase in the number of files created by a virus can significantly raise the steady-state infection of the network. However, in a practical setting, the more new files a virus creates, the more likely a user is to notice them and delete them. Thus, in reality, the recovery rate would likely be an increasing function of c and the high level of infection for viruses creating 50 or more new files upon execution would be unlikely to occur.

B. Virus simulations with varying peer behaviour

The propagation of a virus in a P2P network predicted by our model is based only on the expected values of peer recovery rates: λ_R, peer download rate: λ_S and peer execution network: λ_E. Realistically, one may expect these values to differ significantly among peers. Since our equations do not incorporate the notion of a random distribution of these parameters for each peer, we are essentially modeling a P2P network in which all peers take on the same deterministic parameter values. Therefore, it is of interest to consider how closely the results predicted by our model mirror those which would be seen in a P2P network in which individual peer parameters are randomly distributed. To this end, we present a number of discrete-time simulation results for a peer-to-peer network in which individual recovery and download/execution rates are chosen according to several different probability distributions.

All figures illustrate the evolution of the number of Infected, Exposed, and Susceptible peers over time. The non-hashed lines are the values predicted by our model, and the hashed lines represent the values obtained via our simulations. We consider 20 000 users sharing 600 000 clean files. Parameters not explicitly mentioned below are set to the same values as in Section VII-A. In Figure 6(a), the download/execution rate is uniformally distributed about the mean value of $\frac{5}{24}$ files per

day, with individual rates varying from 0 to $\frac{10}{24}$. Figure 6(b) illustrates the case where the download rate is normally distributed with mean $\frac{5}{24}$ and standard deviation 0.05. Finally, in Figure 6(c), the average length of time between downloads is normally distributed, with mean $\frac{24}{5}$ and standard deviation 5. In figure 7(a) the recovery rate is normally distributed with mean $1/24$ recoveries per day, and standard deviation 0.1. In figure 7(b) the length of the interval between recoveries is normally distributed with mean 24 and standard deviation 5. Finally, in Figure 7(c) both the download and recovery intervals are normally distributed. The key observation from these figures is that the simulation results converge to steady-state values, and that these values are within 10% of the values predicted by our model. Given these facts, we assert that our model provides a good approximation of a P2P network in which individual peer behaviour may vary significantly.

C. Pollution model behaviour and simulations

In order to verify our pollution model, we conducted a discrete-time simulation of a P2P network with polluted files, and compared it to the results predicted by our model. As with our other simulations, we used exponentially-distributed delays between the various events governed by rate parameters. We set $p_{sg} = p_{sb} = p_{db} = 0.3$, $N_g(0) = 10$, $N_b(0) = 100$, $M_i = 20000$, $\lambda_S = \frac{5}{24}$, $\lambda_X = \frac{1}{24}$, $\lambda_r = \frac{2}{24}$. Figure 8(a) shows the number of Susceptible, Infected and Recovered peers versus time for both the simulation and the model. Figure 8(b) shows how p_b varies with time and reaches a steady state. The model and the simulation track each other well, with the steady-state p_b varying by less than 10%.

In Figure 8(c), we examine the impact that the initial number of seeded polluted files has on the steady-state value of p_b. All other parameters are as described above. This plot indicates that the initial number of polluted files seeded will indeed have a significant effect on the long term pollution level of the network.

D. Impact of Object Reputation Schemes on P2P virus propagation

We now report on simulation and model results for the impact of an object reputation scheme such as Credence on the evolution of P2P viruses. Figure 9(a) illustrates how the steady-state proportion of infected files changes as the effectiveness of Credence (as reflected by β, the factor by which the probability of download of an infected file is reduced) increases. Figure 9(b) depicts the reduction in residual infection as the number of peers using Credence increases. These results are obtained for the model parameters described in Section VII-A. The results indicate that if Credence reduces the probability of downloading an infected file by a factor of 0.7 and fifty percent of the peers use Credence, then the residual infection is halved.

Figure 10 compares the behaviour of the deterministic model with a discrete time simulation of the propagation of a virus in a P2P network consisting of 20 000 peers. Fifty percent of the users employ Credence and it has an

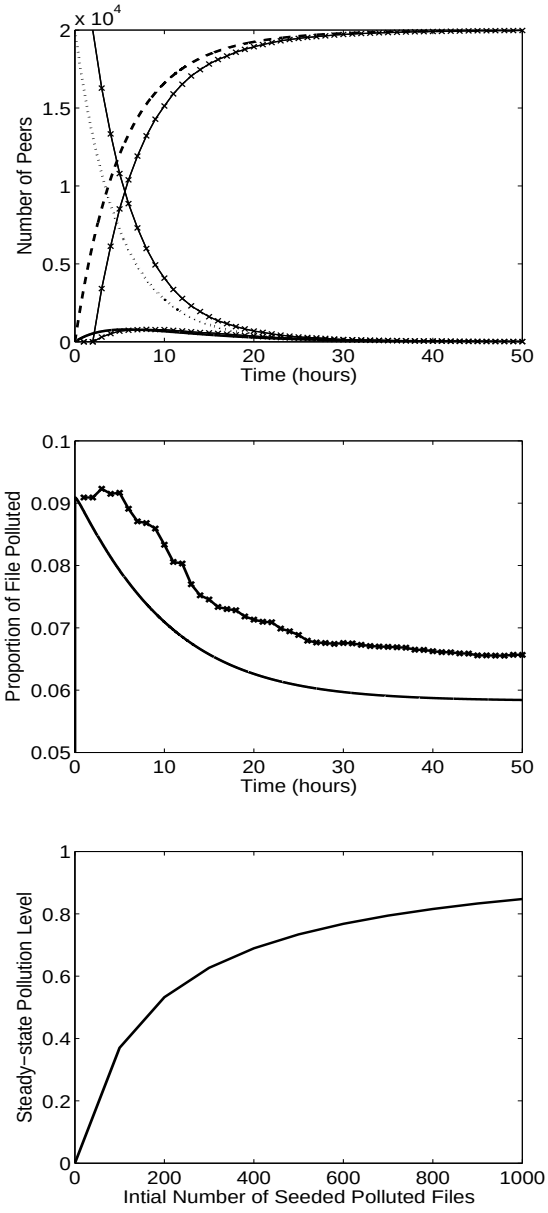


Fig. 8. Examining the behaviour of the pollution model. Hashed lines are simulation results. (a) The evolution of susceptible (dotted), infected (solid) and recovered (dashed) peers. (b) The percentage of polluted files versus time. (c) The steady-state percentage of polluted files as a function of the number of initially "bad" files (with 100 good files).

effectiveness of $\beta = 0.7$. The figure illustrates that there is a good match between the expected behaviour and that of the simulated system.

VIII. Conclusion

We have presented a deterministic epidemiological model of how a P2P virus spreads infection in a P2P network, and derived expressions for the steady-state behaviour in the case where the probability of a peer downloading an infected file is proportional to the prevalence of infection. We have also described an equivalent model for the evolution of pollution

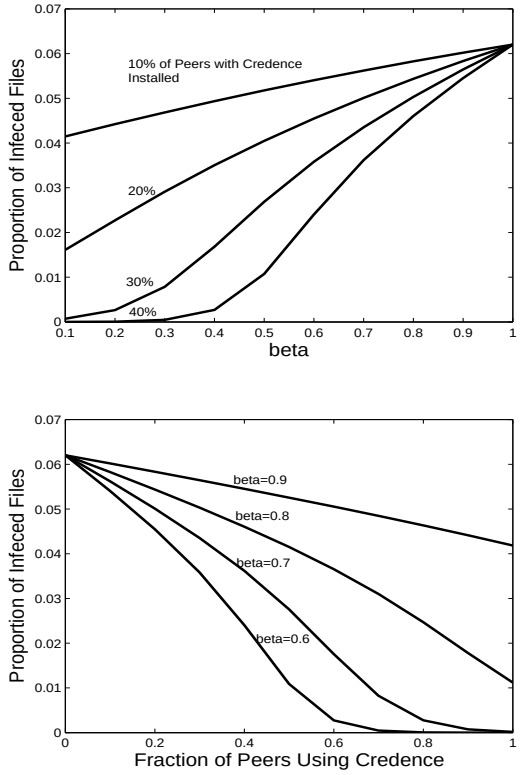


Fig. 9. The impact of using an object reputation scheme such as Credence on the residual proportion of infected files. The proportion of infected files as (a) a function of β, the parameter determining the effectiveness of Credence, and (b) a function of the fraction of peers using Credence.

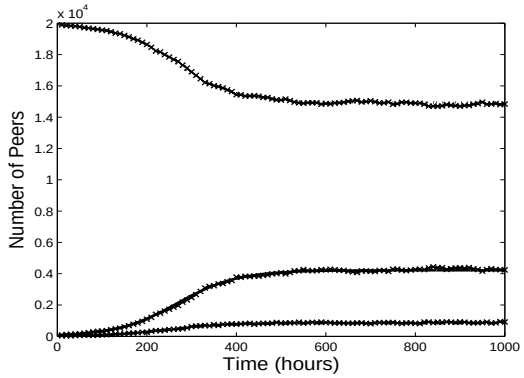


Fig. 10. A comparison between the predicted behaviour according to the epidemiological model and a discrete time simulation. The hashed lines are the results of the simulator (number of susceptible, infected and exposed peers from top to bottom). These lines cover the predicted results for most of the display.

in a P2P network. Discrete-time simulations with varying individual peer behaviour indicates that the models are sufficiently accurate to provide insight into system dynamics despite being based on average behaviour. Our goal in developing these models was to provide a basis for understanding virus and pollution evolution, but also to construct a computationally efficient platform for estimating the efficacy of object reputation systems. In future work we will perform more extensive validation of the models using further empirical measurements of P2P networks and more accurate simulators of P2P networks that fully incorporate the subtleties of object reputation schemes.

ACKNOWLEDGMENT

The research supported in this paper was supported by an NSERC Discovery grant and an NSERC Strategic grant.

REFERENCES

[1] F-Secure, "F-secure hoax information pages: Mp3 virus," `http://www.f-secure.com/hoaxes/mp3.shtml`, 1998.
[2] "Kazaa," `http://www.kazaa.com`.
[3] "Edonkey2000," `http://www.edonkey2000.com`.
[4] "eDonkey2000 server list," `http://ocbmaurice.no-ip.org/slist/serverlist.html`.
[5] D. Dumitriu, E. Knightly, A. Kuzmanovic, I. Stoica, and W. Zwaenepoel, "Denial-of-service resilience in peer-to-peer file-sharing systems," in *Proc. ACM Sigmetrics*, Banff, Canada, June 2005.
[6] K. Walsh and E.G. Sirer, "Thwarting p2p pollution using object reputation," Tech. Rep. Computer Science Department Technical Report TR2005-1980, Cornell University, Feb 2005.
[7] "Gnutella protocol development," `http://rfc-gnutella.sourceforge.net/`.
[8] Viruslist.com, "P2p-worm.win32.achar.a," `http://www.viruslist.com/en/viruses/encyclopedia?virusid=23893`, May 2003.
[9] Symantec, "W32.hllw.gotorm," `http://securityresponse.symantec.com/avcenter/venc/data/w32.hllw.gotorm.html`, August 2003.
[10] Viruscan, "W32.hllw.gotorm," `http://www.virus-scan-software.com/latest-virus-software/latest-viruses/w32bare-worm.shtml`.
[11] Sophos, "Sophos virus analysis: Troj/krepper-g," `http://www.sophos.com/virusinfo/analyses/trojkrepperg.html`, July 2004.
[12] J. Liang, R. Kumar, Y. Xi, and K. W. Ross, "Pollution in P2P file sharing systems," in *Proc. IEEE Infocom*, Miami, FL, Mar. 2005.
[13] A.G. McKendrick, "Applications of mathematics to medical problems," *Proc. Edinb. Math. Soc.*, vol. 44, pp. 98–130, 1926.
[14] W.H. Murray, "The application of epidemiology to computer viruses," *Computers and Security*, vol. 7, pp. 130–150, 1988.
[15] J.O. Kephart and S.R. White, "Directed-graph epidemiological models of computer viruses," in *Proc. IEEE Symp. Security and Privacy*, Oakland, CA, May 1991.
[16] C.C. Zou, W. Gong, and D. Towsley, "Code red worm propagation modeling and analysis," in *Proc. ACM Conf. Computer and Comm. Soc.*, Washington DC, Nov. 2002.
[17] M. Liljenstam, Y. Yuan, B. Premore, and D. Nicol, "Email worm modeling and defense," in *Proc. IEEE Int. Symp. MASCOTS*, Fort Worth, TX, Oct. 2002.
[18] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," *IEEE Security and Privacy*, vol. 1, pp. 33–39, July 2003.
[19] M. Garetto, W. Gong, and D. Towsley, "Modeling malware spreading dynamics," in *Proc. IEEE Infocom*, San Francisco, CA, Mar. 2003.
[20] Z. Chen, L. Gao, and K. Kwiat, "Modeling the spread of active worms," in *Proc. IEEE Infocom*, San Francisco, CA, Mar. 2003.
[21] Z. Ge, D. Figueiredo, S. Jaiswal, J. Kurose, and D. Towsley, "Modeling peer-peer file sharing systems," in *Proc. IEEE Infocom*, San Francisco, CA, Mar. 2003.

[22] K. Gummadi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload," in *Proc. ACM SOSP*, Bolton Landing, NY, Oct. 2003.
[23] G. de Veciana and X. Yang, "Fairness, incentives and performance in peer-to-peer networks," in *Proc. Allerton Conf. Comm., Control and Computing*, Monticello, IL, Oct. 2003.
[24] D. Qiu and R. Srikant, "Modeling and performance analysis of bittorrent-like peer-to-peer networks," in *Proc. ACM Sigcomm*, Portland, OR, Aug. 2004.
[25] R. Thommes and M.J. Coates, "Epidemiological models of P2P viruses and pollution," Tech. Rep., Dept. Electrical and Computer Engineering, McGill University, Jun. 2005, `http://www.tsp.ece.mcgill.ca/Networks/publications.html`.
[26] J. Liang, R. Kumar, Y. Xi, and K. W. Ross, "Pollution in p2p file sharing systems," submitted for publication, 2004.
[27] Eytan Adar and Bernardo A. Huberman, "Free riding on Gnutella," *First Monday*, vol. 5, no. 10, October 2000.
[28] Matei Ripeanu, Adriana Iamnitchi, and Ian Foster, "Mapping the gnutella network," *IEEE Internet Computing*, pp. 50–57, January/February 2002.
[29] Stefan Saroiu, P. Gummadi, and Steven Gribble, "A measurement study of peer-to-peer file sharing systems," in *Proceedings of Multimedia Computing and Networking*, 2002.
[30] Subhabrata Sen and Jia Wang, "Analyzing peer-to-peer traffic across large networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 219–232, April 2004.
[31] K. Tutschku, "A measurement-based traffic profile of the edonkey filesharing service," in *Proc. Passive and Active Measurement Workshop*, Juan-les-Pins, France, Apr. 2004.
[32] "BayTSP," `http://www.baytsp.com`.
[33] John Borland: CNET News, "Kazaa loses p2p crown," `http://asia.cnet.com/news/systems/0,39037054,39197197,00.htm`, October 2004.
[34] Ronald Rivest, "RFC 1186: The MD4 message–digest algorithm," `http://www.rfc-editor.org/rfc/rfc1320.txt`, April 1992.
[35] "Ethereal network protocol analyzer," `http://www.ethereal.com/`.
[36] Oliver Heckmann and Axel Bock, "The eDonkey2000 protocol," Tech. Rep. Version 0.8, Darmstadt University of Technology, December 2002.
[37] "Emule," `http://www.emule-project.net`.
[38] Electronic Frontier Foundation, "RIAA vs. the people," `http://www.eff.org/IP/P2P/riaa-v-thepeople.php`.