A Probabilistic Population Study of the Conficker-C Botnet

Rhiannon Weaver

CERT, Software Engineering Institute
rweaver@cert.org

Abstract. We estimate the number of active machines per hour infected with the Conficker-C worm, using a probability model of Conficker-C's UDP P2P scanning behavior. For an observer with access to a proportion δ of monitored IPv4 space, we derive the distribution of the number of times a single infected host is observed scanning the monitored space, based on a study of the P2P protocol, and on network and behavioral variability by relative hour of the day. We use these distributional results in conjunction with the Lévy form of the Central Limit Theorem to estimate the total number of active hosts in a single hour. We apply the model to observed data from Conficker-C scans sent over a 51-day period (March 5th through April 24th, 2009) to a large private network.

Key words: Botnets, Conficker, Population Estimation, Probability Models, Central Limit Theorem

1 Introduction

When new botnets emerge, the classic question is, "How big is it?" In the statistical literature, population estimation is based on mark-recapture models and their extensions to a wide class of generalized linear models [6]. In network analysis, simple mark-recapture techniques, which reduce to counting intersections among overlapping sets, have been applied to study botnet populations [3, 8, 9], as well as to other phenomena such as peer-to-peer file sharing networks [10, 16]. But the "overlapping sets" method is valid only for closed populations with direct observation of individuals of interest, and equal probability of capture for all individuals. Internet phenomena often violate these assumptions, resulting in the need for more sophisticated modeling techniques.

Extending mark-recapture models to open populations is widely addressed in the literature (eg. [17]), but network phenomena often admit a specific complication of direct observation: we would like to express population sizes in terms of the number of infected machines, but we view botnets through a filter of IP space. The existence of NAT, proxies and DHCP leases complicates the "one IP address, one host" model.

Applying mark-recapture models to machines, as opposed to IP addresses, requires averaging aggregations and distributions of activity across possible configurations of disambiguated hosts. On the other hand, applying mark-recapture

models directly to IP addresses introduces heterogeneity among individuals; for example, a NAT is observed if at least one of its underlying hosts is observed, whereas a DHCP address is observed only if the single host to which it is allocated is observed. [4] present a solution for the case when heterogeneity can be modeled as a series of observable, nominal classes. But heterogeneity in botnet behavior often arises from variations in underlying rates of observed counts.

This leads us to consider a method for measuring the active size of a botnet, based on the observable behavior of a single infected host. Our dynamic measurement of active machines per hour cannot be used to track a botnet's overall "footprint" size [1] across days or weeks. But it is simpler to implement than a fully specified heterogeneous mark-recapture model. A single-host behavioral model is also a necessary component of the generalized mark-recapture methodology, so this work provides a stepping stone toward applying more complicated models.

As an example, the Conficker-C variant that emerged within the Conficker botnet in March 2009 introduced a specific pattern of peer-to-peer (P2P) activity. When a host infected with Conficker-C comes online, it searches for peers by randomly generating a set of destination IP addresses across most of IPv4 space, and attempting UDP connections to these hosts. Connection ports use an algorithm based on the source IP address and date, which was cracked by several independent researchers [5, 13]. As a result, Conficker-C P2P traffic can be observed with high reliability in the large-scale summary information contained in network flow data, making it a good candidate for behavioral modeling.

We model the hourly number of UDP P2P connection attempts that an observer monitoring a proportion δ of IP space would see from a single infected host. Rather than inferring a time series of UDP scan activity for each machine, disambiguated from NAT or DHCP addresses, our model represents the "typical host" by averaging across reasonable probability distributions for many unobservable parameters. This marginal model is used in conjunction with the Central Limit Theorem to estimate the total number of active hosts per hour, with confidence intervals that account for measurement uncertainty, stochastic elements in the Conficker-C protocol, and random variation across network activity.

Section 2 discusses the stochastic components of the Conficker-C P2P protocol and network behavior that inform the marginal model, and formalizes this information into a probability model. Section 3 introduces a version of the Central Limit Theorem that lets us describe the distribution of aggregate scan attempts of all active hosts per hour. Section 4 presents results of applying the method to data collected over a 51-day period from a large network. Section 5 summarizes.

2 Modeling Conficker-C

We develop our model in two steps. First, we use information from published reports and studies of the Conficker-C P2P protocol to specify the distribution of the number M_h of hourly UDP connection attempts made by an infected host.

Next, we specify the conditional distribution of the number y_h of observed hits in the monitored space given M_h. We use $\pi(a)$ to denote the prior distribution of quantity a, and $\pi(a \mid b)$ to denote the conditional distribution of a given b. We use μ_a or $E(a)$ to denote the mean of quantity a, and σ_a^2 or $\mathrm{Var}(a)$ to denote the variance of a.

Protocol and Network Behavior. In September 2009, [15] provided a de-obfuscated reverse engineering of the image of the Conficker-C P2P binary image as it appeared on March 5, 2009. We use this information to determine the protocol-specific variations in the distribution of UDP scan attempts per hour for an active, infected host.

When initiated, the P2P module spawns a global UDP scanning thread for each valid network connection discovered, in order to bootstrap a peer list of up to 2048 peers. Up to 32 threads can run simultaneously. Each thread alternates between a 5-second sleep cycle and a scan phase where it randomly generates a list of up to 100 IP addresses to contact. At each selection, the host chooses an IP address from its list of n peers with probability equal to

$$\gamma_n = \left(1000 - \left\lfloor \frac{950n}{2048} \right\rfloor\right)^{-1}. \tag{1}$$

This expression is taken directly from the C code in Conficker-C's P2P module. If the choice is not to select a peer, the host pseudo-randomly generates an IP address, which is added to the list only if it satisfies the following connection criteria:

1. the IP address is not a DHCP or broadcast address;
2. the IP address is not a private (RFC1918) subnet address;
3. the IP address is not on a Conficker-C filtered address range.

When a generated IP address fails to meet these criteria, the value for the contact list is not updated. The host will try to fill its list slots in order, using up to 100 attempts.

The speed at which UDP packets are sent out over the wire depends on the hardware and network capabilities of the infected host, as well as the amount of bandwidth, drop percentage, etc. of the network. The P2P protocol has a maximum of 1200 scanning connection attempts per minute, but observed accounts of Conficker-C P2P scan activity cite lower numbers. McAfee [11] reported seeing "roughly 2-3 UDP queries per second" (≈ 130 per minute) during the 24 hours leading up to April 1, 2009. A Sophos technical report [7] notes that batches of 100 probes are generated on the wire, and that "probes in each batch are separated by small fixed intervals (2-5 seconds)". [14] performed a sandbox test of an infected Conficker-C host with a single network interface and observed scanning rates that start at approximately 1000 to 2000 IP addresses per 5 minute interval, and decrease over the first two hours of activity to a steady rate of approximately 200 IP addresses per 5 minutes. We base our model roughly on the SRI results, as they most thoroughly explain the time-dependence in scanning rate.

UDP connections. We model M_h as a Poisson process, which is a reasonable model for small-packet scanning activity programmed at regular intervals. [12] note that self-similarity is more common in packet inter-arrival times once connections have been established. Also, in their sandbox experiment, [14] show relatively smooth scanning rates, within both 30-minute and 6-hour time frames.

The marginal model is constructed to minimize dependencies between parameters from hour to hour, so that each population estimate can be calculated using an aggregated count from that hour alone. The goal for this model is not to track individual hosts, but to average over a wide range of possible behaviors in each hour. To that end, we take a simplified approach to the time-dependency of the UDP scanning rate, by defining a latent class η_h as one of three states that an active, infected host can be in for a particular hour h:

1. $\eta_h =$"Start-up" (S): The host comes online and initiates P2P scanning in this hour. This state is characterized by a high scan rate per minute and a small peer list, with activity commencing at some point t within the hour.
2. $\eta_h =$"Running" (R): The host has initiated start-up and is actively scanning for the entire hour. This state is characterized by a low scan rate per minute, a middle-sized to large peer list, and scans occurring throughout the hour.
3. $\eta_h =$"Shut-down" (D): The host has been actively scanning and goes offline during this hour. This state is characterized by a low scan rate per minute, a large peer list, and scans terminating at some point t within the hour.

Each of these states depends on three quantities: scan rate (ϕ), active minutes (t), and number of peers (n), that vary from hour to hour. We describe this variability mathematically using the prior distributions in Table 1. We suppress the index h for ease of notation.

Table 1. Prior distributions by active state

s	$\pi(\phi)$	$\pi(t)$	$\pi(n)$
Start-Up	$\Gamma(\mu_{\phi S} = 130, \sigma_{\phi S} = 20)$	Unif$(1, 60)$	TrGeo$(2048, \alpha_S = 0.950)$
Running	$\Gamma(\mu_{\phi R} = 40, \sigma_{\phi R} = 15)$	$t = 60$ w.p. 1	TrGeo$(2048, \alpha_R = 0.999)$
Shut-Down	$\Gamma(\mu_{\phi D} = 40, \sigma_{\phi D} = 10)$	Unif$(1, 60)$	TrGeo$^*(2048, \alpha_D = 0.999)$

$(*)$ this prior is defined as $\pi(2048 - n)$

Figure 2 shows the Gamma prior $\pi(\phi)$ for each state. Gamma distributions are often used to model the mean of a Poisson process, as they have strictly positive ranges. The mean rates $\mu_{\phi s}$ decrease from Start-Up through Shut-Down, and the standard deviations $\sigma_{\phi s}$ decrease to account for less stable behavior in Start-Up that gradually settles down to the more stable Running and Shut-Down states. Discrete Uniform priors on t represent the total number of active minutes in the Start-Up or Shut-Down states. Truncated Geometric distributions (geometric distribution restricted to a minimum and maximum value) are used for peer list counts. For the shut-down state, the $(*)$ indicates that the truncated

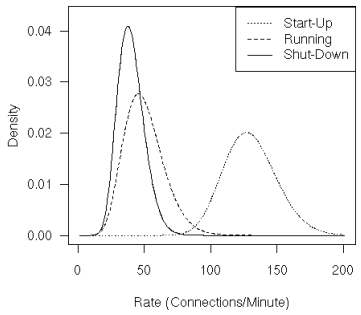

Fig. 1. Prior distributions for UDP scan rates by active state.

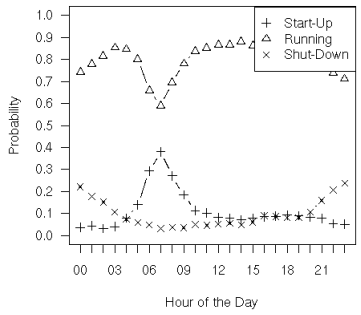

Fig. 2. Prior probabilities π_{ks} by relative hour of the day.

geometric distribution is defined on the range $2048 - n$. The hyperparameters $\pi(n)$ correspond to mean peer list sizes of approximately 20, 700, and 1350 for Start-Up, Running, and Shut-Down states.

We assume that the number of network connections (w) for an infected host does not change between states; based on elicitation from experts we choose a truncated geometric distribution between 1 and 32 connections, with a mean value $\mu_w = 1.67$ network connections per active host. When these quantities are fixed or known, it follows that M has a Poisson distribution with conditional mean μ_M equal to ϕtw.

Again to minimize dependencies between hours, temporal trends in the scan rates are not instituted by a time series component in the single-host model, but by the prior probability of the active state, $\pi_{ks} = \pi(\eta_k = s), s \in \{S, R, D\}, k \in [0, 1, \cdots, 23]$, which varies with the time-zone corrected hour of the day. Intuitively, π_{ks} is an estimate of the proportion of active hosts in the population that are in each state at each time-zone corrected hour. There are 48 free parameters in this distribution, arising from two free parameters per relative hour to estimate the probability of $\{S, R, D\}$ under its sum-to-one constraint. Figure 2 summarizes the values used for π_{ks}. We describe the empirical method used to set these values in Section 4.

Observed Connections. Each of the M connection attempts either hits the monitored proportion δ of IP space, or it does not. Since scan connections are independent, identical events, the distribution $\pi(y \mid M)$ is Binomial with parameters $M$and p, where p is the probability a connection attempt falls within the monitored space.

To determine p, we assume that the monitored space resides completely within the connection criteria from Section 2, and that it does not contain any infected peers listening on Conficker-C's designated ports. The monitored space must be free of infected machines to ensure that the only way of reaching the

monitored space is through a completely random selection of an IP from all of IPv4 space; infected hosts may also reside on peer lists, which we do not choose to model.

With a set of n existing peers, the probability that each connection attempt is generated randomly is $1 - \gamma_n$. If the connection attempt is generated randomly, the probability that it falls in the monitored region is equal to δ/C, where $C = 0.995$ is the approximate proportion of IP space covered by Conficker-C's connection criteria, under the assumption of 1 broadcast address per 256 addresses in the space outside of Conficker-C's internal blacklist and ignored space. From these calculations p is equal to:

$$p_{n\delta} = \frac{(1-\gamma_n)\delta}{C}, \tag{2}$$

where δ is the proportion of monitored IP space.

From a well-known distributional result (see e.g. [18], ch 5, thm 1.2), the marginal distribution of y over all values of M, holding other unknown quantities fixed, is Poisson:

$$\pi(y \mid t, \phi, w, p_{n\delta}) = e^{-\phi t w p_{n\delta}} \frac{(\phi t w p_{n\delta})^y}{y!} \ . \tag{3}$$

Though we have described the model in hierarchical stages, in practice we are interested only in the unconditional distribution of y, which is difficult to express analytically. But, the hierarchical structure of the model makes it easy to obtain a large sample $(y)_1, \cdots (y)_B$ from this distribution, using simulation. For $b = 1$ to B, we do the following:

1. Draw a state η_b from $\{S, R, D\}$ using the prior probabilities π_{ks}, and draw a network connection w_b from $\pi(w)$.
2. Draw ϕ_b, n_b, and t_b using the prior distributions for η_b.
3. Draw y_b from the Poisson distribution with rate equal to $\phi_b t_b n_b w_b$.

We then use the observed proportions in $y_1, ..., y_B$ as Monte Carlo estimates of the marginal probability of y, accounting for prior uncertainty in the underlying parameters. This simulation can be performed easily using statistical packages for languages such as R, C, or Python.

3 Lévy's Central Limit Estimator $\hat{H}$

Suppose $y_1, ..., y_H$ are independent with distribution $\pi(y \mid \pi_{ks}, \mu_{\phi s}, \sigma_{\phi s}, \alpha_w, \alpha_n, \delta, k)$ as defined in Section 2. We will suppress the dependence on hyperparameters in notation for this section. The population size H is unknown, and only $Y = \sum_{i=1}^{H} y_i$ is observed. We define the population estimator:

$$\hat{H} = \frac{Y}{\mu_y} \ . \tag{4}$$

We call $\hat{H}$ the Central Limit Estimator of H. This estimator has the following properties:

1. $E(\hat{H}) = \frac{1}{\mu_y} E(\sum_{i=1}^{H} y_i) = \frac{1}{\mu_y} H\mu_y = H$;
2. $\mathrm{Var}(\hat{H}) = \frac{1}{\mu_y^2} \mathrm{Var}(\sum_{i=1}^{H} y_i) = H \left(\frac{\sigma_y}{\mu_y}\right)^2$;
3. (Lévy result): The distribution of $\hat{H}$ is approximately Normal when H is sufficiently large.

The Lévy form of the Central Limit Theorem (see e.g. [2], Ch. 5, p 243) outlines conditions under which the sum of independent and identically distributed variables converges to a Normal distribution. Using this result, an approximate 95% confidence interval for H is:

$$\hat{H} \pm 1.96\sqrt{\hat{H}}\frac{\sigma_y}{\mu_y} \tag{5}$$

When $y_{k1}, ..., y_{kH_k}$ are identically distributed when grouped within relative hour of the day, $k \in [0, ..., 23]$, then an approximate 95% confidence interval of $\hat{H} = \sum_{k=0}^{23} \hat{H}_k$ is:

$$\hat{H} \pm 1.96\sqrt{\sum_{k=0}^{23} \hat{H}_k \left(\frac{\sigma_{yk}}{\mu_{yk}}\right)^2}\,. \tag{6}$$

We estimate μ_y and σ_y from simulations $y_1, \cdots y_B$ for $B = 1,000,000$, with the formulas:

$$\mu_y \approx \frac{1}{B}\sum_{b=1}^{B} y_b, \qquad \sigma_y^2 \approx \tfrac{1}{B-1}\textstyle\sum_{b=1}^{B}(y_b - \mu_y)^2\,. \tag{7}$$

In practice, B can be set large enough that the Monte Carlo sampling error in these estimates has little effect on the variance of $\hat{H}$. Alternatively, an approximation method such as the Delta method ([2], ch 7) can be used to account for this variability.

4 Analysis and Results

Data Collection. The monitored space in our experiment consists of a large private network comprising approximately 21000 /24 net blocks. To account for uncertainty in this size estimate as well as network availability, we also set an additional prior for δ, $\pi(\delta) = \mathrm{Beta}(15, 13000)$, and add an extra simulation step in the calculation of $\pi(y \mid \pi_{ks}, \mu_{\phi s}, \sigma_{\phi s}, \alpha_w, \alpha_n, \delta, k)$. This corresponds to a mean μ_δ of 0.0012.

Using the SiLK Conficker.C Plug-In [5], we obtained historical records of UDP connection requests with the Conficker-C signature sent into the monitored network space from external hosts over the period from March 5th through April 24th, 2009. We recorded the total number of incoming UDP connection attempts for each external IP address per hour, and aggregated these counts to the /24

level to attempt to account for ephemeral DHCP leases within subnets. A total of 1091013 external /24 net blocks were observed performing Conficker-C UDP scans.. Each net block was assigned roughly to a time zone based on the country code associated with that block, with 1% of blocks remaining unassigned due to satellite locations or unavailable country codes.

Population Estimates. Figure 4 shows the estimates of H_h for the two-month span starting on March 5th, and ending April 2 4th. 95% confidence bands for the hourly counts, calculated using equation 6, are on the order of under ± 10000 and are too tight to be seen on the figure. The large jump occurs on March 17th and corresponds to a binary update released into the Conficker-C botnet. The largest host count associated with the botnet is 1.06 million active hosts. Numbers decline steadily through the month of April, but appear to stabilize toward the end of the month. The overall decline occurs because Conficker-C infections spread only among previously infected machines, with had no means of infecting new hosts.

The heavy lines correspond to a smoothed plot of both host count estimates (solid line), and observed unique IP address counts (dotted line). These lines show a trend that as the botnet ages, it "spreads out" among IP space. The ratio of hosts per IP–observable as the space between the two lines– is large prior to the update in mid-March, but declines steadily afterward. This decline makes sense; large infected networks (often behind proxies) would propagate local infections quickly, while isolated hosts would take longer to reach with P2P bootstrapping. The effect would also appear as larger corporate networks work to clean up enclaves of local infections, suggesting that the persistent infections of Conficker-C are among more isolated machines in IPv4 space.

Hyperparameters and Prior Sensitivity. The prior values π_{sk} can be estimated empirically using a random sample of infected *hosts* from the Conficker-C population. To approximate such a sample, we sampled a set of 1000 /24 blocks from the observed set of 1.09 million, each with probability proportional to the observed average scan rate. In 71% of the sample, the activity behind the sampled net block was sparse enough to roughly equate one block with one active host, and to estimate active from non-active hours. An active hour following two inactive hours, or an active hour preceded and followed by two inactive hours, was classified as "Start-Up". An active hour preceded by at least one active hour in the past two, and followed by two inactive hours was classified as "Shut-Down". All other active hours were classified as "Running". The counts of observed blocks in each state, normalized over hour by time zone, were used as the values for π_{sk}. This method used simple heuristics as opposed to formal models for estimating an active state, but the resulting prior probabilities display a reasonable and intuitive pattern in Figure 2.

The scaling factor $\frac{\sigma_y}{\mu_y}$ was close to 1 for all hours, with the highest value of 1.08 occurring at 8am, relative time. This result indicates that the $\sqrt{H}$ term dominates the confidence interval for $\hat{H}$. The value $1.96\sqrt{H}$ is a very tight bound relative to the size of the population estimate, but its precision is predicated on

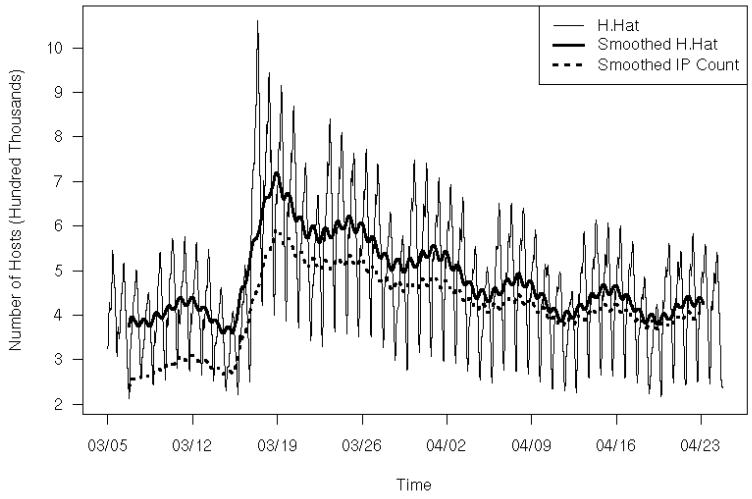

Fig. 3. $\hat{H}$ per hour over the 2-month span.

an unbiased model for $\pi(y \mid \pi_{ks}, \mu_{\phi s}, \sigma_{\phi s}, \alpha_w, \alpha_n, \delta, k)$. Small shifts in the hyperparameters may have a large influence on $\hat{H}$. This suggests that measurement of the uncertainties and behavioral quantities making up a single-host model should be well-informed and precise to take advantage of this simple estimator. We used a reasonable and informed set of 67 hyperparameters $(\pi_{ks}, \mu_{\phi s}, \sigma_{\phi s}, \alpha_w, \alpha_n, \delta, k)$ in this model, and we opted not to model any further levels of uncertainty with probability distributions. In the future, the model can be easily adapted to include another hierarchical level of priors for these hyperparameters, allowing us examine the sensitivity of population estimates to the choice of hyperparameters.

5 Summary and Discussion

A marginal probability model of single-host behavior provides a way of measuring populations based on the number of active infected machines, as opposed to counting net blocks or IP addresses. The model is based on a set of hyperparameters that can be independently measured or assessed based on protocol and network activity profiles. By characterizing this distribution precisely, and applying the Central Limit Theorem, we obtain both a point estimate of the population, and a confidence interval that accounts for variability arising from both the stochastic elements of the protocol and from uncertainty across multiple measurements. In the future we hope to expand the estimation methodology to a fully Bayesian scheme that incorporates priors for the chosen hyperparameters and that allows for the calculation of posterior distributions of the current

model hyperparameters given observed data y, making the model more robust to parameter misspecification. We also hope to develop a full mark-recapture model for comparison with the expanded marginal model.

References

1. Abu Rajab, M., Zarfoss, J., Monrose, F., Terzis, A.: My botnet is bigger than yours (maybe, better than yours): Why size estimates remain challenging. In: Proceedings of the First Annual Workshop on Hot Topics in Botnets (March 2007)
2. Casella, G., Berger, R.: Statistical Inference. Duxbury Press (1990)
3. Chan, M., Hamdi, M.: An active queue management scheme based on a capture-recapture model. IEEE Journal on Selected Areas in Communications 21(4), 572–583 (2003)
4. Dupuis, J., Schwarz, C.: A Bayesian approach to the multistate Jolly-Seber capture-recapture model. Biometrics 63, 1015–1022 (2007)
5. Faber, S.: Silk Conficker.C Plug-in. `http://tools.netsa.cert.org/wiki/display/tt/SiLK+Conficker.C+Plugin` (2009), CERT Code release
6. Fienberg, S., Johnson, M., Junker, B.: Classical multilevel and bayesian approaches to population size estimation using multiple lists. Journal of the Royal Statistical Society: Series A 162(3), 383–405 (1999)
7. Fitzgibbon, N., Wood, M.: Conficker.C: A technical analysis. `http://www.sophos.com/sophos/docs/eng/marketing_material/conficker-anal%ysis.pdf` (March 2009), Sophos white paper
8. Horowitz, K., Malkhi, D.: Estimating network size from local information. Information Processing Letters 88, 237–243 (2003)
9. Li, Z., Goyal, A., Chen, Y., Paxson, V.: Automating analysis of large-scale botnet probing events. In: ASAICCS '09 (March 2009)
10. Mane, S., Mopuru, S., Mehra, K., Srivastava, J.: Network size estimation in a peer-to-peer network. Tech. Rep. TR 05-030, University of Minnesota Department of Computer Science and Engineering (2005)
11. McAfee: Conficker.C over the wire. `http://www.avertlabs.com/research/blog/index.php/2009/04/01/confickerc-%on-the-wire-2` (March 2009), McAfee Network Security blog publication
12. Paxson, V., Floyd, S.: Wide-area traffic: The failure of poisson modeling. IEEE/ACM Transactions on Networking 3(3), 226–244 (1995)
13. Porras, P., Saidi, H., Yegneswaran, V.: Conficker C Actived P2P scanner. `http://www.mtc.sri.com/Conficker/contrib/scanner.html` (2009), SRI international Code release/document
14. Porras, P., Saidi, H., Yegneswaran, V.: Conficker C analysis. Tech. rep., SRI International (2009)
15. Porras, P., Saidi, H., Yegneswaran, V.: Conficker C P2P protocol and implementation. Tech. rep., SRI International (2009)
16. Psaltoulis, D., Kostoulas, D., Gupta, I., Briman, K., Demers, A.: Decentralized schemes for size estimation in large and dynamic groups. Tech. Rep. UIUCDCS-R-2005-2524, University of Illinois Department of Computer Science (2005)
17. Schwarz, C., Arnason, A.: A general methodology for the analysis of capture-recapture experiments in open populations. Biometrics 52(3), 860–873 (1996)
18. Taylor, H., Karlin, S.: An Introduction to Stochastic Modeling. Academic Press (1998)