Physica A 387 (2008) 6190–6200

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Analyzing open-source software systems as complex networks

Xiaolong Zheng [a], Daniel Zeng [a,b,*], Huiqian Li [a], Feiyue Wang [a,c]

[a] The Key Lab of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, China

[b] Department of Management Information Systems, The University of Arizona, USA

[c] Department of Systems and Industrial Engineering, The University of Arizona, USA

ARTICLE INFO

Article history:
Received 15 January 2008
Received in revised form 9 June 2008
Available online 6 July 2008

Keywords:
Open-source software systems
Complex networks
Degree distribution

ABSTRACT

Software systems represent one of the most complex man-made artifacts. Understanding the structure of software systems can provide useful insights into software engineering efforts and can potentially help the development of complex system models applicable to other domains. In this paper, we analyze one of the most popular open-source Linux meta packages/distributions called the Gentoo Linux. In our analysis, we model software packages as nodes and dependencies among them as edges. Our empirical results show that the resulting Gentoo network cannot be easily explained by existing complex network models. This in turn motivates our research in developing two new network growth models in which a new node is connected to an old node with the probability that depends not only on the degree but also on the "age" of the old node. Through computational and empirical studies, we demonstrate that our models have better explanatory power than the existing ones. In an effort to further explore the properties of these new models, we also present some related analytical results.

1. Introduction

In the past few years, we have witnessed dramatic growth in the body of the literature studying complex networks in a wide variety of fields. Examples include the movie actor collaboration [1,2], science collaboration [3], WWW [4–6], Internet [7,8], protein–protein interaction networks [9], transportation networks [10], and e-mail networks [11]. Software systems [13–18] represent another important class of systems that can be studied using the complex network analysis framework. Such systems are one of the most complex man-made artifacts and play a critically important role in the modern society. The U.S. National Institute of Standards and Technology (NIST) estimated in 2002 that software faults alone can cost the U.S. economy $59.5 billion annually, representing about 0.6% of the U.S. gross domestic product [12]. Understanding the structure of software systems can potentially lead to better software engineering practice and increased system reliability. In particular, complex network theory can contribute to the development of quantifiable measures of software structure that in turn can help to develop large software systems with desirable structural designs and identify potential structural problems.

Research on studying software from the perspective of complex systems is emerging but to date has not received wide-spread attention [13]. We suspect that the main reasons behind this are the difficulties with data collection and the lack of applicable models [14]. In the past few years, however, the fast growth of open-source software has allowed researchers to collect data easily. In addition, the recent complex network literature is also making an increasingly rich set of models available to analyze software systems. In this paper, we focus on an analysis of software systems using complex

* Corresponding address: Department of Management Information Systems, The University of Arizona, McClelland Hall 430K, Tucson, USA.
E-mail address: zeng@email.arizona.edu (D. Zeng).

0378-4371/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.physa.2008.06.050

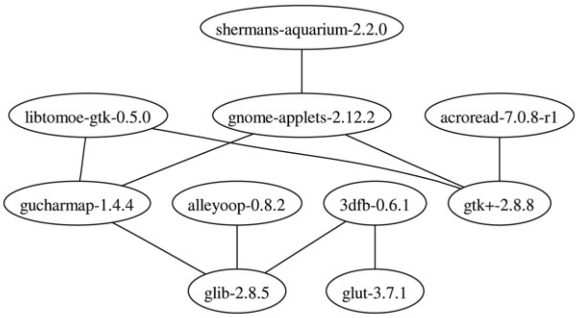

Fig. 1. The Gentoo network.

network theory. A software system is composed of many interacting units and subsystems and the interactions among them directly reflect the design, coding, and execution of software. As such, analyzing and modeling software systems as complex networks can afford us meaningful understanding regarding the formation and evolution of code-based software structures and the processes governing the development of software systems. Furthermore, understanding the functional organization of evolving software systems may also provide new insights into network growth in a significant engineering setting that could potentially help us to gain better understanding of other classes of complex networks.

In this paper, we analyze one of the most popular Linux meta packages/distributions called the Gentoo Linux operating system. In Gentoo Linux, software applications are distributed in the form of packages. As is common in open-source software development, the developers of many software applications rely on using other open-source packages. Such dependencies among packages can manifest themselves in different forms. For instance, one package needs the source code of another package to compile correctly. In another form, source-code dependencies do not exist; rather, the binary-level library sharing is required for a package to function properly. Such software package dependencies often span across many different projects. A package management system serves the purpose of managing such dependencies, which is important for both functioning and maintenance (e.g., automated updating) of software packages. The Portage system is a sophisticated package management system used by Gentoo Linux fashioned after the well-tested package management system used in another major open-source project—the Berkeley Software Distribution (BSD). Gentoo's portage manages more than 10000 packages and has explicit information about dependencies among packages. Such dependencies represent to a large degree the true structure of a large software.

In our research, we study the package dependencies existing in Gentoo Linux. By treating packages as nodes and their dependencies as edges, we construct the Gentoo network.The rest of this paper is organized as follows. In Section 2, we present an empirical analysis of the Gentoo network. As some key properties of this network cannot be explained by the existing network evolution models, we have been motivated to develop our own models. In Section 3, we discuss three related known models and then propose two new evolution models, in which a new node is connected to an old node with the probability that depends not only on the degree but also on the age of the old node. Section 4 presents a computational study based on these models. The results of this study are compared against the findings from the real-world Gentoo network. These simulation results show that our models have better explanatory power than the existing models. In Section 5, we present some analytical results related to our new models. Finally, Section 6 concludes the paper by summarizing our findings and discussing possible future research.

2. An empirical analysis of an open-source software dependence graph

Linux is an open-source operating system that has a rich set of application software packages. Gentoo Linux is one of the most popular Linux meta package distributions with a flexible and powerful software management facility called the Portage system. The Portage system explicitly tracks dependencies among software packages, making software installation and upgrading automatic and efficient. The Gentoo Linux has been quite popular in both its user base and developer community since its inception, in part due to the power and flexibility of Portage. The number of the software packages in Gentoo Linux has been growing very quickly in the past few years. Until February 2007, Gentoo's Portage has managed over 16800 packages. Most of the data used in this paper were collected from the Portage and the Gentoo Web site [19]. Note that our data are quite different from those used in previous studies concerning software. In [13], the authors analyze small-scale software systems (e.g., DM consisting of 187 packages, VTK 788 packages). Data used in [14] are mainly concerned with software change events from revision histories or logs and do not capture package dependencies.

Fig. 1 illustrates what the Gentoo package dependency network (referred to as the Gentoo network thereafter) looks like, where the nodes represent the packages available from Gentoo and the edges the dependencies among packages. For example, nodes labeled as 3dfb-0.6.1 and glib-2.8.5 represent packages 3dfb-0.6.1 and glib-2.8.5 respectively; whereas the edge between nodes 3dfb-0.6.1 and glib-2.8.5 indicates the dependency between these two packages.

Table 1
summary of various symbols for graph measures

Symbol	Measure
Nn	Number of nodes in the network
Ne	Number of edges in the network
k	Degree
R	Degree growth rate
M	The Mth month
$P(k)$	Degree distribution of a node with degree k
C	Clustering coefficient

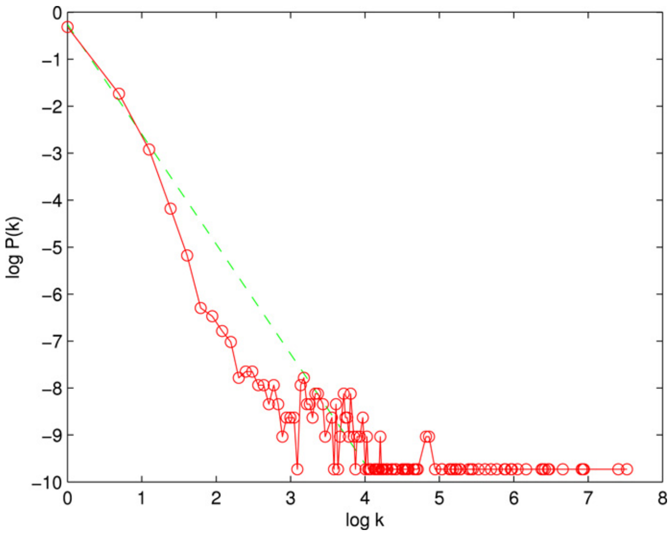

Fig. 2. Plot of the degree distribution of the Gentoo network.

In this section, we make brief observations concerning the properties of the Gentoo network. Some symbols used in this section are summarized in Table 1. We mainly focus on degree distribution but also discuss the sparsity, clustering coefficient, degree growth rate, and node growth of the network.

Sparsity

As of February 2007, the number of nodes Nn in the Gentoo network is 16 803, and the average degree $\langle k \rangle$ is about 2.7576. The total number of edges Ne is 23 168 while the total number of edges of the complete graph with 16 803 nodes is $C^2_{16\,803} = (16\,803 \times 16\,802)/2 = 141\,162\,003$. The ratio between them is about 0.016%. The Gentoo network is sparse in general.

Clustering Coefficient

The clustering coefficient C_i of the node i is defined as $2e_i/k_i(k_i - 1)$, where e_i is the number of the existing edges between the k_i neighbors of node i. For nodes with degree 0 or 1, we set $C_i = 0$. The clustering coefficient C for the entire network is the average of C_i over all the nodes [20]. The clustering coefficient C of the Gentoo network is about 0.0318 while the clustering coefficient of a corresponding random graph $C_{\text{rand}} = \langle k \rangle / Nn = 0.0001641$. That is, the clustering coefficient of the Gentoo network is about 194 times higher than that of the random graph.

Degree distribution

Degree distribution in random graphs is either binomial or Poisson when the size of the graph is large. However, many real-world networks have been found to follow different patterns. For instance, many networks' degree distribution follows the power-law property, while others' exhibits non-power-law features such as exponential cutoffs [21–24]. Fig. 2 shows the degree distribution of the Gentoo network in $\log k$ to $\log P(k)$ view. Note that log in this paper is natural logarithm. The plot shows that the degree distribution of the Gentoo network does not follow the Poisson distribution and a pure power-law distribution. Rather they closely resemble a stretched exponential distribution because the curve declines more rapidly that the dashed line with a slope 2.3391. However, the tail of the curve appears to be much longer than that of a stretched

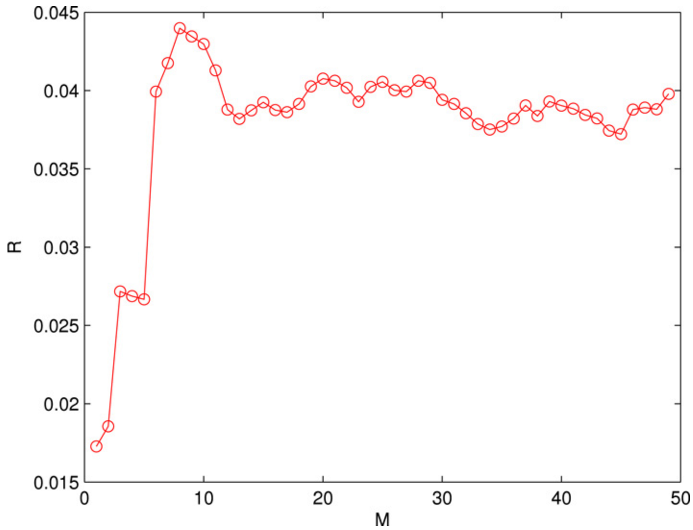

Fig. 3. Average growth rate as a function of the time.

exponential distribution. Amaral et al. [21] and Albert et al. [25] have found this similar phenomenon in other real-world networks and suggested that the aging effect of the nodes can be a leading explanation.

Degree growth rate

To further illustrate the impact of the age of nodes on their attractiveness (in terms of being used by a new package), we conduct an empirical analysis based on degree growth rate for a selected set of 539 packages/nodes. These packages are those that are already recorded in Gentoo's portage system by January 2003 and have been kept as an active part of Gentoo until at least February 2007. We observe how these nodes' degrees change on average over this 50-month time window. To quantify the degree growth rate for month t for package i, we use $R_i(t)$, defined as $(k_i(t) - k_i(t-1))/k_i(t)$ where $k_i(t)$ denotes the degree of i at month t. Fig. 3 plots the average degree growth rate as a function of the time. This plot indicates that the average degree growth rate of these packages exhibits an interesting two-phase behavior. In the first phase, the rate grows and reaches a maximum point whereas in the second phase, the rate begins to decrease slowly. This two-phase behavior strongly suggests the presence of the aging effect and has an intuitive explanation. As Gentoo Linux grows, some important software packages are heavily utilized in new packages, exhibiting the "winner takes all" phenomenon. However, with the rapid increase of available software packages in Gentoo Linux as shown in Fig. 4, the number of packages that can provide similar functions or capacities as those popular packages also increases. As a result, more choices become available, potentially reducing the attractiveness of those high-degree packages.

3. Model development

As shown in our empirical analysis presented in the previous section, the Gentoo network is sparse and has a high clustering coefficient. In addition, some nodes have very large degrees and the degree distribution has a large tail. This section presents our work attempted to develop models to explain the evolution of the Gentoo network. We start with a brief review of some related known models and then present our models.

3.1. Related models

In the past few years, a large number of network models [26–34] have been developed. Some of these models are extensions to the classic BA model [2], which is based on two mechanisms: incremental growth and preferential attachment. We discuss briefly three models relevant to our study, namely the BA model, the Krapivsky–Redner–Leyvraz (KRL) model [27], and the Dorogovtsev–Mendes (DM) model [28].

BA model

In this model, the network starts with an initial set of nodes. At each time the network grows with the addition of new nodes. For each newly added node, new edges are added between it and some old nodes. The nodes to receive new edges are chosen following a linear preferential attachment rule, that is, the probability $\Pi(k)$ of an old node receiving a new edge is proportional to its degree k,

$$\Pi(k) \sim k. \tag{1}$$

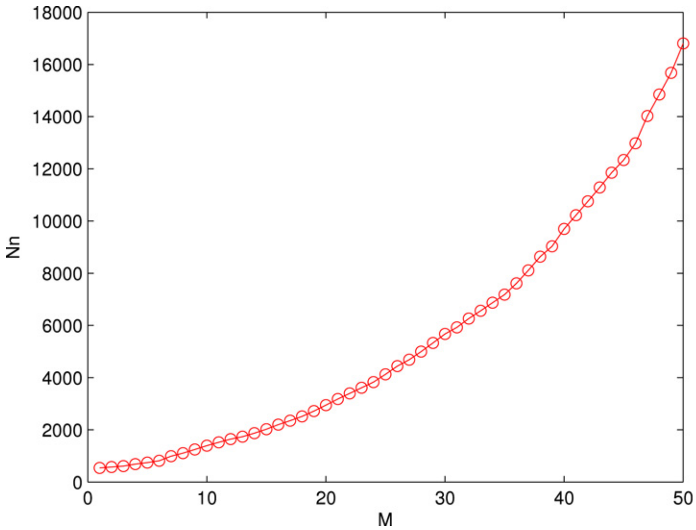

Fig. 4. The Node growth.

When k is sufficiently large, the degree distribution $P(k)$ follows a power-law dependence $P(k) \propto k^{-\gamma}$ with a fixed exponent. The BA model captures a mechanism that can result in the power-law degree distribution. On the other hand, it predicts a power-law degree distribution with a fixed exponent, whereas the exponents measured for many real networks can vary between 2.1 and 4. In addition, the degree distribution of real networks can have non-power-law features such as exponential cutoffs or saturation for small k [21–25].

KRL model

The model is proposed by Krapivsky, Redner, and Leyvraz [27]. Replacing the BA model's linear preferential attachment, they use a nonlinear preferential attachment rule. When choosing the nodes to which a new node connects, the probability $\Pi(k)$ depends on k^{α},

$$\Pi(k) \sim k^{\alpha}, \tag{2}$$

where α is a tunable parameter.

DM model

The BA model and the KRL model have one common characteristic: their preferential attachment rules depend only on the degree of the old node. However, in applications such as reference networks aging occurs: the authors rarely cite very old papers. Dorogovtsev and Mendes proposed an extended model in which the probability $\Pi(k)$ is dependent not only on the degree k of the old node but also on its age τ, that is,

$$\Pi(k) \sim k\tau^{-\beta}, \tag{3}$$

where β is a tunable parameter.

3.2. Two new models

Based on our empirical observations, we note that in the Gentoo network some nodes have a clear finite lifetime. Standard models such as the BA and the KRL do not model such aging effect. See Fig. 5(a) and (b) for simulation results showing the unsuitability of these models in explaining the Gentoo network. The DM model explicitly considers not only the node's degree but also its age. However, there is still a large gap between what the model predicts and the real-world observations as shown in Fig. 5(c). Part of the reason seems to be that when the nodes are very "young," the DM model shows approximately linear preferential attachment whereas many real-world networks including the Gentoo network display nonlinear preferential attachment.

We propose two new models to explain the evolution of networks exemplified by the Gentoo network: the Degree Dependent adjustable Evolution with Aging (DDEA) model and the Degree and Age dependent Adjustable Evolution (DAAE) model. These two models extend the KRL and DM models by following similar growth patterns but with different preferential attachment rules. In DDEA and DAAE, the probability that a new node is connected to an old node is not only proportional to the degree k of the old node but also dependent on the age τ of the old node. Details of these two models are as follows.

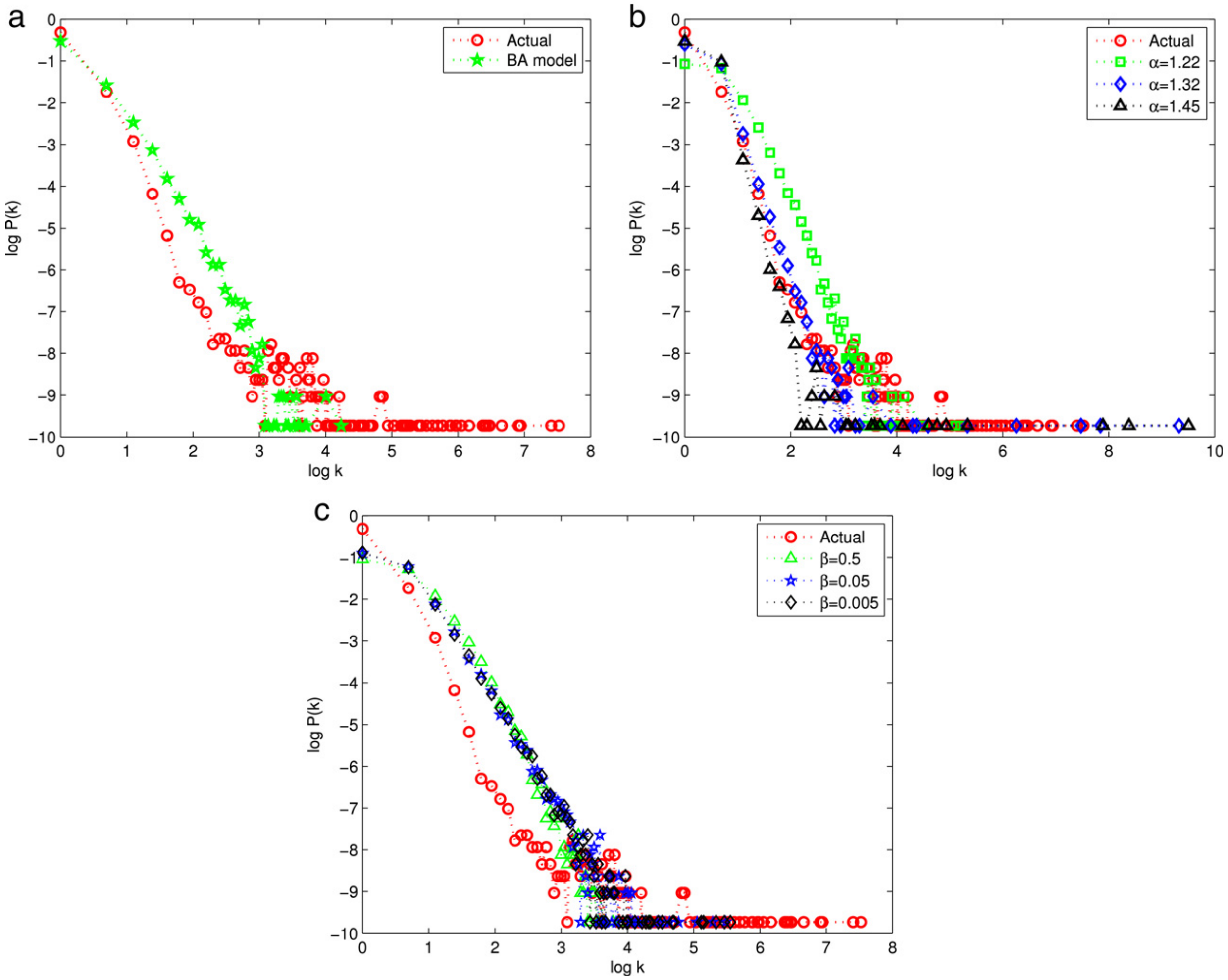

Fig. 5. Comparison of the degree distributions in log–log plots. (a) Simulation results of the BA model. (b) Simulation results of the KRL model, for $\alpha = 1.22, 1.32, 1.45$ respectively. (c) Simulation results of the DM model, for $\beta = 0.5, 0.05, 0.005$ respectively.

Network growth

We observe that the Gentoo network starts with a small number of nodes and then the number of nodes in the network grows very rapidly. Both DDEA and DAAE start with a small number (m_0) of connected nodes. Subsequently, at each time step we add a new node with m edges that link this new node to m existing nodes. Note that the number of newly added edges m is not fixed. Rather it is drawn from a known discrete distribution.

Preferential attachment

In the Gentoo network, some nodes have very high degrees, corresponding to some popular software packages based on which many other applications have been developed. This suggests preferential attachment based on the degree of the old node. Also we often observe that with the growth of Gentoo Linux, new software packages become available with a set of functionalities extending (or reimplementing) those offered by the existing popular packages. Soon after that, developers are starting to use these new packages and the popularity of the old ones starts to drop. This substitution effect leads to our modeling effort that adjusts the preferential attachment probabilities based on both the degree and the age of an node. The preferential attachment rules of DDEA and DAAE follow different forms.

(1) In DDEA, the attachment probability $\Pi(k)$ is given by

$$\Pi(k) \sim k^{\alpha} e^{-\beta\tau}, \tag{4}$$

where τ is equal to $t - s_k$, with t denoting the current time and s_k the time the node k was born. (We dropped the subscript denoting the node index from τ to simplify the notation.) Parameters α and β are tunable. In this model, a young high-degree node has a high chance to be connected to a new node. The aging effect, however, will make the node less attractive as time progresses even if it is already well-connected.

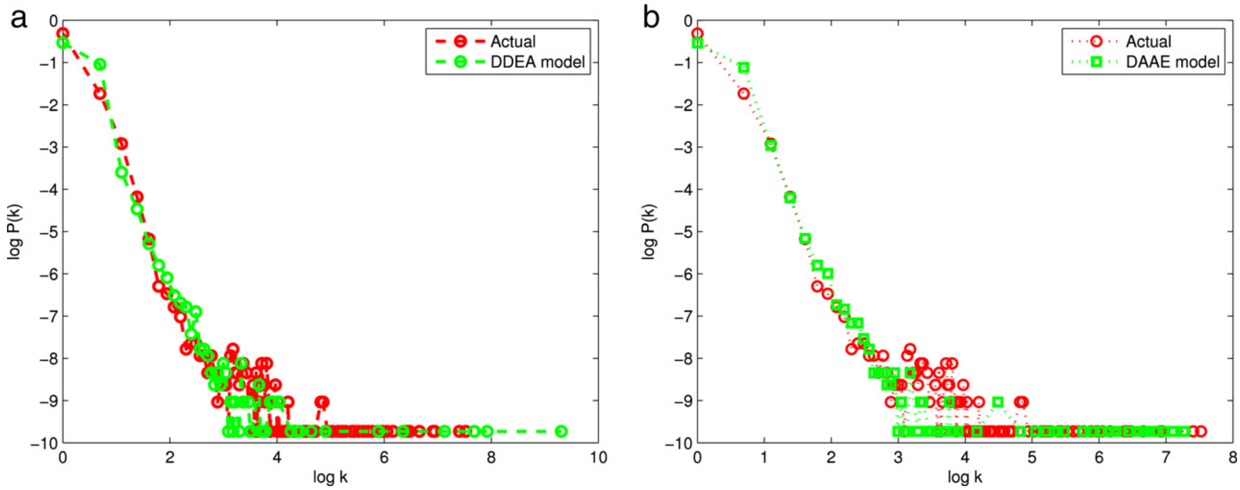

Fig. 6. Comparison of the degree distributions in log–log plots. (a) Simulation results of the DDEA model, for $\alpha = 1.24$ and $\beta = 0.0006$. (b) Simulation results of the DAAE model, for $\alpha = 2.14$ and $\beta = 0.00073$.

(2) In DAAE, $\Pi(k)$ takes the following form:

$$\Pi(k) \sim k^{\alpha}\tau^{-\beta k}. \quad (5)$$

Similar to DDEA, DAAE ensures that a young, well-connected node will have high probability to be connected to a new node and that aging will reduce the connection probability. The key difference between these two models is that in DAAE a young node not yet well-connected will have a higher probability to be connected to a new node.

From a modeling perspective, we hypothesize that DDEA and DAAE can be potentially fruitfully applied to describe other similar networks whose nodes have a finite lifetime (e.g., reference networks) or whose edges have finite capacity (e.g., the Internet router network). Such studies are beyond the scope of this paper.

4. A computational study

Based on the models described in Section 3, we have conducted a computational study. The simulation results are compared with the actual observations made from the Gentoo network. The simulation procedure governing the initial network setup and subsequent growth closely resembles the experimental method used by Liu et al. [6]: We start with two connected nodes. At each step of network growth, a new node with m edges will be connected to m different old nodes until the total number of nodes reaches the number of actual nodes in the Gentoo network. Here m is not a constant but a random value drawn from the set $\{1, 2\}$ with corresponding probabilities $\{p_1, p_2\}$. These two probabilities are estimated by equations $N_1 + N_2 = N_{\text{new}}$ and $N_1 + 2N_2 = E_{\text{new}}$, where N_{new} denotes the total number of newly added nodes, E_{new} the total number of newly added edges, and N_1 and N_2 the numbers of the newly added nodes with one edge and two edges, respectively. Since the total numbers of nodes and edges in the Gentoo network are 16 803 and 23 168 respectively, in our simulations, N_{new} is set to 16 801 and E_{new} to 23 167. We calculated $p_1 = \frac{N_1}{N_{\text{new}}} = 0.6211$ and $p_2 = \frac{N_2}{N_{\text{new}}} = 0.3789$. Because of the stochastic nature of the network growth, each simulation setup was repeated for 30 times and the results are the averages over these 30 replications. Fig. 5 summarizes the simulation results of the BA, KRL, and DM models. Fig. 6 summarizes the simulation results based on our DDEA and DAAE models.

Fig. 5(a) shows the degree distributions of the BA model and the Gentoo network. We note that the degree distribution of the Gentoo network deviates significantly from that produced by the BA model. The absolute value of the slope of the Gentoo network is larger than that of the BA model. The Gentoo network's degree distribution is less smooth than the BA model's and has a significantly longer tail. The clustering coefficient produced by BA model is 0.000261, which is far smaller than that of the observed value 0.0318. Fig. 5(b) compares the degree distributions of the KRL model and the Gentoo network. In the case of $\alpha = 1.32$, the degree distribution of the KRL model provides a good match for the Gentoo network when the degree is not large. However, as the degree increases, the differences start to show. The maximum degree of the KRL model simulation results is far larger than that of the Gentoo network. The clustering coefficient in this case is 0.2361, which is much larger that the actual value. When α is equal to other values, we also cannot obtain good results. For example, when α is equal to 1.45, in this case, the clustering coefficient 0.3440 and the degree distribution are different from the actual value. When α is equal to 1.22, though the clustering coefficient is 0.0584, which is close to the actual one. However, the degree distribution in this case significantly deviates from the actual curve. Fig. 5(c) shows the degree distributions of the DM model and the Gentoo network. We can clearly see their differences. When the parameter β is equal to 0.5, 0.05, and 0.005, the absolute values of the slope of the DM model are smaller than that of the actual curve and the tails are much shorter. The resulting clustering coefficients are equal to 0.0002447, 0.0008696, and 0.0016. All of these three values are much smaller than that of the actual value. Setting β to other values seems to reduce the overall descriptive power of the DM model. The general observation is that none of these three models provide a good explanation of the observed properties of the Gentoo network.

The degree distributions of the DDEA and DAAE models are shown in Fig. 6 (a) and (b), respectively. From Fig. 6(a) we observe that the simulation results of the DDEA model are similar to the actual observations in general, indicating that the DDEA gives a better description of the data than the three existing models examined before. We also note the remaining problems: The maximum node degree of the networks generated by DDEA is significantly larger than that of the actual Gentoo network. The clustering coefficient in this case is 0.1801, which is a little larger than that of the actual value. We experimented with various settings of α and β. It is possible to obtain smaller maximum degree but always at the expense of much poorer descriptive power with respect to other aspects of the degree distribution. Also the clustering coefficients are different from the actual value. From Fig. 6(b), we observe that the simulation results of the DAAE model mimic the actual network very well. The slope of the DAAE's degree distribution is very close to that of the Gentoo network when the degree is not so large. The tail end of the DAAE degree distribution also closely resembles that of the Gentoo network with similar maximum degrees. The clustering coefficient produced by our DAAE is 0.0381, which is very close to the actual value. Overall, the DAAE model provides better characterization of the Gentoo network than other models in multiple dimensions and could provide useful insights into the evolution of Gentoo Linux and networks alike.

5. Analytical properties of the DAAE model

In this section, we derive some useful analytical properties of the DAAE model. These results help to frame some of the computational findings discussed in Section 4 in a formal setting. The technical approach adopted here is largely based on the effective medium approach used by Dorogovtsev and Mendes [28]. Since the results for the exponents do not depend on the number of edges added to the network each time [35], we consider exclusively network growth with one new node and one edge.

We observe that $P(k, s, t)$, the probability that the degree of node s at time t is equal to k, can be obtained by the δ-function, i.e. $P(k, s, t) = \delta(k - \bar{k}(s, t))$, where $\bar{k}(s, t)$ is the mean degree of node s at time t. We have

$$\frac{\partial \bar{k}(s,t)}{\partial t} = \frac{\bar{k}(s,t)^{\alpha}(t-s)^{-\beta\bar{k}(s,t)}}{\int_0^t \mathrm{d}u \bar{k}(u,t)^{\alpha}(t-u)^{-\beta\bar{k}(u,t)}}, \tag{6}$$

where $\bar{k}(0, 0) = 0$, $\bar{k}(t, t) = 1$. We apply $\int_0^t \mathrm{d}s$ to Eq. (6) and obtain

$$\int_0^t \bar{k}(s,t)\mathrm{d}s = 2t. \tag{7}$$

In order to solve Eq. (6), we let

$$\bar{k}(s,t) = \kappa(\xi), \tag{8}$$

where $\xi = s/t$. From Eqs. (7) and (8), we obtain

$$\int_0^1 \kappa(\xi)\mathrm{d}\xi = 2. \tag{9}$$

Based on Eqs. (6) and (8), we derive

$$-\frac{\mathrm{d}\kappa(\xi)}{\mathrm{d}\xi}\xi = \frac{\kappa(\xi)^{\alpha}(t-t\xi)^{-\beta\kappa(\xi)}}{\int_0^1 \kappa(\zeta)^{\alpha}(t-t\zeta)^{-\beta\kappa(\zeta)}\mathrm{d}\zeta} \tag{10}$$

and

$$\kappa(1) = 1. \tag{11}$$

Using the fact

$$\int_0^1 \kappa(\zeta)^{\alpha}(t-t\zeta)^{-\beta\kappa(\zeta)}\mathrm{d}\zeta = \gamma, \tag{12}$$

we rewrite Eq. (10) as

$$-\frac{\mathrm{d}\kappa(\xi)}{\mathrm{d}\xi}\xi = \frac{1}{\gamma}\kappa(\xi)^{\alpha}(t-t\xi)^{-\beta\kappa(\xi)}. \tag{13}$$

We now give two properties.

Property 1. *If $\alpha \neq 1$, and $\beta \to 0$, then at time t, degree distribution $P(k,t)$ decreases as a stretched exponential in k, such that:*

$$P(k,t) \sim k^{-\alpha} e^{\frac{\gamma}{1-\alpha}(1-k^{1-\alpha})}. \tag{14}$$

Proof. If $\alpha \neq 1$, Eq. (13) can be transformed into

$$d\kappa(\xi)^{1-\alpha} = -\frac{1-\alpha}{\gamma\xi}(t-t\xi)^{-\beta\kappa(\xi)}. \tag{15}$$

As it is difficult to integrate Eq. (15) directly, we apply an approximation scheme using Taylor's formula to expand $(t-t\xi)^{-\beta\kappa(\xi)}$.

$$d\kappa(\xi)^{1-\alpha} \approxeq -\frac{1-\alpha}{A_1\gamma\xi[1+M_1(\xi-\xi_0)]}d\xi, \tag{16}$$

where

$$A = (t-t\xi_0)^{\beta\kappa(\xi_0)}, \tag{17}$$

$$M = \beta\left[\kappa'(\xi_0)\ln(t-t\xi_0) - \frac{\kappa(\xi_0)}{1-\xi_0}\right]. \tag{18}$$

We now solve Eq. (16) and obtain

$$\kappa(\xi)^{1-\alpha} \approxeq -\frac{1-\alpha}{A\gamma(1-M\xi_0)}\ln\left[\frac{\xi}{1+M(\xi-\xi_0)}\right] + C_1. \tag{19}$$

Recalling Eq. (11), we can determine constant C_1. The final solution is

$$\kappa(\xi)^{1-\alpha} \approxeq -\frac{1-\alpha}{A\gamma(1-M\xi_0)}\ln\left[\frac{\xi[1+M(1-\xi_0)]}{1+M(\xi-\xi_0)}\right] + 1. \tag{20}$$

If $\beta \to 0$, from Eqs. (17), (18) and (20) we can see that

$$\kappa(\xi) \sim \left(1 - \frac{1-\alpha}{\gamma}\ln\xi\right)^{\frac{1}{1-\alpha}}. \tag{21}$$

Since $P(k,t) \propto \partial s/\partial k$, we conclude

$$P(k,t) \sim k^{-\alpha} e^{\frac{\gamma}{1-\alpha}(1-k^{1-\alpha})}.$$

Property 2. *If $\alpha = 1$, and $\beta \to 0$, then at time t, degree distribution $P(k,t)$ shows the power-law property and follows*

$$P(k,t) \sim k^{-3}. \tag{22}$$

Proof. If $\alpha = 1$, Eq. (13) can be transformed into

$$d\ln\kappa(\xi) = -\frac{1}{\gamma\xi}(t-t\xi)^{-\beta\kappa(\xi)}d\xi. \tag{23}$$

To obtain an approximate solution, we apply Taylor's formula to $(t-t\xi)^{-\beta\kappa(\xi)}$ and obtain,

$$d\ln\kappa(\xi) \approxeq -\frac{1}{A\gamma\xi[1+M(\xi-\xi_0)]}d\xi. \tag{24}$$

The solution to Eq. (24) is

$$\ln\kappa(\xi) \approxeq -\frac{1}{A\gamma[1-M\xi_0]}\ln\left[\frac{\xi}{1+M(\xi-\xi_0)}\right] + C_2. \tag{25}$$

Recalling Eq. (11), we can determine constant C_2 and get

$$\ln\kappa(\xi) \approxeq -\frac{1}{A\gamma(1-M\xi_0)}\ln\left[\frac{\xi[1+M(1-\xi_0)]}{1+M(\xi-\xi_0)}\right]. \tag{26}$$

If $\beta \to 0$, we obtain

$$\ln \kappa(\xi) \sim -\frac{1}{\gamma} \ln \xi. \tag{27}$$

Recalling Eqs. (9) and (12), we obtain

$$\gamma \to 2. \tag{28}$$

Then

$$\kappa(\xi) \sim \xi^{-\frac{1}{2}}. \tag{29}$$

Finally, we obtain the degree distribution

$$P(k,t) \propto \partial s/\partial k \propto k^{-1-\gamma} \sim k^{-3}. \quad \square \tag{30}$$

From the analysis above, in the DAAE model, we have found that if the parameter α is not equal to 1 and β is very small, then the degree distribution decreases as a stretched exponential. However, if the parameter α is equal to 1 and β is sufficiently small, then the degree distribution of this model exhibits the power-law property.

6. Concluding remarks

In this paper, we present an empirical analysis of the Gentoo network. We have analyzed network properties including degree distribution, sparsity, clustering coefficient, degree growth rate, and node growth. As existing models do not provide satisfactory explanation for the observed characteristics concerning the Gentoo network's degree distribution and clustering coefficient, we have developed and evaluated two new network growth models, DDEA and DAAE, based on the KRL and DM models. By comparing simulation results with empirical observations, we demonstrate that these new models, in particular, the DAAE model, are able to reproduce the observed properties of the Gentoo network better than their predecessors. The DAAE model has two notable features. (a) One component of the preferential attachment probability is based on a nonlinear term of the node degree. (b) The exponent of the aging-related term is dependent on the node degree.

For future research, additional evaluation of the DDEA and DAAE models needs to be conducted, in both software research and other application domains. As to open-source research, work is being pursued by several groups to analyze the developer network [15–18]. It will be interesting to study the co-evolution between developer and software package networks. From both engineering (package dependency) and management (developer network) perspectives, investigating intrinsic fitness measures for each node as in Ref. [33] and incorporating such measures in model development for open-source software present many interesting research opportunities. As pointed out in Ref. [13], investigating software systems from a complex network perspective helps one to gain better understanding of relationships among software network structure, object complexities, object interactions, development processes, and system evolution. Study a number of open-source software systems beyond Gentoo Linux and developing models and measures that could be applicable to open-source software engineering in general could lead to fruitful research contributions.

Acknowledgments

We thank the reviewers for the valuable comments.We would like to thank Liang Sun, Wei Cui, Feng Zhang, and Hong Mo, for useful discussions and helpful suggestions. This work was supported by the National Natural Science Foundation of China under Grants. 60621001 and 60573078, by the Chinese Academy of Sciences through an international collaboration Grants 2F05NO1 and 2F07C01, and by the Ministry of Science and Technology under the 973 program Grants. 2006CB705500 and 2004CB318103. The second author wishes to acknowledge support from the U.S. National Science Foundation through Grants IIS-0527563 and IIS-0428241.

References

[1] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (1998) 440.
[2] A.L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 509.
[3] S. Redner, How popular is your paper? An empirical study of the citation distribution, Eur. Phys. J. B 4 (1998) 131.
[4] R. Albert, H. Jeong, A.L. Barabási, The diameter of the world-wide web, Nature 401 (1999) 130.
[5] B. Huberman, P. Pirolli, J. Pitkow, R. Lukose, Strong regularities in world wide web surfing, Science 280 (1998) 95.
[6] J.G. Liu, Y.Z. Dang, Z.G. Wang, T. Zhou, Relationship between the in-degree and out-degree of WWW, Physica A 371 (2006) 861.
[7] M. Faloutsos, P. Faloutsos, C. Faloutsos, On paw-law relationships of the internet topology, Comput. Commun. Rev. 29 (1999) 251.
[8] R.W. Dai, L.B. Cao, Internet — an open complex giant system, Sci. China (Ser. E) 33 (2003) 289.
[9] V. Colizza, A. Flammini, A. Maritan, A. Vespignani, Characterization and modeling of protein–protein interaction networks, Physica A 352 (2005) 1.
[10] J.R. Banavar, A. Maritan, A. Rinaldo, Size and form in efficient transportation networks, Nature 399 (1999) 130.
[11] J. Wang, P.D. Wilde, Properties of evolving e-mail networks, Phys. Rev. E 70 (2004) 066121.
[12] NIST, The economic impacts of inadequate infrastructure for software testing, Gaithersburg, MD, May 2002.

[13] C.R. Myers, Software systems as complex networks: Structure, function, and evolvability of software collaboration graphs, Phys. Rev. E 68 (2003) 046116.
[14] C.F. Kemerer, S. Slaughter, An empirical approach to studying software evolution, IEEE Trans. Softw. Eng. 25 (1999) 493.
[15] J. Xu, Y. Gao, S. Christley, G. Madey, A topological analysis of the open source software development community, in: Proceedings of the 38th Hawaii International Conference on System Sciences, 2005 (Paper presented).
[16] Krogh G. Von, S. Spaeth, K.R. Lakhani, Community, joining, and specialization in open source software innovation: A case study, Res. Policy. 32 (7) (2003) 1217.
[17] S. Sowe, I. Stamelos, L. Angelis, Identifying knowledge brokers that yield software engineering knowledge in OSS projects, Inform. Softw. Technol. 48 (11) (2006) 1025.
[18] J. Xu, S. Christley, G. Madey, Application of social network analysis to the study of open source software, in: The Economics of Open Source Software Development, Elsevier, Amsterdam, 2006, p. 247.
[19] http://www.Gentoo.org.
[20] M.E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2003) 167.
[21] L.A.N. Amaral, A. Scala, M. Barthélémy, H.E. Stanley, Classes of small-world networks, Proc. Natl. Acad. Sci., USA 97 (2000) 11149.
[22] H. Jeong, S.P. Mason, A.-L. Barabási, Z.N. Oltvai, Lethality and centrality in protein networks, Nature 411 (2001) 41.
[23] M.E.J. Newman, Scientific collaboration networks. I. Network construction and fundamental results, Phys. Rev. E 64 (2001) 016131.
[24] M.E.J. Newman, Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality, Phys. Rev. E 64 (2001) 016132.
[25] R. Albert, A.L. Barabási, Statistical mechanics of complex networks, Rev. Modern Phys. 74 (2002) 47.
[26] S.N. Dorogovtsev, J.F.F. Mendes, A.N. Samukhin, Size-dependent degree distribution of a scale-free growing network, Phys. Rev. E. 63 (2001) 062101.
[27] P.L. Krapivsky, S. Redner, F. Leyvraz, Connectivity of growing random networks, Phys. Rev. Lett. 85 (2000) 4629.
[28] S.N. Dorogovtsev, J.F.F. Mendes, Evolution of reference networks with aging, Phys. Rev. E. 62 (2000) 1842.
[29] E. Ravasz, A.L. Barabási, Hierarchical organization in complex networks, Phys. Rev. E. 67 (2003) 026112.
[30] S.N. Dorogovtsev, A.V. Goltsev, J.F.F. Mendes, Pseudofractal scale-free web, Phys. Rev. E. 65 (2002) 066122.
[31] S. Jung, S. Kim, B. Kahng, A geometric fractal growth model for scale free networks, Phys. Rev. E. 65 (2002) 056101.
[32] A.L. Barabási, E. Ravasz, T. Vicsek, Deterministic scale-free networks, Physica A 299 (2001) 599.
[33] G. Bianconi, A.L. Barabási, Competition and multiscaling in evolving networks, Europhys. Lett. 54 (2001) 436.
[34] G. Ergün, G.J. Rodgers, Growing random networks with fitness, Physica A 303 (2002) 261.
[35] A.L. Barabási, R. Albert, H. Jeong, Mean-field theory for scale-free random networks, Physica A 272 (1999) 173.