INSTITUTO DE FÍSICA
Universidade Federal Fluminense

ANÁLISE DA ESTRUTURA DA REDE DE PACOTES DO SISTEMA OPERACIONAL DEBIAN GNU/LINUX

por Orahcio Felício de Sousa

Niterói-RJ

2009

Orahcio Felício de Sousa

ANÁLISE DA ESTRUTURA DA REDE DE PACOTES DO SISTEMA OPERACIONAL DEBIAN GNU/LINUX

Dissertação apresentada ao Instituto de Física da UFF como requisito parcial para obtenção do grau de mestre em Física.

Abril de 2009

Dissertação de mestrado sob o título *"Análise da Estrutura da Rede de Pacotes do Sistema Operacional Debian GNU/Linux"*, defendida por Orahcio Felício de Sousa.

Prof. Dr. Thadeu Josino Pereira Penna
Orientador
Universidade Federal Fluminense

Prof. Dr. Márcio Argollo de Menezes
Co-Orientador
Universidade Federal Fluminense

Prof. Dr. Jeferson Jacob Arenzon
Universidade Federal do Rio Grande do Sul

Prof. Dr. Antônio Tavares da Costa Júnior
Universidade Federal Fluminense

Agradecimentos

Esse trabalho é fruto de minha interação com vários indivíduos que não posso deixar de citá-los. Primeiramente os culpados de minha existência e grande parte do meu caráter, mãe e pai, Maria Gilvanda Felício de Sousa e José Benício de Sousa, muito obrigado mais uma vez.

A chegada à cidade de Niterói não seria tão legal se não fosse um grande amigo e futuro compadre o professor Nilton de Almeida Araújo (UNIVASF-BA) pela estada inicial e ótimas discussões político-científico-sociais nas tradicionais cervejas pós-almoço de domingo, muito obrigado.

Aos grandes amigos que surgiram ao longo dessa breve estada, Vladimir Gonçalves Miranda pelas ótimas discussões e cultura em geral, Pâmella Gonçalves Barreto, amigos e companheiros de bandejão (obrigado por existir) Daniel Lourenço e Luiz Fernando; muito obrigado.

Ao pessoal da "República do O Paí Ó" pelo ótimo convívio e grandes amizades, Míriam Goes de Aragão Ferreira, Érico Raimundo Pereira de Novais e Joyce Martins; o mestrado foi muito mais proveitoso no convívio com vocês, muito obrigado.

Aos professores e amigos Márcio Argollo de Menezes e Thadeu Josino Pereira Penna pelo ótimo período de pesquisa e aprendizado, muito obrigado.

O agradecimento institucional torna-se mais fácil pelo bom acolhimento durante minha estada no programa de pós-graduação daqui, muito obrigado à todos e todas num agradecimento póstumo à João, que conduzia muito bem os trâmites da secretaria. A impossibilidade de pós-graduação fora do estado seria certa se não fosse os ógãos que disponibilizam os recursos para tal, a CAPES e a FAPERJ foram as agências responsáveis por isso.

Em outras palavras, os agentes (...) caracterizados pelo volume de seu capital determinam a estrutura do campo em proporção ao seu peso, que depende do peso de todos os outros agentes (...)

Pierre Bourdieu

Resumo

O sistema operacional Debian GNU/Linux é pioneiro em gerenciamento de pacotes por dependências, isso permite a construção de sua rede conforme a precedência de instalação de pacotes neste sistema.

Uma análise dessa estrutura é realizada como estudo de grafos direcionados, extraindo características como distribuição de conectividade, propagação de danos na estrutura, centralidade de entroncamento e coeficiente de agrupamento. Observou-se com isso possíveis mudanças na estrutura de desenvolvimento por meio das versões estável, de teste e instável do Debian.

A formação desse sistema analisada por meio da detecção de estruturas modulares dentro da rede, para essa análise algumas outras redes, tais como redes sociais, foram utilisadas para uma primeira validação das metodologias empregadas: método de modularidade espectral, e modularização por arrefecimento simulado. Verifica-se que tais estruturas são formadas em torno de pacotes mais fundamentais no sistema.

Uma revisão bibliográfica a respeito de grafos não direcionados é realizada inicialmente para um melhor entendimento das grandezas calculadas.

Abstract

The Debian GNU/Linux operational system is known by managing packages and its dependencies since the very first years of Linux, allowing the study of its network according to the precedence of the packages installation in this system.

A closer examination of its structure is performed as study of directed graphs, extracting features such as degree distribution, the propagation of bugs in its structure, betweenness centrality and clustering coefficient. We observe changes in the structure of development through the Debian distributions: stable, testing and unstable.

We analyzed the processes of the system formation by the detection of modular structures within this network and previously tested in other ones as social networks using other methodologies as modularity spectral and modularisation by simulated annealing. We found that such structures are built around the most basic packages in the system.

Sumário

Lista de Figuras

Lista de Tabelas

Introdução

O estudo da estrutura de rede de sistemas reais é bastante difundido atualmente. Estes podem ser montados a partir da existência de muitos componentes que possuem algum tipo de relação entre si: rede de atores que contracenam com outros colegas construindo vínculos profissionais; rede de citações científicas com artigos referenciados em outros artigos; rede de interação genética entre agentes inibidores ou catalizadores que participam de reações de síntese protéica; etc.

Sistemas tecnológicos, como a *internet*, em particular possuem em sua natureza a estrutura de rede e não são apenas abstrações que transpõem o sistema para um grafo. Consequentemente, nestes sistemas as grandezas estudadas estão diretamente ligadas às condições reais de tráfego de informação, consumo energético, e outros fatores dependentes da topologia do sistema.

Os pacotes que constituem o sistema operacional livre Debian GNU/Linux, possuem um sistema de dependências que permite um estudo de sua estrutura de rede. Embora outras distribuições também possuam uma estrutura semelhante, a escolha do Debian ocorreu pelo pioneirismo desse projeto na adoção de tal metodologia gerenciadora de pacotes e principalmente pelo grande *ensemble* que esse projeto proporciona – um grupo de desenvolvedores mantendo mais de 20000 pacotes.

Esse trabalho busca fornecer elementos para um melhor entendimento da dinâmica de formação de grandes redes, visto que a rede de dependências traz informações a respeito da história de cada pacote. São feitas sugestões de organização de manutenção do projeto, baseado nos resultados para os agrupamentos que surgem nessa estrutura e as medidas de propagação de falhas na mesma.

O uso da estrutura de rede do Debian é o pano de fundo para o estudo de redes complexas. O

conhecimento acumulado desses tipos de sistema permite enquadrar o Debian numa categoria bem frequente de redes, as redes livre-de-escala. Elas aparecem em outros sistemas como a *World Wide Web*, redes de atores, colaboração científica, citações de artigos, etc.

Para o entendimento desse tipo de rede, o primeiro capítulo irá descrever os principais tipos de redes conhecidas, entendendo os casos que possuem ordenamento e os aleatórios para diferir os sistemas que possuem algum grau de organização (ficando entre o ordenado e o aleatório), como ocorre com a rede de dependências. O conceito de grafo não-direcionado é usado neste ponto para introduzir as grandezas relevantes na identificação dos diversos efeitos que surgem dessas estruturas.

No segundo capítulo a estrutura de rede do projeto Debian começa a ser estudada juntamente com uma revisão das generalizações feitas para o caso em questão de grafos direcionados. Uma análise na mudança dessa estrutura por meio das versões correntes do Debian (estável, de teste e instável) é feita com base nas medidas de tais grandezas. Um breve histórico é apresentado para melhor entendimento de como esse tipo de estudo é possível atualmente com esse sistema.

Os diversos sistemas reais voltam à tona no terceiro capítulo com a análise da estrutura de comunidades. Geralmente sistemas reais possuem densidade de ligações bastante diferente de uma distribuição aleatória, sugerindo a formação de agrupamentos, que podem ser classificados usando metodologias simples de otimização, bastante conhecidas na Física. Tendo em vista tais conjuntos de agentes que possuem interações que ocorrem por algum fator bem determinado (vínculos de amizade, colaborações profissionais, conexão física entre computadores, etc.), é razoável a suposição de agrupamentos preferenciais a partir da distribuição das ligações existentes. Assim algumas outras redes foram usadas nessa análise, dentre elas uma rede social de pessoas e uma rede social de baleias. Nestas redes esse tipo de comportamento é esperado, dada a existência de algum tipo de afinidade, idade, descendência, etc.

Para o caso de redes tecnológicas, a rede de comunicação de dados da Universidade Federal Fluminense também foi usada. Nesta a estrutura de comunidades surge por fatores como

condições de alocação de equipamento, estrutura de ligações entre as máquinas, etc.

Finalmente o estudo de comunidades na rede de dependências permite a identificação de tais estruturas, por conta do processo de criação de pacotes que sempre aproveita as tecnologias já existentes. É possível assim determinar os pacotes centrais dessas estruturas além da comparação de dois métodos de otimização quando aplicados em sistemas com grande número de componentes, como é o caso da rede de dependências.

1 Redes: Definições úteis

O início da teoria de grafos se deu em 1736 com o problema das sete pontes de Königsberg na Prússia (atual Kaliningrado, Rússia), onde era perguntado se será possível percorrer todas as pontes sem repetir nenhuma. A resposta da inexistência de tal caminho foi dada por Leonhard Euler ao transpor o problema para um grafo, `alo` introduzindo os conceitos de vértices e ligações que definem um grafo, veja figura 1.1.

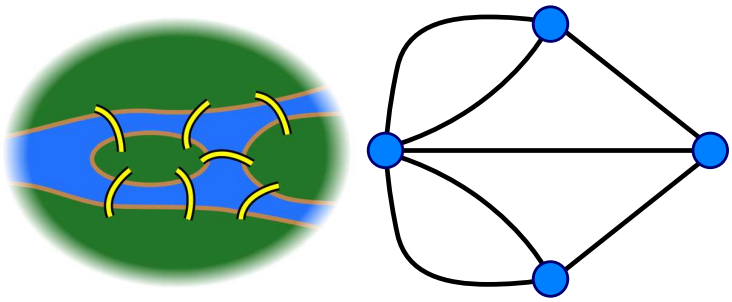


Figura 1.1: Transpondo o problemas das sete pontes de Kaliningrado para um grafo. Ilustração retirada do endereço eletrônico `http://en.wikipedia.org/wiki/Seven_bridges`.

Definimos um grafo como sendo o conjunto G de n vértices V e m ligações E,

$$G := \{V, E\} \tag{1.1}$$

$$V := \{i = 0, 1, ..., n-1\}, n \in \mathbb{N} \tag{1.2}$$

$$E := \{e = (i, j)\}, i, j \in V. \tag{1.3}$$

Os pares do conjunto E geralmente são representados por uma matriz de adjacências (**A**) com elementos $A_{ij} = 1$ se $e_{ij} \in E$, que são simétricas para o caso em estudo neste capítulo, as redes não-direcionadas.

Pode-se também construir grafos em que as ligações são ponderadas como, por exemplo,

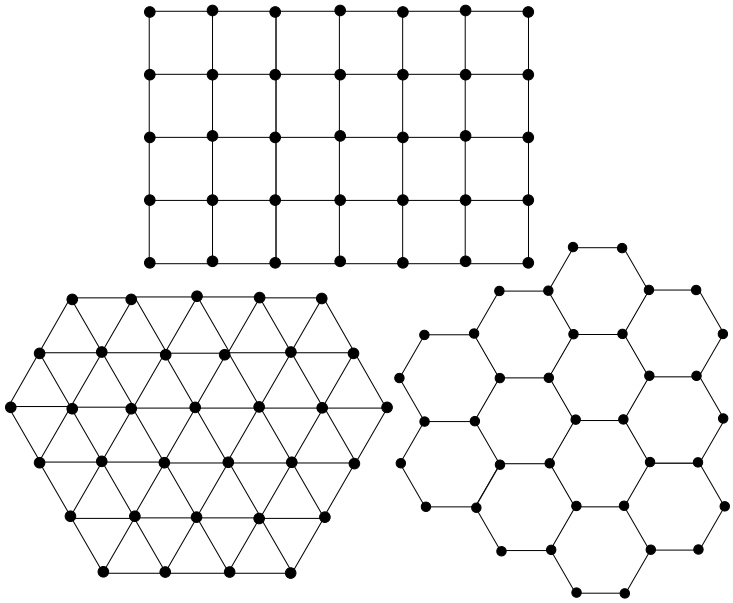


Figura 1.2: Exemplo de redes regulares em duas dimensões: quadrada, hexagonal e colméia.

as redes de aeroportos em que as ligações (pontes aéreas) podem ser ponderadas pela distância entre aeroportos. Neste caso a matriz adjacência **A** pode ser trocada por uma matriz ponderada **W** e seus elementos possuem valor não-nulo se existir ligação. Estes conjuntos são usados para representar diversas estruturas, desde estruturas bem ordenadas até as totalmente aleatórias.

O caso regular (ou ordenado) é bem conhecido na Física do Estado Sólido, como as redes de Bravais [2], em que a disposição das partículas permite a definição de células unitárias que compõe um sistema cristalino. A simetria destas células permite a descrição do sistema por alguns poucos parâmetros, visto que temos a mesma configuração de vértices (átomos) e ligações (compartilhamento de elétrons) se repetindo por toda a estrutura, definindo a célula unitária, que reproduz todo o cristal por simples replicação.

Redes regulares são frequentes e aplicadas em diversos problemas como o da difusão de um gás (gás de rede [25]) e nos problemas de percolação, etc. Suas propriedades geométricas são bem definidas, permitindo uma separação eficiente dos efeitos dessa ordem por comparação entre as diversas geometrias possíveis: redes quadradas, triangulares, hexagonais (veja figura 1.2) e mesmo em dimensões superiores como as redes de Bravais de três dimensões ou hipercúbicas – redes regulares de dimensões maiores que três.

Em contraposição às redes regulares, pode-se imaginar estruturas em que seus elementos unitários não possuem distinção para as interações entre eles, todos os pares de ligações são igualmente possíveis de existirem com uma densidade pequena das mesmas. Isso distingue bastante da rede regular pois não é mais possível verificar periodicidade na estrutura de ligações por exemplo, o número de coordenação já não é bem definido, embora seja possível fazer uso de médias dentro deste conjunto.

1.1 Características gerais das redes aleatórias

1.1.1 Distribuição de conectividade

Uma distribuição define o espectro de possibilidades dentro de um conjunto de eventos (*ensemble*). Mais especificamente para as redes, podemos iniciar o estudo de sua estrutura ao analisarmos como os sítios estão conectados. Para isso supomos uma rede não-direcionada (as ligações não possuem direção preferencial) com n vértices e cada vértice i terá uma probabilidade $p(k,i,n)$ de possuir k conexões. De maneira que a distribuição dada por:

$$P(k,n) = \frac{1}{n}\sum_{i=1}^{n} p(k,i,n), \tag{1.4}$$

é a probabilidade de existência de uma conectividade k, numa rede com n vértices [6].

O primeiro momento desta distribuição será a conectividade média, que é o número médio de ligações que cada vértice apresenta:

$$\langle k \rangle = \sum_{i=1}^{n} k_i P(k_i,n). \tag{1.5}$$

A distribuição de conectividade de uma rede regular é dada por uma ou mais funções do tipo Delta de Dirac e frequentemente seu valor médio coincide com o número de coordenação da rede, por exemplo, numa rede de Bravais cúbica simples, $\langle k \rangle = 6$.

Distribuição de Poisson

Solomonoff e Rapoport em 1957 [24] propuseram um modelo seguindo a idéia de uma rede aleatória, dado um conjunto de n vértices, que são conectados aos pares com uma probabilidade p. De forma independente Erdös e Rényi [8] propuseram este mesmo modelo que define um *ensemble* $G_{n,p}$ de grafos em que m ligações aparecem com probabilidade $p^m(1-p)^{M-m}$, com $M = \frac{1}{2}n(n-1)$. Outro modelo semelhante foi desenvolvido por Erdös e por Rényi em que define o *ensemble* $G_{n,m}$ em que os grafos de exatamente m ligações aparecem com igual probabilidade. Os resultados dos dois modelos são equivalentes. A figura 1.3 mostra o esquema de construção desses gráficos.

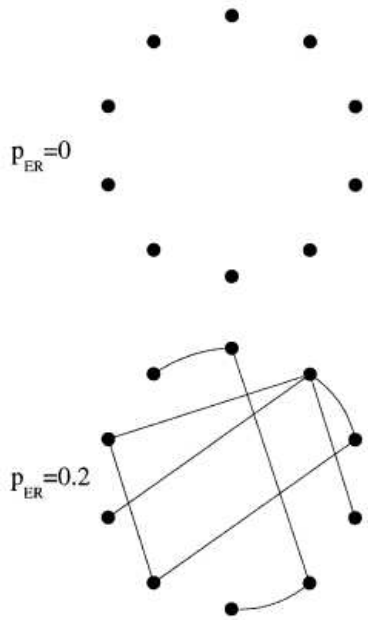


Figura 1.3: Esquema de construção de uma rede aleatória com o modelo Erdös-Rényi. Todos os pares são percorridos sendo ligados com uma probabilidade p_{ER} definindo a concentração de ligações na rede.

A distribuição de conectividade desses grafos segue uma distribuição binomial, visto que existem $\binom{n}{k}$ possíveis grafos com k ligações que aparecem com a probabilidade supracitada, assim,

$$P(k) = \binom{n}{k} p^m (1-p)^{M-m} \tag{1.6}$$

Os grafos aleatórios com o número de vértices grande ($n \to \infty$) e conectividade média finita, apresentam a distribuição dada pela equação 1.7 (distribuição de Poisson). Neste caso, a conectividade média é realmente a conectividade presente na maioria dos vértices (*efeito democrático*). Nestes grafos, a probabilidade de existir uma ligação num sítio é uniforme é proporcional ao inverso do número total de sítios (p=$\frac{\alpha}{n}$), coincidindo com a densidade de ligações, desta forma, se o número de conectividade é muito menor que o de sítios, temos uma segunda equivalência satisfeita.

$$P(k) \simeq \frac{e^{-\langle k \rangle} \langle k \rangle^k}{k!}, \tag{1.7}$$

que é a distribuição de Poisson.

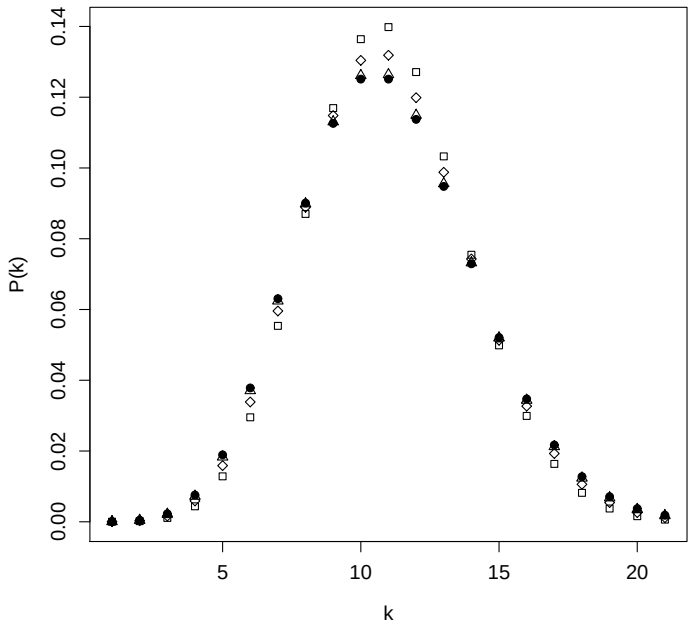


Figura 1.4: Distribuição de Poisson (círculos sólidos) para uma amostragem com $\langle k \rangle = 10$. E distribuição binomial (formas vazias) com a mesma média para $n = 50$ (quadrados), $n = 100$ (losangos) e $n = 500$ (triângulos).

A figura 1.4, onde apresentamos a equivalência entre a distribuição binomial e a de Poisson para n grande, é clara a definição de uma escala por essa distribuição, pois valores muito afastados da média são muito pouco prováveis, mostrando uma variância finita.

Redes de Mundo-Pequeno

Existe um tipo de grafo que faz uma interface entre as redes regulares e as aleatórias, as rede de mundo-pequeno. A origem desse termo se dá com o experimento de Stanley Milgram em 1960, no qual uma amostragem de cartas eram enviadas de uma costa dos Estados Unidos (Omaha, Nebraska e Wichita, Kansas), de um conhecido a outro, até alcançar outra costa (Boston, Massachusetts). Em média, a carta passava por 5 ou 6 pessoas, um número de passos pequeno em relação às dimensões percorridas.

Um modelo reproduz o efeito descrito pelo experimento de Milgram, em conjunto com o alto agrupamento do sistema, foi proposto por Duncan J. Watts e Steven H. Strogatz (WS) em 1998 [26]. Partindo de uma rede regular de número de coordenação K e condições de contorno periódicas, é introduzida uma probabilidade p_{WS} de reconexão aleatória das ligações já existentes. Desta forma ligações de longo alcance são introduzidas na rede pela substituição de $p_{WS}NK/2$ das ligações existentes [26], vide figura 1.5. Desta maneira, o modelo reproduz uma rede regular para $p_{WS} = 0$ e uma rede aleatória em $p_{WS} = 1$, visto que nesta condição todas as ligações serão rearranjadas de forma aleatória.

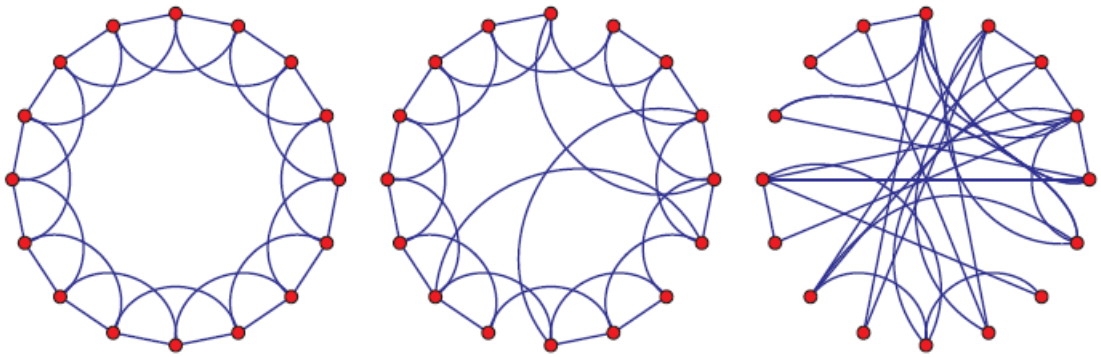


Figura 1.5: Esquema de construção de uma rede mundo-pequeno com o modelo Watts-Strogatz, as figuras mostram a evolução do valor de $p_{WS} = 0$, $0 < p_{WS} < 1$ e $p_{WS} = 1$. Ilustração retirada da referência [26].

O modelo WS possui uma distribuição de conectividade que vai desde uma distribuição do tipo Delta de Dirac (definida apenas em um ponto) que caracteriza uma rede regular, $p_{WS} = 0 \Rightarrow P(k) = \delta(k-K)$, até a distribuição de Poisson que caracteriza a rede aleatória (vide equação 1.4)

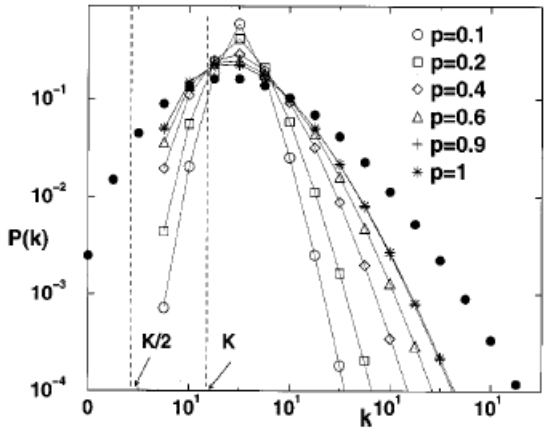


Figura 1.6: Distribuição de conectividade para redes construídas com o modelo WS, com $N = 1000$ e $K = 3$. A linha corresponde à equação 1.7, por comparação os símbolos sólidos corresponde a distribuição de uma rede aleatória. Nota-se que existe apenas dados superiores que $K/2$. Ilustração retirada da referência [1].

para o valor máximo de p_{WS}. A figura 1.6 mostra a distribuição para o modelo WS.

Distribuição em lei de potência

$$P(k) \propto k^{-\gamma}. \tag{1.8}$$

A distribuição do tipo lei de potência pode caracterizar uma distribuição livre-de-escala que diferentemente da distribuição de Poisson, não define uma escala característica, ou seja, estão presentes os mais diversos números de conectividades entre os sítios. Apesar da média finita, aparece uma divergência na dispersão dessa distribuição que ocorre para uma faixa de valores de γ. Ao calcularmos a dispersão ($\sigma_k^2 = \langle k \rangle^2 - \langle k^2 \rangle$) da conectividade neste tipo de distribuição, ao tomarmos a variável k como contínua no limite de $n \to \infty$, temos:

$$\left\langle k^2 \right\rangle \sim \int_0^\infty k^2 \cdot k^{-\gamma} dk = \lim_{k\to\infty}[\ln k] - \lim_{k\to 0}[\ln k], \tag{1.9}$$

a última igualdade é válida para $\gamma = 3$, veja que o segundo termo possui uma divergência muito acentuada, para a média de k temos

$$\langle k \rangle \sim \int_0^\infty k \cdot k^{-\gamma} dk = \lim_{k\to\infty}[\ln k] - \lim_{k\to 0}[\ln k], \tag{1.10}$$

novamente a última igualdade se torna válida quando $\gamma = 2$, o segundo termo novamente possui uma divergência. Temos portanto uma dispersão da variância para a faixa de valores de γ maiores que 2 e menores que 3:

$$\sigma_k^2 = \langle k \rangle^2 - \langle k^2 \rangle = \left(\frac{k^{2-\gamma}}{2-\gamma} \right)^2 - \frac{k^{3-\gamma}}{3-\gamma}, \tag{1.11}$$

essa expressão converge para um valor finito quando $\gamma > 3$, definindo uma escala típica para a conectividade. Para valores menores que dois a média diverge, não sendo portanto de interesse nesse estudo.

O expoente negativo indica simplesmente que os vértices mais conectados são os mais raros, consequentemente, os vértices menos conexos são mais frequentes.

Diversos sistemas reais apresentam este tipo de distribuição com o expoente γ bem definido. A rede de colaboração entre atores, na qual um vértice e designado como um ator e as conexões são formadas por alguma participação conjunta em filmes, é verificada uma distribuição de conectividade em lei de potência com $\gamma_{\text{atores}} = 2,3 \pm 0,1$, figura 1.7A. A WWW, com os vértices sendo os documentos armazenados e as ligações são propriamente os *links* entre eles, possui um expoente $\gamma_{\text{www}} = 2,1 \pm 0,1$, figura 1.7B [1].

O modelo BA, proposto por Albert-László Barabási e Réka Albert em 1999 [3], reproduz tal efeito sem-escala, em que existe a formação de pólos aglutinadores (escassos sítios de alta conectividade) e muitos sítios com baixa conectividade. Para isso dois ingredientes fundamentais são usados:

- *Crescimento*: a cada instante t um sítio i é adicionado à rede inicial ($t = 0$) que possui m_0 sítios, totalizando $N = m_0 + t$ vértices;
- *Ligações preferenciais*: Cada sítio que surge na rede será conectado à $m(\leq m_0)$ diferentes vértices já presentes na rede com uma probabilidade Π dependente da conectividade do sítio, tal que:

$$\Pi(k_i) = \frac{k_i}{\sum_{j \neq i}^{N-1} k_j}, \tag{1.12}$$

sendo assim os vértices com maior conectividade são os mais prováveis de receber novas ligações.

Uma pequena alteração nesse modelo foi feita no presente trabalho levando-se em conta que $N \gg m_0$, a rede inicial possui igualmente m_0 ligações para que em $t = 1$ os sítios sejam igualmente prováveis em receber ligações. No modelo original [3] a soma na equação 1.12 ao final de t passos de tempo é dada por $2mt$, com a alteração temos:

$$\sum_j k_j = 2 \cdot (mt + m_0) \tag{1.13}$$

ou, para o caso particular em que $m_0 = 2$:

$$\sum_j k_j = 2 \cdot (mt + 1)\,. \tag{1.14}$$

A rede resultante, não afeta o expoente verificado inicialmente por Barabási-Albert, obedece portanto a uma lei de potência com expoente $\gamma_{\mathrm{BA}} = 2,9 \pm 0,1$ [3].A distribuição de conectividade das redes BA pode ser verificada a partir de uma aproximação contínua.

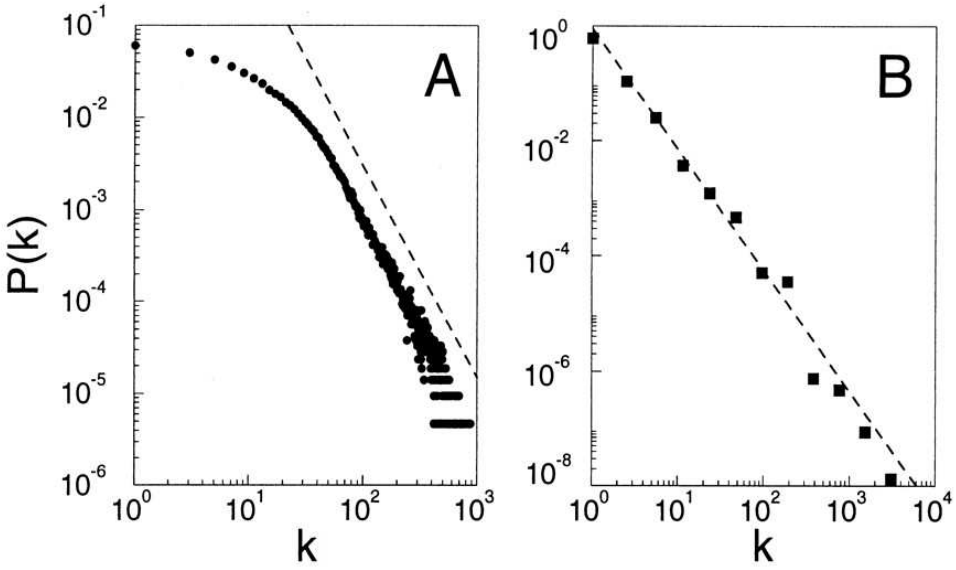


Figura 1.7: Distribuição de conectividade para várias redes extensas. (**A**) Colaboração entre atores com $N = 212.250$ e conectividade média $\langle k \rangle = 28,78$. (**B**) WWW, $N = 325.729$, $\langle k \rangle = 5,46$. As linhas tracejadas possuem expoentes (A) $\gamma_{\mathrm{atores}} = 2,3$ e (B) $\gamma_{\mathrm{www}} = 2,1$. Ilustração retirada da referência [3].

Assumimos aqui a variável k_i contínua e real. A taxa de crescimento desta conectividade é proporcional à quantidade de novas ligações m que surgem a cada instante de tempo, ponderada pela probabilidade de ser efetuada uma nova ligação nesse sítio, equação 1.12. Dessa maneira a equação dinâmica pode ser escrita como:

$$\frac{\partial k_i}{\partial t} = \Pi(k_i)m = m\frac{k_i}{\sum_{j=1}^{N-1} k_j}. \tag{1.15}$$

Substituindo a soma por seu valor citado acima ($2mt$), a equação dinâmica se torna:

$$\frac{\partial k_i}{\partial t} = \frac{k_i}{2t}. \tag{1.16}$$

Com as condições iniciais de cada sítio i da rede, todos surgem com m ligações, isto é, $k_i(t_i) = m$, obtemos a solução:

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{1/2}. \tag{1.17}$$

Se num dado instante t temos um k maior que um dado $k_i(t)$ temos portanto um tempo inicial t_i maior que um tempo $\frac{m^2 t}{k^2}$ obtido da equação acima. Dessa maneira podemos igualar as probabilidades:

$$P[k_i(t) < k] = P\left(t_i > \frac{m^2 t}{k^2}\right). \tag{1.18}$$

Como os vértices são adicionados em intervalos iguais de tempo, a densidade de probabilidade é igual para todo valor t_i:

$$P(t_i) = \frac{1}{m_0 + t}. \tag{1.19}$$

Substituindo essa densidade na equação 1.18, isto é, a probabilidade de um tempo menor que $m^2 t/k^2$ (o tempo no qual o vértice adquire uma conectividade k), é dada por:

$$P(t_i < \frac{m^2 t}{k^2}) = \frac{m^2 t}{k^2(m_0 + t)}, \tag{1.20}$$

dessa forma,

$$P\left(t_i > \frac{m^2 t}{k^2}\right) = 1 - \frac{m^2 t}{k^2(m_0 + t)}. \tag{1.21}$$

A distribuição de conectividade é obtida com a derivada parcial da probabilidade $P[k_i(t) < k]$ em relação a variável de conectividade k:

$$P(k) = \frac{\partial P[k_i(t) < k]}{\partial k} = \frac{2m^2 t}{m_0 + t}\frac{1}{k^3}. \tag{1.22}$$

No limite assintótico, redes com um número de vértices muito grande ($t \to \infty$), obtemos:

$$P(k) \sim 2m^2 k^{-\gamma} \tag{1.23}$$

com γ igual a 3, confirmando assim as simulações numéricas. A figura 1.8 ilustra uma dessas simulações.

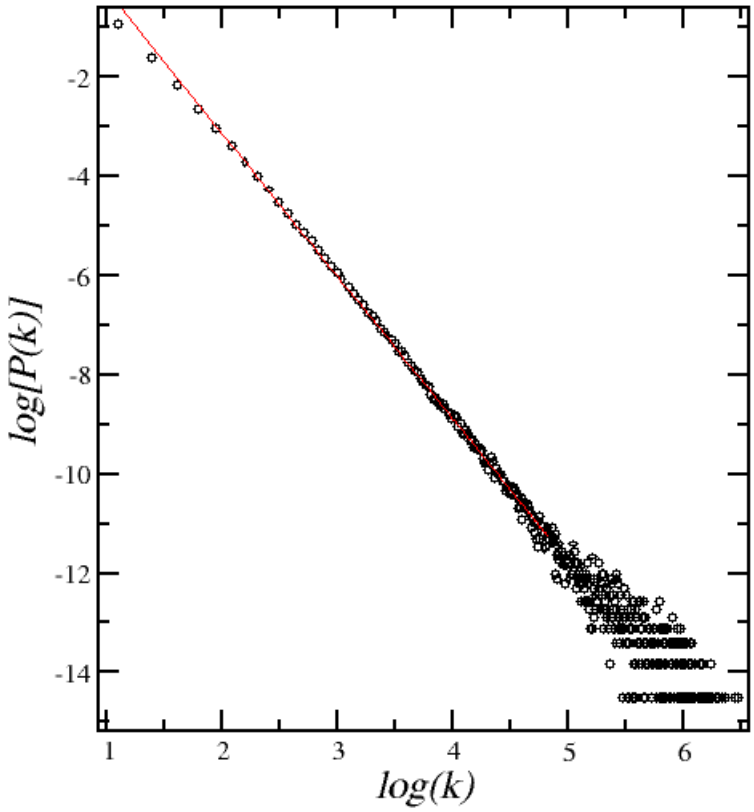


Figura 1.8: Distribuição de conectividade para uma rede de 20.000 vértices com os parâmetros $m_0 = m = 3$. O ajuste linear nos dá uma reta de inclinação $2,87 \pm 0,01$. Cada ponto é uma média sobre 100 grafos.

1.1.2 Menor caminho médio

Seja $P(\ell)$ a distribuição de menores caminhos entre sítios na rede, com ℓ a variável de caminho, temos a distância média dada por:

$$\langle \ell \rangle \equiv \sum_{\ell} \ell P(\ell). \tag{1.24}$$

Esta grandeza relaciona o menor número médio de ligações entre dois nós da rede, e é definida como a menor distância (distância geodésica) entre dois vértices. Definimos a maior dessas geodésicas como sendo o diâmetro da rede [21]. Nas redes regulares a distância geográfica acaba coincidindo com a definição de menor caminho nos grafos, nestas tal grandeza é o parâmetro de rede.

O menor caminho médio é muito útil no estudo da propagação de informações numa rede, seja na implementação de mecanismos mais eficientes de busca ou até mesmo no estudo de propagação de epidemias [27], no caso do presente estudo, danos (*bugs*) em pacotes [5].

Em geral podemos definir a distância média ao considerarmos a distância de um vértice *i* até ele mesmo como sendo:

$$\langle \ell \rangle = \frac{2}{N(N-1)} \sum_{i \geq j} d_{ij} \quad , \tag{1.25}$$

tendo em vista que $d_{ii} = 0$.

Quando é necessário contabilizar distâncias entre vértices isolados é mais conveniente utilizar a forma:

$$\left\langle \ell^{-1} \right\rangle = \frac{2}{N(N-1)} \sum_{i \geq j} d_{ij}^{-1}, \tag{1.26}$$

na qual as parcelas que contém distâncias infinitas ($d_{ij} \to \infty$) são desprezíveis.

Essa propriedade define uma das características mais recorrentes nas redes complexas, o efeito de mundo pequeno, caracterizado por um menor caminho médio muito pequeno em relação ao tamanho do sistema. Em diversas redes, incluindo as aleatórias, o número de sítios com uma dada distância *r* para um vértice central, cresce exponencialmente com *r*. Podemos

usar o fato de que numa rede grande um aglomerado local de sítios pode ser visto como uma árvore, e a cada passo, distanciando do sítio central, o número de sítios com aquela distância cresce exponencialmente por conta de bifurcações para outros sítios a cada passo. Assim a distância média crescerá com $\log n$ [21]. Esse efeito é mais acentuado nas redes livre de escala, pois o menor caminho médio não aumenta mais rápido que $\frac{\log n}{\log\log n}$ [21].

1.1.3 Coeficiente de agrupamento

Esse parâmetro estrutural caracteriza uma densidade de conexões que envolvem um agrupamento de vértices – vizinhos de um sítio i que também são vizinhos entre si. Definimos localmente esse coeficiente contabilizando apenas o sub-grafo, que é o subconjunto formado pelos primeiros vizinhos do sítio i, incluindo este e as ligações entre eles:

$$C_i \equiv \frac{2y_i}{k_i(k_i-1)} \tag{1.27}$$

onde y é o número de ligações existentes entre os vizinhos mais próximos do sítio i e k_i é o número de vizinhos mais próximos deste sítio, que coincide com a conectividade do sítio i. Em outras palavras simplesmente definimos tal densidade como a razão entre y e o número de ligações possíveis no sub-grafo do sítio i ($k_i(k_i-1)/2$). Globalmente, contabilizando todo o grafo, temos:

$$C \equiv \frac{1}{N}\sum_{i=1}^{N}\frac{2y_i}{k_i(k_i-1)} \tag{1.28}$$

O efeito de mundo-pequeno fica totalmente caracterizado após o conhecimento dessa grandeza na rede junto ao conhecimento de ℓ, quando podemos observar um alto valor de C com um baixo valor de ℓ, isto é, alto agrupamento com uma distância média entre os vértices curta, figura 1.9. O que implica que a propagação de informação na rede será mais eficiente não apenas de um ponto isolado a outro da rede e sim de uma dada região a outra.

Esse coeficiente tem valor bem definido definido nas redes regulares do tipo cadeia periódica com um número de coordenação K, como no modelo WS com $p_{WS} = 0$ (figura1.5).

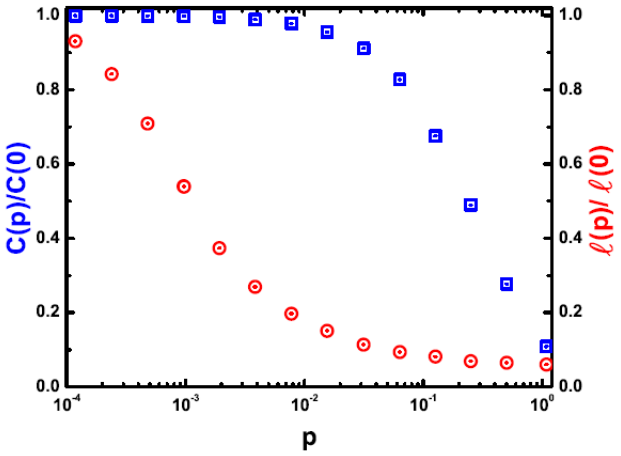


Figura 1.9: Cálculo do coeficiente de agrupamento C e do menor caminho médio l em função da probabilidade de reagrupamento (*rewiring probality*, p_{WS}), equações 1.28 e 1.25. Ilustração retirada da referência [1].

Da equação 1.28 temos:

$$C(0) = \frac{3(K-1)}{2(2K-1)}. \tag{1.29}$$

Para as redes de mundo-pequeno a probabilidade do agrupamento permanecer após o processo de reordenamento das ligações é $(1-p_{ws})^3$, e o coeficiente de agrupamento nesta rede é o coeficiente $C(0)$ ponderado por essa probabilidade. Para as redes livre de escala esse coeficiente escala com o número de sítios em $C \sim n^{0,75}$ [1].

1.2 Conclusão

Uma revisão pelos tipos mais estudados de redes foi apresentada neste capítulo. O efeito de mundo-pequeno sugere que a complexidade esteja num intervalo entre a ordem e a desordem. Porém outros fatores contribuem para que ocorra a complexidade nas redes como a estrutura modular existente nessas estruturas, como veremos no último capítulo.

As redes não-direcionadas ($\mathbf{A} = \mathbf{A}^T$) representam o marco inicial para o estudo das redes complexas. Outros efeitos podem ser observados ao usar redes direcionadas ($\mathbf{A} \neq \mathbf{A}^T$), a quebra desta simetria ocasiona por exemplo mudanças no fluxo de informação, portanto o menor caminho médio não será suficiente para descrever comportamentos globais nesta estrutura. A seguir usaremos a rede de pacotes do sistemas operacional Debian GNU/Linux como protótipo

para o estudo do caso direcionado.

2 *Características gerais da rede de dependências Debian.*

Na primeira parte deste trabalho, uma revisão a respeito das características gerais das redes complexas foi estabelecida como parte dos primeiros estudos nesse campo [1, 21]. Para o caso da rede de dependências entre pacotes Debian, temos um caso de grafos direcionados. Neste capítulo faremos o estudo de algumas propriedades desses grafos, em que faz-se necessária algumas generalizações, tais como conectividade e coeficiente de agrupamento dependendo da direção. A partir delas iremos fazer um estudo a respeito do fluxo de informação nesta estrutura como a propagação de danos em pacotes e a centralidade de entroncamento.

Ao menos dois outros trabalhos já foram publicados, utilizando alguns dos dados analisados aqui [17, 10]. No primeiro uma análise da evolução do número de pacotes durante as versões do debian é usada para elucidar o aparecimento da lei de Zipf[1] na conectividade desse sistema, por meio de um crecimento estocástico do sistema. No segundo é mostrado por meio da rede regulatória, neste caso pascotes possuem dependência e conflito, que a estrutura livre-de-escala é mais favorável para uma rede ativa do que uma aleatória.

2.1 Introdução histórica

O sistema operacional Debian GNU/Linux foi escolhido para ser o objetivo deste trabalho devido sua filosofia de licença de *software* empregada pelas pessoas que contribuem nesse projeto. Esta permite um livre acesso a todas as informações disponíveis sobre tal sistema, por

[1] Essa lei descreve a proporcionalidade inversa entre uma variável classificado por seu valor e sua frequência. A equação 1.8 enuncia essa lei com uma substituição $\gamma \to 1 + \mu$.

ser código-aberto (exceto por comentários do desenvolvedor que precisam ficar intactos), de distribuição e uso totalmente livres[2].

A primeira versão desse sistema (Debian 0.01) foi iniciada em 1993 por Ian Murdok, com o patrocínio do Projeto GNU, projeto que visava desenvolver um sistema operacional livre compatível com o sistema UNIX daí a origem de GNU, ***G****NU is* ***N****ot* ***U****NIX*. Nesta versão existiam apenas alguns pacotes primários que permitiam os usuários manipularem outros (o sistema de dependências ainda era pouco estabelecido).

Os pacotes Debian podem ser entendidos como uma união de funções que compõe o sistema operacional ou que é executado por ele, sejam bibliotecas de funções, *software* para o usuário final, compiladores, etc. Essas unidades tomaram corpo na versão 0.93 do Debian quando 200 pacotes foram enviados para essa versão. Desde então o projeto vem ampliando, tanto em pacotes quanto em desenvolvedores. A tabela 2.1 mostra a evolução desse sistema até sua versão atual.

Lançamento	versão (codinome)	desenvolvedores	pacotes
dezembro de 1996	1.2 (Rex)	120	848
julho de 1997	1.3 (Bo)	200	974
julho de 1998	2.0 (Hamm)	–	1500
março de 1999	2.1 (Slink)	400	2500
agosto de 2000	2.2 (Potato)	450	2600
julho de 2002	3.0 (Woody)	1000	9000
junho de 2005	3.1 (Sarge)	–	15400
dezembro de 2008	4.0 (Etch)	918	19170
fevereiro de 2009	5.0 (Lenny)	1018	25113

Tabela 2.1: Cronologia das distribuições Debian. Dados obtidos nas páginas `http://www.br.debian.org/devel/developers.loc`, ou observando o quórum requerido de votações em `http://www.debian.org/vote/`.

Esse breve histórico mostra o quão complexa tornou-se a estrutura da rede de dependências entre pacotes Debian, após o salto da versão Woody, está havendo uma tendência em uma estabilização por volta de 21000 pacotes na recente versão estável *Lenny* (atual versão de teste) e a versão eterna instável *Sid*. Quase a totalidade deste trabalho foi realizado durante a época

[2]Veja o contrato social do projeto Debian, `http://www.debian.org/social-contract`

em que a versão *Lenny* era a versão de teste enquanto a antiga estável *Etch* 4.0 era a distribuição oficial.

Em outubro de 2000, a fila de pacotes foi implementada, fazendo com que os diretórios que armazenam as distribuições deveriam conter apenas os arquivos *Packages* que possuem referências para a fila de pacotes. Estes arquivos contém a informação necessária para montagem da rede de dependências.

2.2 Construindo um grafo direcionado

Podemos olhar para rede de dependências como sendo um agrupamento de sítios (*pacotes*) os quais estão conectados por meio de ligações direcionadas conforme a dependência de bibliotecas ou funções específicas entre pacotes: caso seja necessária para instalação do pacote j, a instalação de i antes, então temos uma ligação e_{ji} (sentido de j para i). Um exemplo é que diversos pacotes podem depender do pacote *iceweasel*, que é um navegador de páginas *Web*, porém não à necessidade dessa dependência ser direta, um ou mais pacotes intermediários podem existir no caminho, veja figura 2.2.

A rede pode ser montada por meio da leitura da saída do comando no *shell*[3]:

```
apt-cache show $(apt-cache pkgnames).
```

No presente trabalho usamos o arquivo que é lido por esse comando unindo os tipos *main*, *contrib* e *non-free*, isto é, a rede foi formada supondo uma instalação completa do Debian GNU/Linux Etch (na maior parte das análises, usamos também a Lenny e a Sid). Tal arquivo pode ser encontrado no endereço `http://ftp.br.debian.org/debian/dists/etch/main/binary386/Packages.bz2`. O diretório `main`, pode ser trocado por `contrib` ou `non-free` a fim de obter os dados que designam os pacotes de acordo com a licença que rege o direito de uso dos mesmos: *main* são pacotes licenciados conforme a definição de *software livre* para o projeto Debian, *contrib* são pacotes livres que trazem dependências com códigos-fonte sem

[3]Terminal que oferece uma interface de entrada de dados para o usuário do sistema operacional.

uma licença de livre uso, e *non-free* são pacotes de uso e distribuição restritas.

Uma vez feita a leitura da rede, podemos verificar grandezas de interesse nesta rede: conectividade, coeficiente de agrupamento (*clustering*), centralidade que depende do menor caminho como o *betweenness* (centralidade de entroncamento) e a propagação de danos na rede (processo de contato), além de uma verificação da estrutura modular dessa rede, uma detecção de um agrupamento por afinidade de funções, isto é, comunidades.

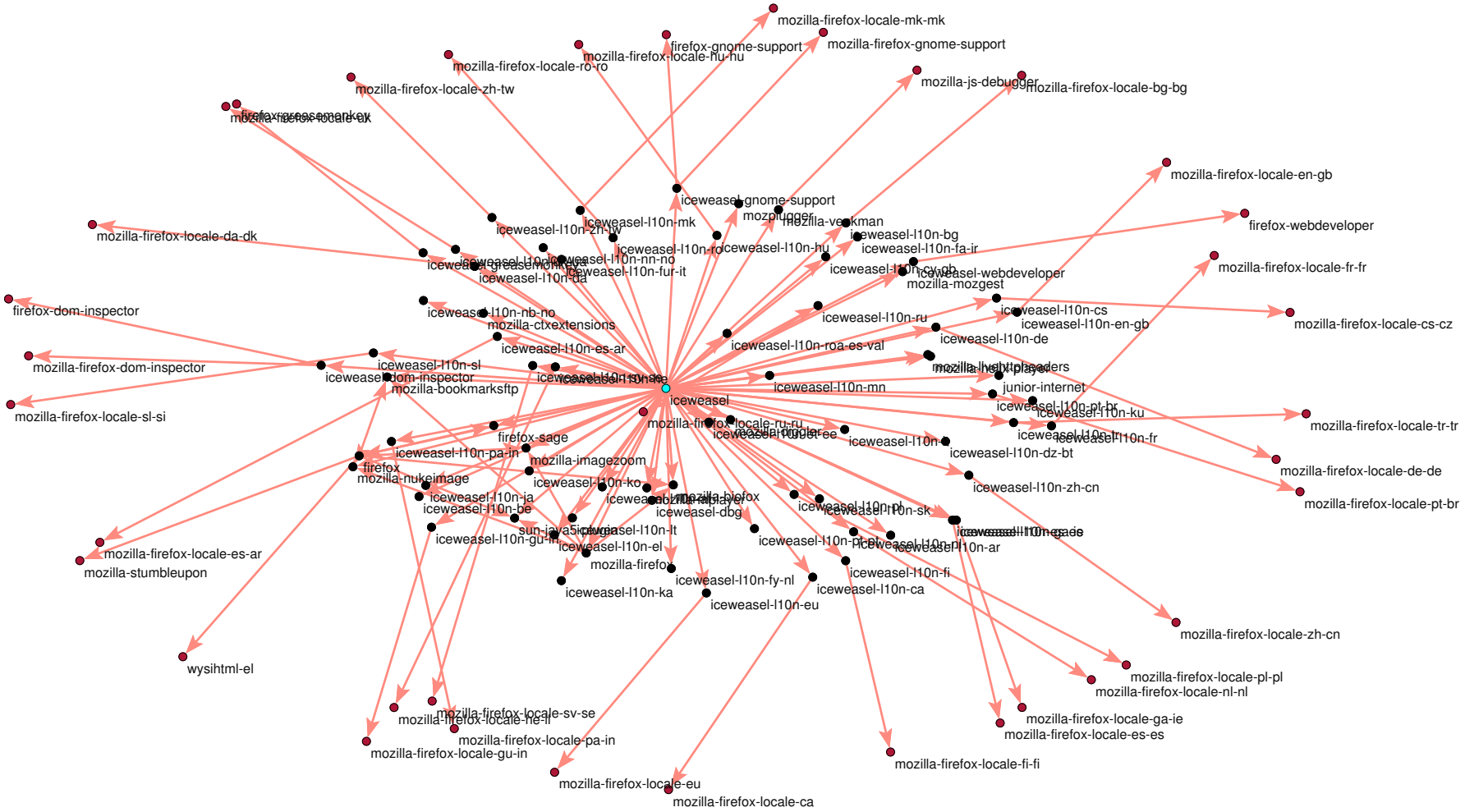


Figura 2.1: Exemplo de uma rede de dependências a partir do pacote *iceweasel*, ao centro da imagem. As cores representam o grau de dependência dos sítios com o pacote central *iceweasel*.

2.3 Conectividade e Danos

Por tratar-se de uma rede direcionada obtemos duas distribuições de conectividade: $P(k^{in})$ para ligações de entrada (grau de dependência direta) e $P(k^{out})$ para ligações de saída (grau de fundamentalidade direta):

$$k_i^{in} = \sum_{j \neq i} A_{ij} = (\mathbf{A}^T \mathbf{1})_i, \tag{2.1}$$

$$k_i^{out} = \sum_{j \neq i} A_{ij} = (\mathbf{A1})_i \tag{2.2}$$

O grau total (k_i) é a soma simples das duas grandezas acima, e podemos verificar também as auto-interações, isto é, quando existe mutuamente e_{ij} e e_{ji} [9]:

$$k_i^{\leftrightarrow} = \sum_{j \neq i} A_{ij} A_{ji} = A_{ii}^2 \tag{2.3}$$

A distribuição de conectividade desta rede mostra uma clara lei de potência para a conectividade de saída (lei de Zipf [17]), figura 2.2. Este sistemas é constituído portanto de poucos pacotes de base (necessários para instalação de uma grande parcela do sistema), e quase a totalidade dessa rede é constituída por pacotes menos fundamentais (com uma ampla lista de dependência). Podemos assim afirmar que, os pacotes fundamentais foram criados para suprir as necessidades de todos os pacotes existentes (não importa quantos sejam criados), porém a maior parte dos pacotes são limitados quanto sua funcionalidade, dando ao sistema uma escala típica, figura 2.2.

Vamos estudar como um *bug* (ou dano) de um dado pacote *i* será propagado por meio dos pacotes que dependem de *i* direta ou indiretamente. Para isso usamos o algoritmo de *busca em largura* ou de Leath com probabilidade de existência de ligações $p = 1$ (veja apêndice A) para verificar o número de pacotes "danificados", c_i, a partir de uma falha num dado pacote *i*. Portanto *c* é uma medida real do grau de fundamentalidade de um pacote.

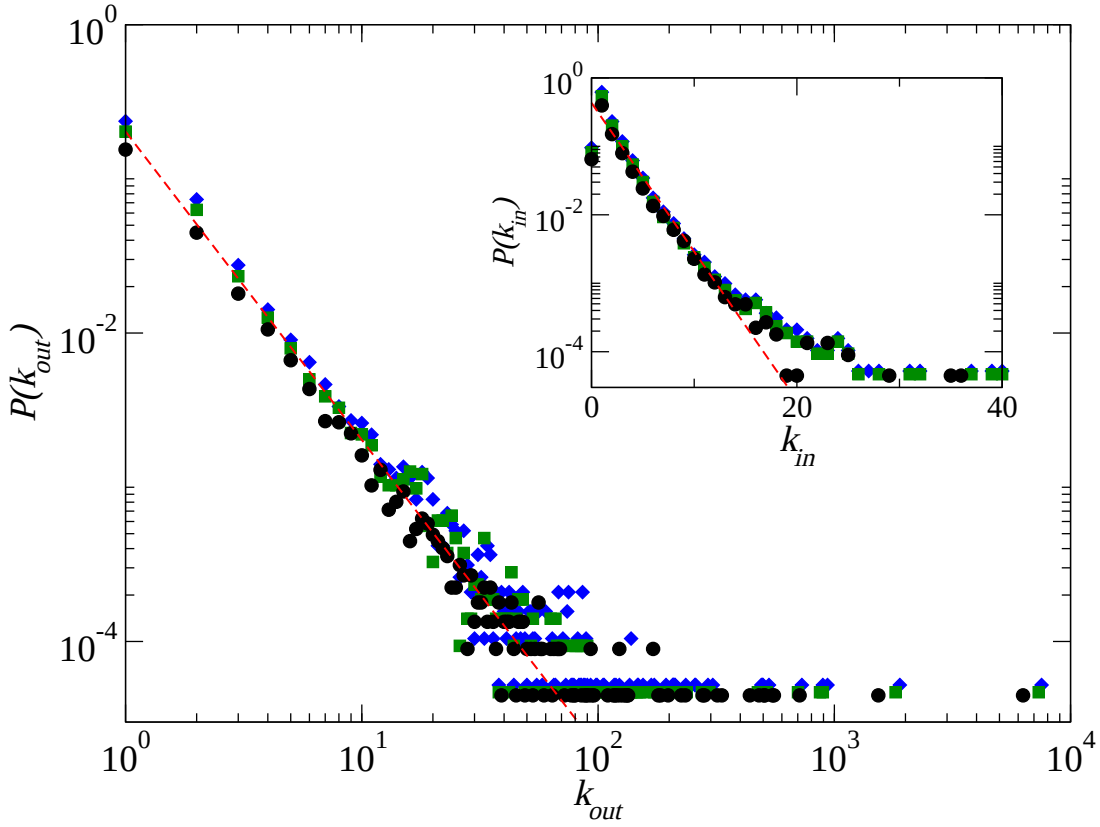


Figura 2.2: Distribuição de conectividade (k_{out}) para as versões Etch (círculos), Lenny (quadrados) e Sid (losangos) do Debian. O ajuste em lei de potência nos fornece $\gamma = -1,7 \pm 0,2$. Para o gráfico interno (k_in) o ajuste do tipo exponencial retorna $\lambda = -0,47 \pm 0,02$.

Esse processo pode ser compreendido como um processo de contato, onde existe também um período para que a infecção possa tomar totalmente a componente conexa da fonte. Esse período pode ser definido como a profundidade de propagação l da infecção.

Observamos na figura 2.3 que o efeito de tamanho finito está presente na profundidade l do dano. O sistema é livre de escalas para o número de pacotes danificados, sugerindo que o mesmo é robusto para falhas aleatórias. Porém falhas específicas podem afetar quase a totalidade nesse sistema. As redes livre-de-escala apresentam tais propriedades mencionadas acima, como discutido por Barabási e Albert em 2002 [1].

2.4 Coeficiente de Agrupamento

A definição da rede de dependências pode sugerir a construção de uma estrutura em árvore, na qual pacotes fundamentais definem ramos para instalação de pacotes finais. Porém é ob-

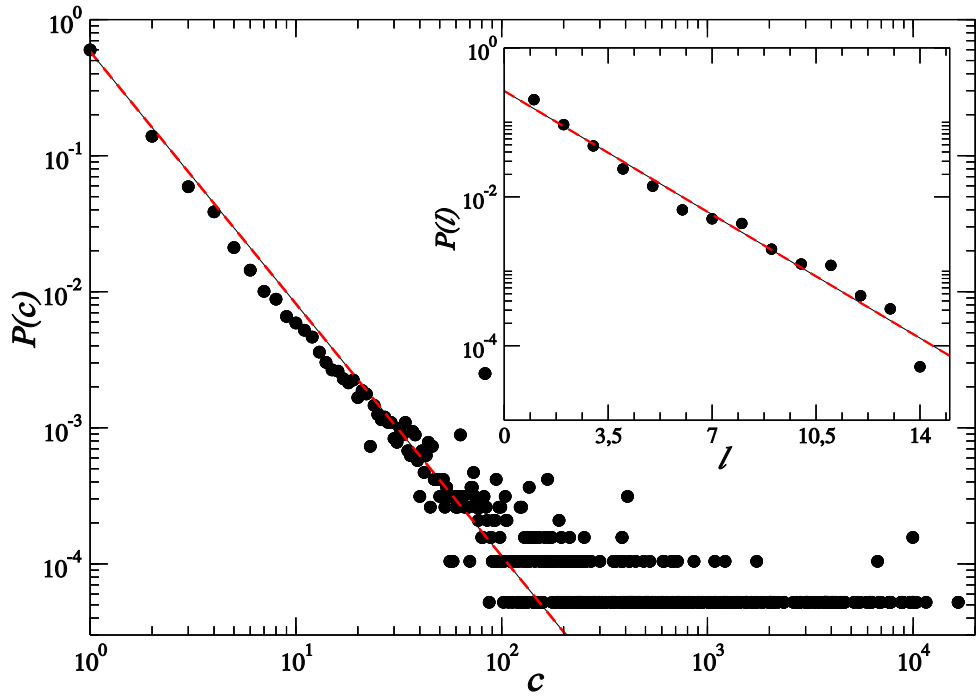


Figura 2.3: Distribuição de pacotes danificados para o Debian Etch 4.0, o ajuste em lei de potência fornece $\gamma = 1,85 \pm 0,06$. O gráfico inserido mostra a distribuição de profundidades, com um ajuste exponencial com $\lambda = 0,54 \pm 0,03$.

servado que essa estrutura se afasta de uma árvore e pode-se observar agrupamentos entre os elementos do grafo, existência de círculos entre pacotes.

Um cuidado adicional deve ser tomado para a definição de agrupamento, pois trata-se de uma rede direcionada. Observando a equação (1.27) podemos reescrevê-la com base na matriz adjacência da rede, evidenciando as definições posteriores de clustering direcionado:

$$C_i(A) = \frac{\frac{1}{2}\sum_{j\neq i}\sum_{h\neq(i,j)} A_{ij}A_{ih}A_{jh}}{\frac{1}{2}k_i(k_i-1)} = \frac{(\mathbf{A}^3)_{ii}}{k_i(k_i-1)}, \tag{2.4}$$

onde A_{ij} é um elemento da matriz adjacência $\mathbf{A}$. O produto dos três elementos, $A_{ij}A_{ih}A_{jh}$, terá valor não-nulo apenas se os sítios i, j e h forem um triângulo [9].

Para grafos direcionados, podemos definir quatro tipos de padrões de agrupamento [9]: ciclo, que define uma relação cíclica entre os vizinhos ($i \to j \to h$), *middleman*, em que um sítio i é o ponto médio para o caminho entre seus vizinhos j e h, entrada, um sítio i recebe duas

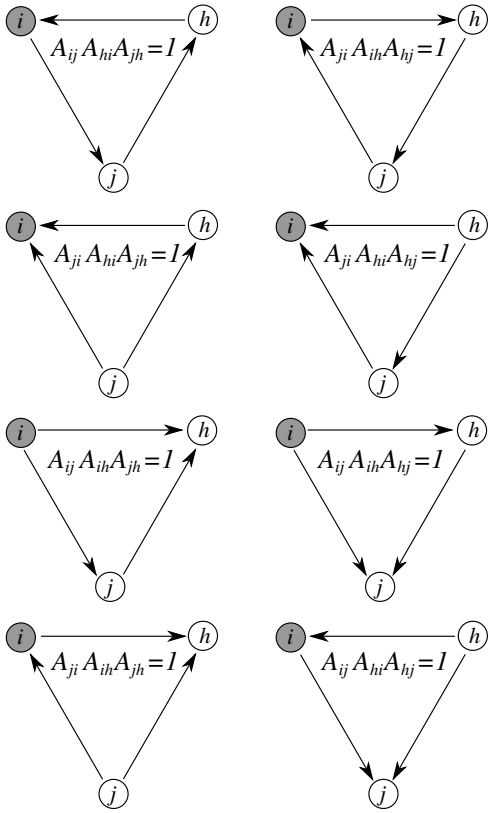


Figura 2.4: Tipos de agrupamento, de cima para baixo as duas possibilidades de cada um: ciclo, entrada, saída e *middleman*. Ilustração construída com base na referência [9].

ligações de seus vizinhos conectados, e saída, o agrupamento é formado com i levando duas ligações para dois vizinhos conectados, equação (2.4). Localmente tais agrupamentos podem ser definidos por:

$$C_i^{cyc} = \frac{(A^3)_{ii}}{k_i^{\leftarrow}k_i^{\rightarrow} - k_i^{\leftrightarrow}}, \tag{2.5}$$

$$C_i^{mid} = \frac{(AA^TA)_{ii}}{k_i^{\leftarrow}k_i^{\rightarrow} - k_i^{\leftrightarrow}}, \tag{2.6}$$

$$C_i^{in} = \frac{(A^TA^2)_{ii}}{k_i^{\leftarrow}(k_i^{\leftarrow} - 1)}, \tag{2.7}$$

$$C_i^{out} = \frac{(A^2A^T)_{ii}}{k_i^{\rightarrow}(k_i^{\rightarrow} - 1)}. \tag{2.8}$$

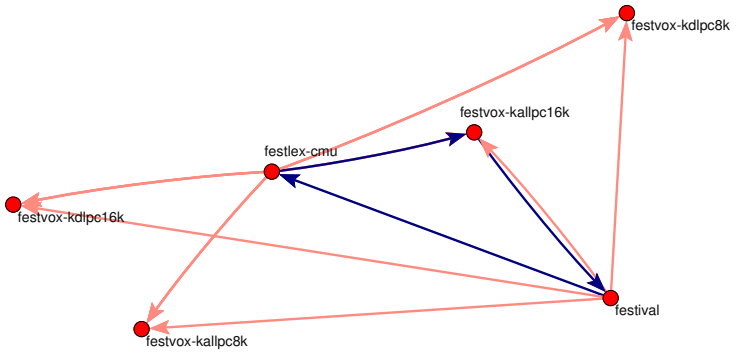


Figura 2.5: Exemplo de agrupamentos encontrados no Debian GNU/Linux Etch, em destaque um agrupamento do tipo ciclo.

Os coeficientes globais serão definidos pela média sobre todos os sítios do grafo.

$$C^* = \langle C_i^* \rangle = \frac{1}{n} \sum_{i=1}^{n} C_i^*, \tag{2.9}$$

onde o $*$ designa o tipo de agrupamento.

Esses coeficientes foram obtidos para três versões do Debian (Etch, Lenny e Sid) e comparados com os coeficientes na rede aleatória equivalente $C_{ER} = \frac{m}{n(n-1)}$ (modelo Erdös-Rényi), isto é, densidade do grafo:

Tabela 2.2: Comparação entre coeficientes de agrupamento, para cada versão e rede Erdös-Rényi equivalente. Os agrupamentos do tipo *out* e *middleman* são mais freqüentes. Triângulos do tipo *cycle* são bugs nas versões de pacotes não instaláveis, a figura 2.5 mostra um exemplo de ciclo no etch.

	C^{cyc}	C^{mid}	C^{in}	C^{out}	C	C_{ER}
Etch (antiga estável)	$1,51 \cdot 10^{-3}$	$0,247$	$9,06 \cdot 10^{-2}$	0.584	0.667	$2,90 \cdot 10^{-4}$
Lenny (teste)	$8,32 \cdot 10^{-4}$	$0,258$	$9,31 \cdot 10^{-2}$	$0,585$	$0,668$	$2,19 \cdot 10^{-4}$
Sid	$1,14 \cdot 10^{-3}$	$0,260$	$9,42 \cdot 10^{-2}$	$0,583$	0.666	$2,10 \cdot 10^{-4}$
Lenny (atual estável)	$1,34 \cdot 10^{-3}$	$0,267$	$9,76 \cdot 10^{-2}$	$0,589$	0.674	$2,20 \cdot 10^{-4}$

A estrutura básica da rede de dependências não parece ser afetada no nível global com a inserção de novos pacotes, a verificarmos a evolução da versão instável (*Sid*) até a antiga estável (*Etch*), passando pela versão de testes[4] (*Lenny*), devido a uma constância nos coeficientes de agrupamentos durante as etapas.

Podemos verificar que os tipos *out* e *middleman* são os triângulos que mais aparecem nessa

[4] A versão *Lenny* ainda era distribuição de teste no momento desses cálculos

estrutura (possuem uma densidade bem maior que o caso aleatório), isso porque existem estruturas interconectadas sendo formadas no entorno dos pacotes mais fundamentais além de outros pacotes que estão no meio do caminho conseguirem formar aglomerados. O agrupamento de entrada, torna-se pequeno (porém acima da densidade da rede) identificando o efeito de tamanho finito observado com a profundidade das falhas, figura 2.3. Dessa forma existe uma hierarquia presente nos aglomerados, a rede torna-se bastante coesa junto aos pacotes nos primeiros níveis e os grupos ficam menos densos nas distâncias finais.

2.5 Menor caminho médio e centralidade de entroncamento.

Uma medida bastante estudada nas redes sociais, definida como uma medida do potencial de comunicação de um ponto [11] é a centralidade de entroncamento, ou *betweenness*. Esse potencial pode ser calculado a partir da soma dos menores caminhos que passam por um dado vértice i, definindo $b_{ij}(k)$ como sendo o potencial comunicativo do sítio h em termos do caminho ij temos:

$$b_{ij}(k) = \frac{g_{ij}(k)}{g_{ij}}, \tag{2.10}$$

onde g_{ij} é o número de caminhos geodésicos entre os vértices i e j. Essa é portanto a probabilidade em que um ponto k pertence um caminho geodésico entre i e j.

A centralidade de entroncamento é portanto a soma dessas probabilidades:

$$C_B(k) = \sum_i \sum_{j \neq i} b_{ij}(k). \tag{2.11}$$

Para o caso de redes direcionadas deve-se levar em conta todos os possíveis caminhos, portanto a soma é apara todo par de caminho $i \neq j$ existente na rede, dado que precisamos contabilizar o potencial de comunicação entre sítios.

Para o verificar a distribuição de centralidade de entroncamento na rede foi usado o algoritmo de Brandes [4] proposto no mesmo ano que o algoritmo usado por Newman [20],

descrito no apêndice B. Verifica-se que a distribuição desta centralidade também mostra uma independência de escala com expoente $\gamma = -1.67$. Essa distribuição é mais alargada e tal independência é observada após $C_b \approx 10^{-5}$. Dessa maneira o potencial comunicativo de alguns sítios pode se estender por todo o sistema, enquanto a maior parte dos pacotes apenas são receptores de informação.

Os sítios com essa centralidade elevada podem ser entendidos como delimitadores de pacotes, eles são responsáveis por reunir diversas funcionalidade do sistema e distribuí-las ao *software* final. Isto torna também esses pacotes mais frágeis quanto à falhas no núcleo[5] do sistema, porém como essa fragilidade obedece uma lei de potência, assim ela acaba ficando escondida no próprio sistema, poucos sítios possuem centralidade de entroncamento elevada.

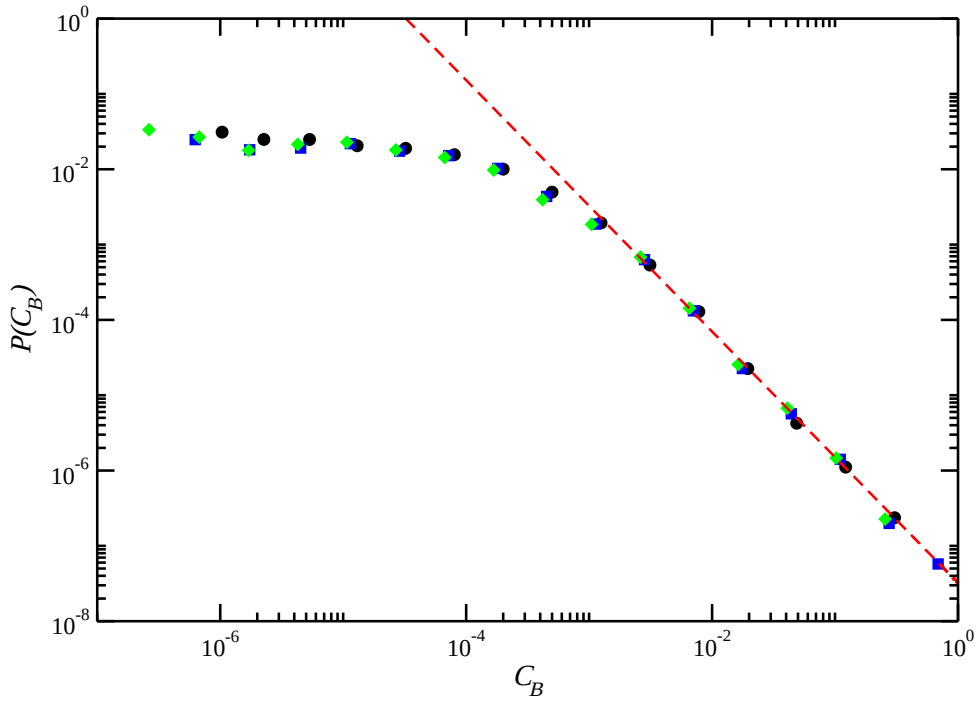


Figura 2.6: Distribuição de *betweenness* para as versões Etch (círculos), Lenny (quadrados) e Sid (losangos). Esse gráfico representa apenas cerca de 40% da rede que possui centralidade não-nula. O ajuste em lei de potência fornece $\gamma = -1.67 \pm 0.03$. O coeficiente da equação 2.11 foi normalizado pela quantidade de ligações.

[5]Entende-se aqui núcleo pelo conjunto de pacotes com as bibliotecas fundamentais e não propriamente o *kernel* do sistema operacional

2.6 Conclusão

No presente capítulo passamos a conhecer melhor a estrutura física da rede de dependências Debian, e é verificado que a estrutura não se altera consideravelmente entre as versões correntes (Etch, Lenny e Sid), tanto em conectividade(figura 2.2), agrupamentos (tabela 2.2) e *betweenness* (figura 2.6).

Como a rede permanece razoavelmente estática durante o processo de correção de falhas até chegar à versão estável, podemos sugerir uma detecção de estruturas que compõe o sistema operacional aglutinando pacotes afins, isto é, comunidades entre pacotes. O próximo capítulo empregaremos duas técnicas para que possamos verificar a existência dessas estruturas.

3 *Análise da Estrutura de Comunidades*

A estrutura de comunidades em redes complexas, isto é, a detecção de módulos (ou agrupamentos) formados de forma espontânea tem sido objeto de interesse nos últimos sete anos [12]. Na Biologia é comum esse tipo estudo em busca de estruturas padrões, *motifs*, em redes biológicas como regulação de transcrição de genes, cadeia alimentar e rede de neurônios [18]. Nestas estruturas algumas seqüências de interação são esperadas.

Já que existem sítios que detém grande parte dos caminhos para que a informação possa fluir pela rede (centralidade *betweenness* elevada) [12], a estrutura de comunidades puderam ser detectadas nas redes complexas. Este capítulo tem por objetivo fazer uma detecção de tais comunidades na rede de dependências por meio da definição de módulos proposta por Newman em 2007 [22].

3.1 Modularidade Espectral

Para a rede de dependências entre pacotes de um dado software é razoável supor uma formação de estruturas com um grau de ordenamento. A definição de Newman para verificar tais agrupamentos se dá pela definição da função de modularidade [22]:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i \cdot k_j}{2m} \right] \delta_{c_i c_j} \tag{3.1}$$

aqui observamos uma rede com m ligações, distribuídas na matriz adjacência A, onde a soma de i e j é realizada por todos os n sítios da rede, k_i é a conectividade de um dado sítio i, e $\delta_{c_i c_j}$ é o delta de Kronecker, que terá valor 1 se o rótulo c_i for igual ao rótulo c_j, ou seja, cada

comunidade recebe um valor que a rotula para que apenas as parcelas com sítios de mesma comunidade efetuem o somatório. Para o caso de matriz ponderada ($A_{ij} \rightarrow W_{ij}$), temos apenas a mudança desses elementos, e as conectividades agora são as somas dos pesos das ligações do sítio i em questão. A modularidade é definida como a diferença entre a fração de ligações das comunidades e a fração esperada de ligações de uma configuração aleatória para a qual a probabilidade de existir uma dada ligação entre os sítios i e j é dada por $\frac{k_i k_j}{2m^2}$.

A generalização da função de modularidade para grafos direcionados vem logo em seguida [15]:

$$Q = \frac{1}{m} \sum_{ij} \left[A_{ij} - \frac{k_i^{\leftarrow} k_j^{\rightarrow}}{m} \right] \delta_{c_i c_j}. \tag{3.2}$$

Define-se aqui de forma conveniente A_{ij} como sendo 1 se existir ligação de j para i e zero caso contrário. A probabilidade da configuração aleatória depende agora da capacidade do sítio i em receber ligações ($k_i^{\leftarrow}$) e do sítio j mandar ligações ($k_j^{\rightarrow}$). A detecção de comunidades é realizada por meio da maximização da função Q, isto é, escolher a distribuição de rótulos $\delta_{c_i c_j}$ que separe cada sítio em sua respectiva comunidade.

O problema pode ser solucionado de forma mais simples ao dividir a rede em duas comunidades, assim teremos $\delta_{c_i c_j} = \frac{1}{2}(s_i s_j + 1)$ então:

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i^{\leftarrow} k_j^{\rightarrow}}{m} \right] (s_i s_j + 1), \tag{3.3}$$

s_i assume o valor 1 ou -1, conforme o sítio i se encontra na comunidade 1 ou 2 respectivamente. Temos também os vínculos

$$\sum_{ij} s_i^2 = n \text{ e } \sum_{ij} A_{ij} = \sum_i k_i^{\leftarrow} = \sum_j k_j^{\rightarrow} = m, \tag{3.4}$$

assim:

$$Q = \frac{1}{2m} \left[\sum_{ij} B_{ij} s_i s_j + \sum_{ij} B_{ij} \right] = \frac{1}{2m} \mathbf{s}^T \mathbf{B} \mathbf{s}. \tag{3.5}$$

A matriz **B** possui os elementos:

$$B_{ij} = A_{ij} - \frac{k_i^{\leftarrow} k_j^{\rightarrow}}{m}. \tag{3.6}$$

No caso não-direcionado a matriz **B** é simétrica, mas podemos restaurar a simetria do problema, no caso direcionado, ao usar a transposta $\mathbf{B}^T$:

$$Q = \frac{1}{4m}\mathbf{s}^T \left(\mathbf{B}^T + \mathbf{B}\right)\mathbf{s}. \tag{3.7}$$

Neste ponto o temos um problema de autovalores, isto é, dividir a rede entre duas comunidades é obter o autovetor **v** referente ao maior autovalor da matriz $\left(\mathbf{B}^T + \mathbf{B}\right)$. O valor ótimo da função modularidade pode ser obtido fazendo uso de um multiplicador de Lagrange λ [23] para o vínculo (3.4):

$$\Lambda(\mathbf{s},\lambda) = \frac{1}{4m}\sum_{ij}\left(B_{ij} + B_{ji}\right)s_i s_j + \lambda\left(\sum_i s_i^2 - n\right), \tag{3.8}$$

calculando uma componente do gradiente de Λ que deve ser nula no estado de máxima modularidade:

$$\frac{\partial}{\partial s_k}\Lambda(\mathbf{s},\lambda) = \sum_i \left(B_{ik} + B_{ki}\right)s_i + 2\lambda s_k = 0, \tag{3.9}$$

que é válido para qualquer componente k de **s**. Este sistema de equações pode ser identificado como uma equação de autovalor ao usarmos $\beta = -2\lambda$,

$$\left(\mathbf{B}^T + \mathbf{B}\right)\mathbf{v} = \beta\mathbf{v}. \tag{3.10}$$

Desta forma o vínculo $s_i = \pm 1$ é relaxado, pois os elementos de **v** agora são reais. O vetor que precisamos (**s**) deve ser tão paralelo à **v** quanto possível, portanto podemos após a obtenção de **v** fazer $s_i = 1$ se $v_i > 0$ e $s_i = -1$ do contrário. Para elementos $v_i = 0$ os dois valores de s_i são igualmente boas soluções.

O algoritmo para encontrar todas as comunidades pode ser definido da seguinte forma:

- após a construção da matriz modularidade obtém-se seu maior autovetor e identifica o vetor **s** de componentes ± 1 mais paralelo à **v**,

- cada comunidade agora pode ser tratada como um grafo separadamente e
- repetir o mesmo procedimento até que nenhuma divisão possa mais aumentar o valor final de Q.

Nas sucessivas divisões da rede, podemos obter como maior autovalor algo do tipo $\beta_g \leqslant 0$ (autovalor do autovetor que divide o sub-grafo g), essa subdivisão não é aceita, visto que no máximo obteremos um novo módulo sem maximização de Q.

Para qualquer sub-grafo g, após a primeira divisão da rede, devemos observar que a próxima divisão implicará uma diferença total da função de modularidade Q_0 quando todos os sítios pertenciam à mesma comunidade g, e o novo valor da função de modularidade Q_1 do sub-grafo:

$$\begin{aligned}
\Delta Q &= Q_1 - Q_0 = \frac{1}{2m}\left[\sum_{i,j\in g}(B_{ij}+B_{ji})\frac{s_i s_j+1}{2} - \sum_{i,j\in g}(B_{ij}+B_{ji})\right] \\
&= \frac{1}{4m}\sum_{i,j\in g}\left[(B_{ij}+B_{ji}) - \delta_{ij}\sum_{k\in g}(B_{ij}+B_{ji})\right]s_i s_j \\
&= \frac{1}{4m}\mathbf{s}^T\left(\mathbf{B}^{(g)T}+\mathbf{B}^{(g)}\right)\mathbf{s}. \qquad (3.11)
\end{aligned}$$

Os demais elementos do grafo permanecem inalterados, assim o algoritmo pode continuar da mesma forma que a primeira divisão, com apenas uma pequena alteração da matriz:

$$B_{ij}^{(g)} = B_{ij} - \frac{1}{2}\delta_{ij}\sum_{k\in g}(B_{ik}+B_{ki}). \qquad (3.12)$$

Como apenas o maior autovalor e autovetor da equação 3.10 é de interesse, pode-se usar um método de potências para obtenção dos mesmos, vide anexo A. Nas divisões subsequentes precisamos mudar apenas a matriz $B \to B^{(g)}$.

Ao resolver o problema de autovalor, é relaxado o vínculo $s_i = \pm 1$, o passo subseqüente de melhor paralelização de $\mathbf{s}$ e relação a $\mathbf{v}$ pode carregar erros que poderão ser propagados nas próximas subdivisões. Para reduzir isso foi usado um refinamento após essa etapa visando encontrar algum valor de mudança da modularidade que maximize mais a função de modulari-

dade.

Para isso, a forma encontrada foi a introdução de uma busca aleatória pelos estados de modularidade, mudando a cada passo o valor de uma das componentes de **s** e verificando se isso causa um aumento da modularidade. Esse passo não requer muito tempo de computação visto que um bom estado inicial já foi fornecido com o método espectral.

3.2 Arrefecimento Simulado

O problema de detecção de estruturas de comunidades pode ser visto como mais um problema de otimização, desta maneira podemos usar a metodologia de Kirkpatrick et al. [13] de otimização por *Simulated Annealing* ou arrefecimento simulado.

Definindo a função custo $C(\mathbf{s})$ como sendo a função de modularidade da equação 3.1, podemos usar um parâmetro de temperatura que irá definir um estado estacionário caracterizado por uma dada distribuição $\delta_{c_i c_j}$ de rótulos de comunidades. Tal convergência ocorre por uma dinâmica de Metropolis. A dinâmica consiste usar transições microscópicas a partir de uma probabilidade p de aceitação do microestado, definida por um conjunto $V(s)$ de estados vizinhos, essa transição é nula entre estados que não pertencem à mesma vizinhança [7]. Usando a forma mais simples de Metropolis onde os conjuntos de estados vizinhos são definidos com o mesmo número de estados "q", podemos definir a matriz estocástica na forma:

$$T(s',s) = \frac{1}{q}\exp\left\{-\beta[C(s')-C(s)]\right\},\ \text{se}\, H(s') > H(s) \tag{3.13}$$

e

$$T(s',s) = \frac{1}{q},\ \text{se}\, H(s') < H(s) \tag{3.14}$$

onde C é a função custo do sistema (hamiltoniano) e β é o fator de Boltzman, proporcional ao inverso da temperatura que será regulado à a cada passo de otimização, note que neste caso a função custo é minimizada.

Tal definição segue a condição do balanço detalhado, em que o sistema possui probabilida-

des de transições, a partir de um dado estado inicial, iguais para qualquer que seja sua condição inicial, taxas de transição simétricas no tempo:

$$T(s,s')P(s') = T(s',s)P(s). \tag{3.15}$$

Portanto teremos o sistema convergindo de um estado s_0 para um estado s_l, em que l é suficientemente grande com uma probabilidade de equilíbrio $P(s_l)$, dado que qualquer estado pode ser alcançado a partir de qualquer outro.

Considerando dois estados quaisquer s_1 e s_2 tais que $C(s_1) > C(s_2)$, de acordo com as equações (3.13) e (3.14), temos as transições dadas por:

$$T(s_1,s_2) = \frac{1}{q}\exp\{-\beta[C(s_1)-C(s_2)]\} \text{ e } T(s_2,s_1) = \frac{1}{q}. \tag{3.16}$$

As probabilidades dos estados serão iguais à:

$$P(s_1) = \frac{1}{Z}\exp\{-\beta C(s_1)\} \text{ e } P(s_2) = \frac{1}{Z}\exp\{-\beta C(s_2)\} \tag{3.17}$$

com Z sendo a função de partição do sistema,

$$Z = \sum_s \exp\{-\beta C(s)\}. \tag{3.18}$$

Assim temos o balanceamento detalhado satisfeito, garantindo a existência de um estado de equilíbrio, ao calcularmos a probabilidade de ir ao estado s_2 a partir do estado s_1:

$$\begin{aligned} T(s_1,s_2)P(s_2) &= \frac{1}{q}\exp\{-\beta[C(s_1)-C(s_2)]\}\frac{1}{Z}\exp\{-\beta C(s_2)\} \\ &= T(s_1,s_1)P(s_1) \end{aligned} \tag{3.19}$$

Para a otimização da função custo $C(\mathbf{s})$, definida pela função de modularidade (3.1), usamos:

- Se $\Delta C \leqslant 0$, é calculado $p = \exp\{\beta\Delta C\}$ e geramos um número aleatório ξ uniformemente distribuído no intervalo $[0,1]$. Se $\xi \leqslant p$ então a transição ocorre.
- Se $\Delta C \geqslant 0$, o novo estado é aceito.

Aplicando essa dinâmica à simulação de arrefecimento (inicialmente a temperatura deve ser alta permitindo que o sistema convirja para um estado uniformemente aleatório), os próximos passos partem da configuração atingida no passo anterior, diminuindo a temperatura a cada passo de forma que aprisione o sistema num estado ótimo (máximo) da função custo (modularidade).

Esse método requer que o número de comunidades q seja preestabelecido, sendo que o tempo de convergência aumenta com este parâmetro.

Computacionalmente existem alguns problemas para serem supridos:

1. A ordem de grandeza do fator β_0 no passo inicial;
2. A função que regula o fator β no tempo (passo de aplicação da dinâmica de Metropolis), $\beta(t)$.

A solução adotada neste trabalho foi a escolha de β_0 como o inverso do valor da função custo com o vetor $\mathbf{s}_0$ inicial em que é definido todas as componentes (vértices da rede) na mesma comunidade. Para regular o parâmetro beta foi introduzido um novo parâmetro (γ) na simulação, numa programação de resfriamento com base em [19], seguindo uma lei de potência:

$$\beta(t) \sim \beta_0 \cdot t^{-\gamma} \tag{3.20}$$

em que γ foi ajustado de acordo aos testes entre os valores $\gamma = 1$, linear, apresenta convergência lenta mas permite maior número de estados visitados, e $\gamma = 3$ expoente de lei de potência maior faz uma convergência mais rápida sem visitar muitos estados, podendo obter uma convergência para um estado de otimização local, isto é, uma distribuição de comunidades mal sucedida.

3.3 Resultados

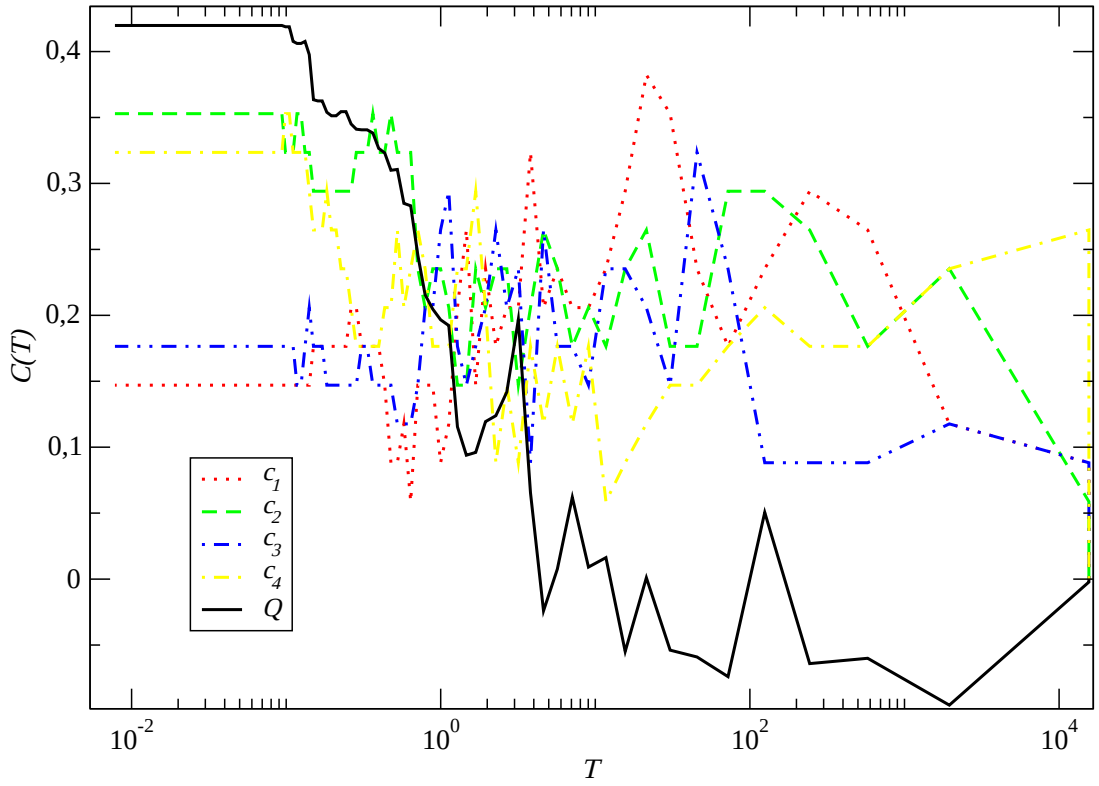
Algumas redes além da rede de dependências foram usadas para ilustração dos métodos, duas redes sociais: uma de pessoas e outra de baleias; e duas redes tecnológicas: a rede de roteadores da Universidade Federal Fluminense e a rede de dependências entre pacotes Debian.

A estrutura de comunidades dessas redes é mostrada a seguir com uma análise mais detalhada a respeito da mais populosa delas, a rede de dependências.

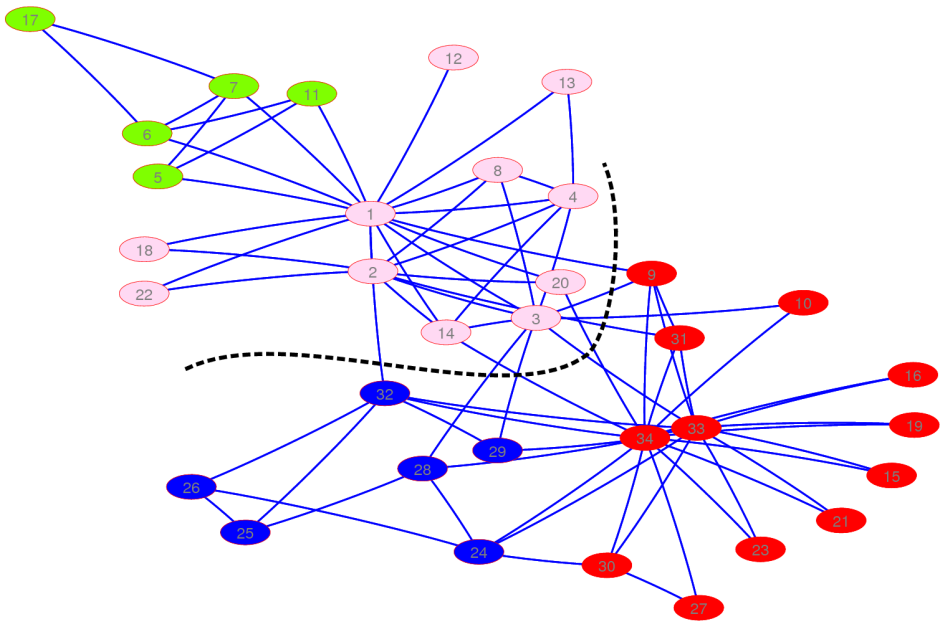
Particularmente uma rede bastante usada é a rede social do *Zachary's Karate Club* em que as ligações são determinadas conforme o vínculo de amizade entre as pessoas que compões o clube. O método de maximização da modularidade consegue uma correspondência com as facções conhecidas do clube [22], para a primeira divisão da rede. A evolução da população em cada comunidade e a modularidade é mostrada na figura 3.1(a) para o método de arrefecimento, pois um valor maior de modularidade é encontrado para quatro comunidades, apesar deste valor não diferir tanto do valor encontrado para duas comunidades (menos de 10% do valor máximo atingido com o *annealing*). Uma ilustração da divisão dessa rede não-direcionada é mostrado na figura 3.1(b).

Para a rede de orcas montada conforme observações de baleais por biólogos, as ligações entre os sítios (orcas) se dá pela freqüência em que cada exemplar é visto junto. Pode-se usar portanto a nomenclatura de rede social para esse tipo de rede. Para essa rede usamos o caso não-ponderado (apenas existe ligação conforme observados ou não dois indivíduos) e o caso ponderado conforme a freqüência de observações. A figura 3.2 mostra essa divisão. Durante a simulação de arrefecimento o mesmo comportamento qualitativo da figura 3.1(a) para a evolução da fração de indivíduos em cada comunidade é observado.

As informações a respeito das máquinas servidoras e conexões da rede de computadores da Universidade Federal Fluminense permitiram uma montagem do grafo com ligações ponderadas pela velocidade de banda (na faixa entre $64k$ e $10G$) entre os roteadores. Obtivemos resultados semelhantes na detecção das comunidades para o caso ponderado e não-ponderado, sendo que uma convergência mais lenta se deu para o caso ponderado, vide figura 3.3.

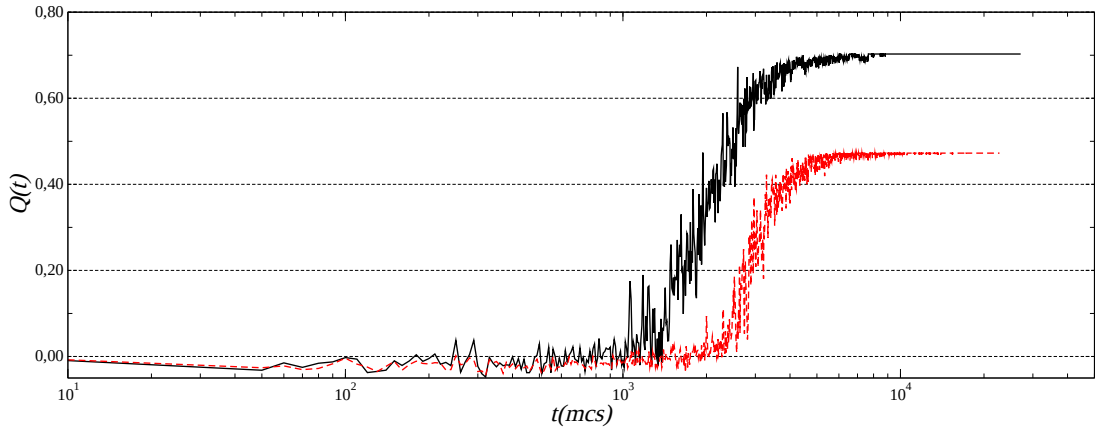


(a) Evolução da função custo para formação de quatro módulos c_i. No eixo das abcissas está a temperatura de simulação, que decai até um estado de modularidade constante.

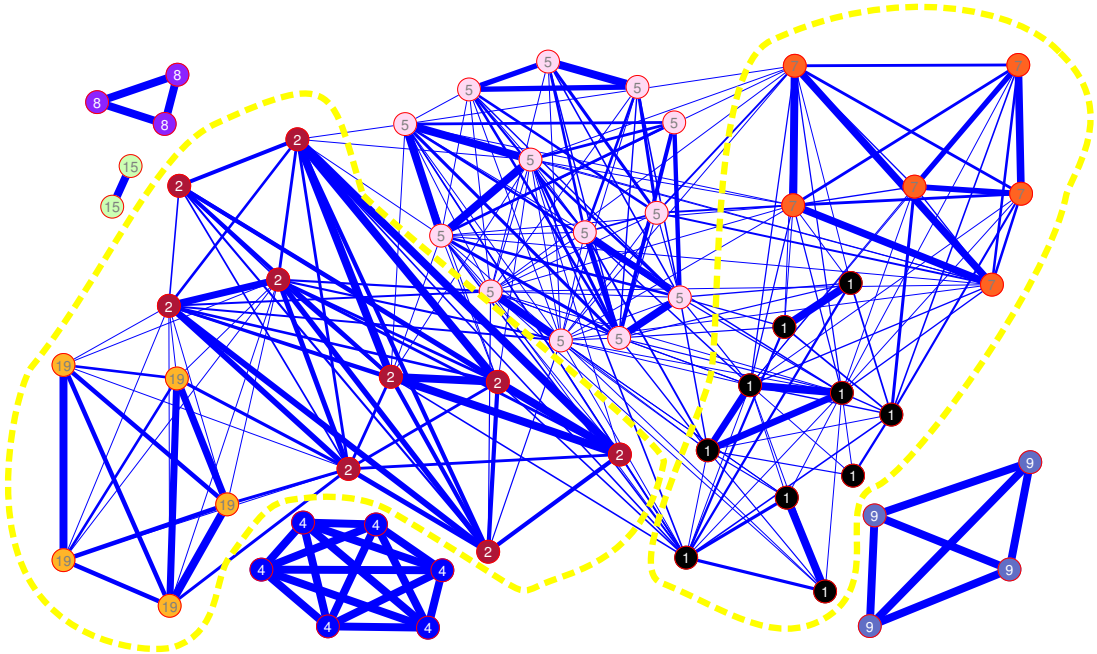


(b) Divisão em comunidades pela maximização da função modularidade, equação 3.1. A linha pontilhada corresponde à primeira subdivisão pelo método espectral.

Figura 3.1: Rede social *Zachary's Karate Club*.

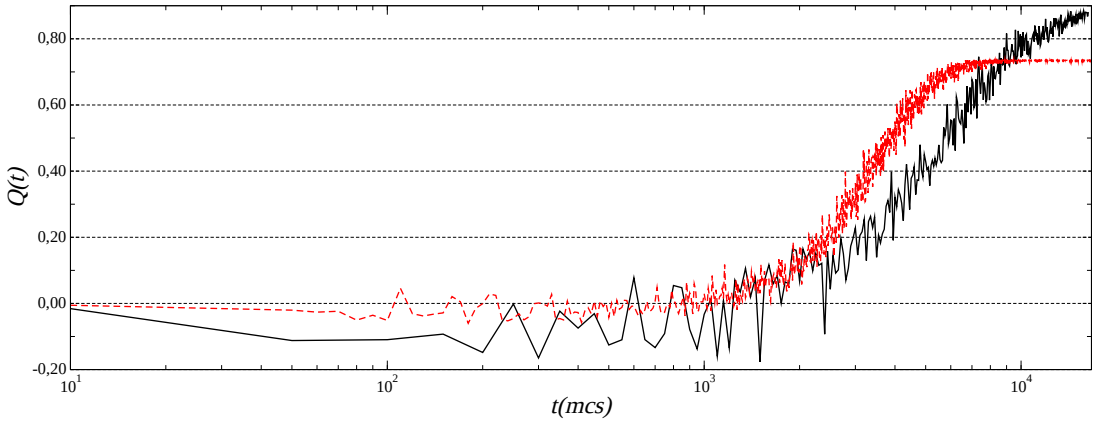


(a) Evolução da função custo para rede de orcas. No eixo das abcissas está a quantidade de passos Monte-Carlo (*mcs*) para que o resfriamento da temperatura alcançasse um estado de máxima modularidade. O caso ponderado (linha contínua) resulta numa resolução maior que o caso não-ponderado (linha pontilhada).

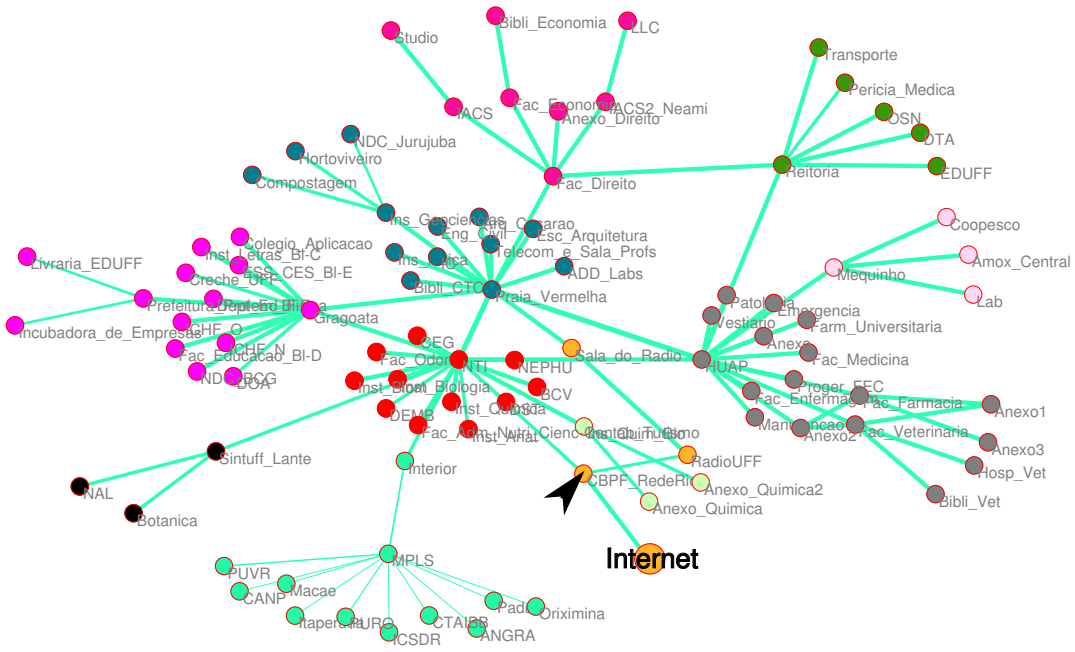


(b) Divisão da rede social de orcas em comunidades pela maximização da função modularidade, equação 3.1. A linha pontilhada corresponde às comunidades separadas sem levar em conta o peso das ligações, representados pela espessura da linha.

Figura 3.2: Rede social dos baleais.



(a) Evolução da função custo. No eixo das abcissas está a quantidade de passos Monte-Carlo (*mcs*) para que o resfriamento da temperatura alcançasse um estado de máxima modularidade. O caso ponderado (linha contínua) possui uma convergência mais lenta que no caso não-ponderado (linha pontilhada), ambos os casos fornecem divisão equivalente.



(b) Divisão de roteadores em comunidades pela maximização da função modularidade, equação 3.1. A seta indica o roteador que fornece o acesso à *internet* para toda a rede. A espessura das ligações estão na escala do intervalo 64k–10G.

Figura 3.3: Rede tecnológica dos roteadores da UFF.

3.3.1 Estrutura de comunidades na rede de dependências Debian

Nas redes anteriores as estruturas se resumiam sempre em algumas dezenas de sítios sendo a mais densa o grafo das orcas com 58 baleias e 278 pares de avistamento. Para a rede de dependências do Debian as escalas são bem maiores; até o mês de novembro de 2008 as componentes gigantes das distribuições analisadas possuíam: 17543 sítios e 91192 ligações para a estável; 21361 sítios e 100969 ligações para a versão teste; e 22369 sítios e 106942 ligações para a instável.

Esses números proporcionam grandes problemas computacionais antes não ocorridos, visto que para redes pequenas o método de arrefecimento simulado mostrou-se tão veloz quanto o método espectral para os microcomputadores atuais, menos de um minuto para tais redes na ordem de dezenas de sítios. A primeira dificuldade encontrada foi o tamanho do espaço de fase para obter a distribuição de rótulos de comunidades, pois, para redes grandes e coeficiente de clustering baixo, imagina-se um número grande de comunidades. Com isso o annealing torna-se um método bastante custoso e sujeito à diversas tentativas de reaquecimento para obtenção de um novo estado de modularidade mais alta.

Outra dificuldade concernente aos dois métodos, foi o armazenamento da matriz modularidade **B** (equação 3.6), faz-se necessário o uso de computadores com memória de acesso aleatório (*RAM*[1]) de pelo menos $2Gb$.

O método espectral mostrou-se o mais eficiente para tratar sistemas desse tamanho, visto que o problema de acesso aos dados armazenados diminui exponencialmente a cada subdivisão – apenas é necessária a montagem da submatriz $\mathbf{B}^{(g)}$ da equação 3.12 a cada passo de divisão. Além dos resultados numéricos obtidos nas simulações desse trabalho, o método espectral retornou valor da modularidade maior que no método de arrefecimento.

A importância do *simulated annealing* para esse tipo de rede ocorre no passo de refinamento do método espectral, além de oferecer a possibilidade de refinamento da saída espectral, fornecendo como chute inicial o vetor obtido pelos consecutivos autovalores das matrizes $\mathbf{B}^{(g)}$.

[1] *Abreviação do inglês Random Acess Memory.*

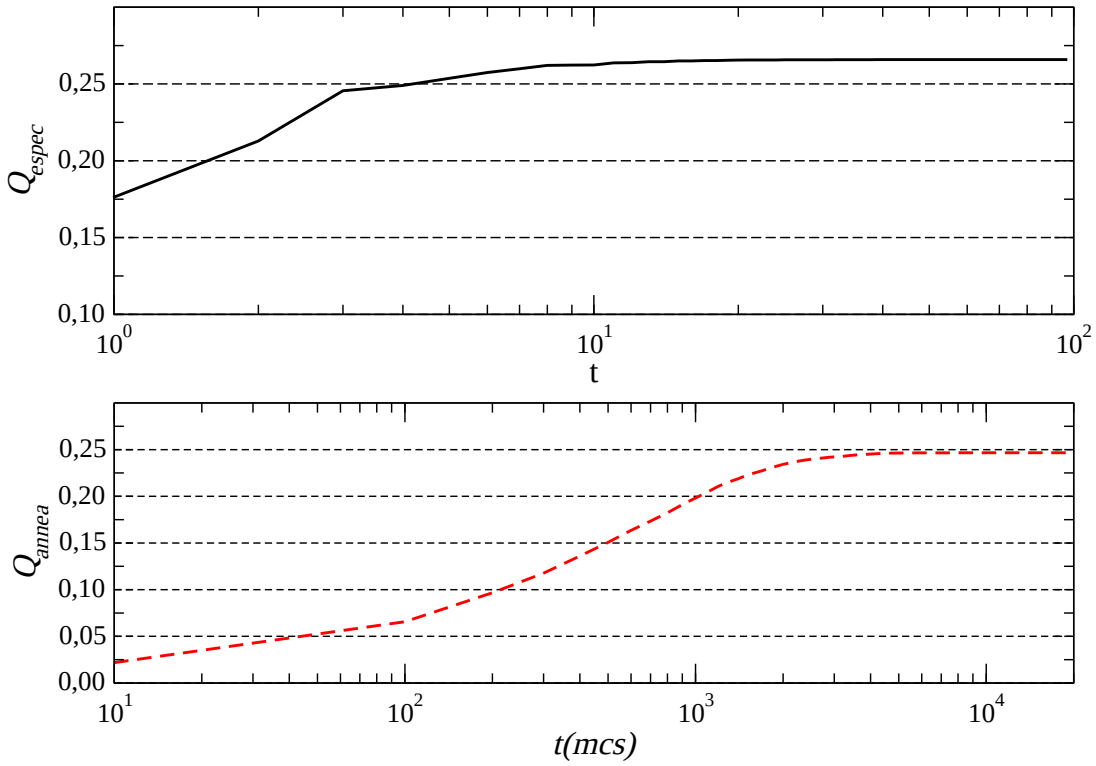


Figura 3.4: Comparando a eficiência dos algoritmos com a função de modularidade. Cerca de oito horas foram necessárias para completa divisão da rede no método espectral (gráfico superior), porém os passos (t) não são iguais. Em média cinco dias foram necessários para uma convergência no método de arrefecimento (gráfico inferior), os passos Monte-Carlo (*mcs*) gastam o mesmo tempo. O microcomputador usado nesta análise foi um modelo *AMD Athlon(tm) 64 X2 4200+* com CPU 2.2*GHz* e memória RAM de 2*Gb*.

Neste trabalho tal refinamento não foi obtido, visto que os métodos já retornaram principais comunidades bem coerentes.

O teste melhor sucedido para simulações de arrefecimento, com maior valor de modularidade, foi para um número total de comunidades igual à 400, o algoritmo acabou prendendo o sistema num número menor de comunidades. Ao observarmos a evolução da função Q para o caso espectral, em passos de subdivisão, fica ainda mais visível que o número de comunidades é inferior à 400, esse tipo de cálculo encerrou o algoritmo em 98 módulos, a comparação é ilustrada na figura 3.4.

A evolução do método espectral pode ser melhor entendida ao observarmos o dendrograma do processo de divisão, em que os rótulos de comunidade foram dados conforme o nome do pacote com maior grau de conectividade de saída, figura 3.5.

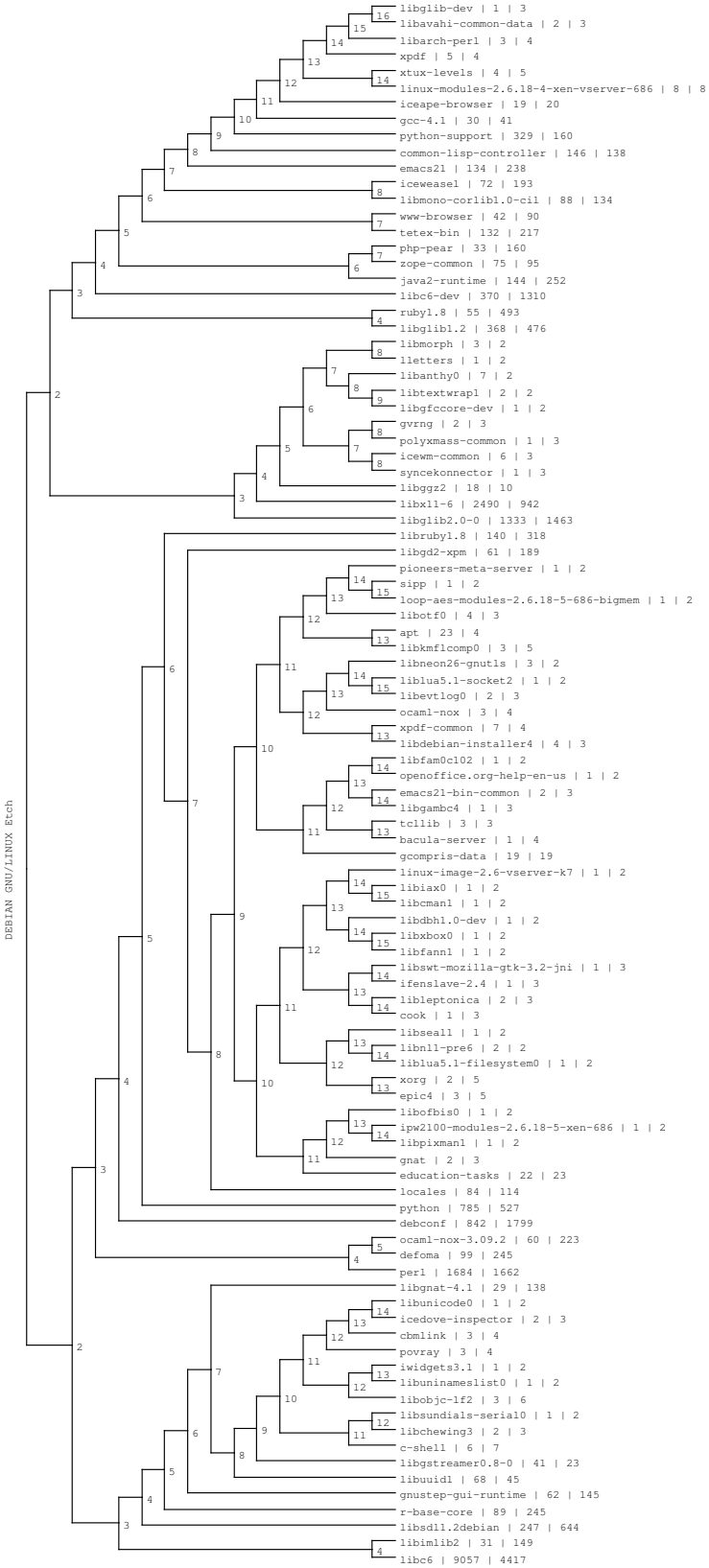


Figura 3.5: Dendrograma com a evolução das divisões de comunidades do Debian Etch, os rótulos das bifurcações designam o nível de divisão. O rótulo das comunidades foram escolhidos como os sítios mais fundamentais, sua conectividade e o tamanho do agrupamento.

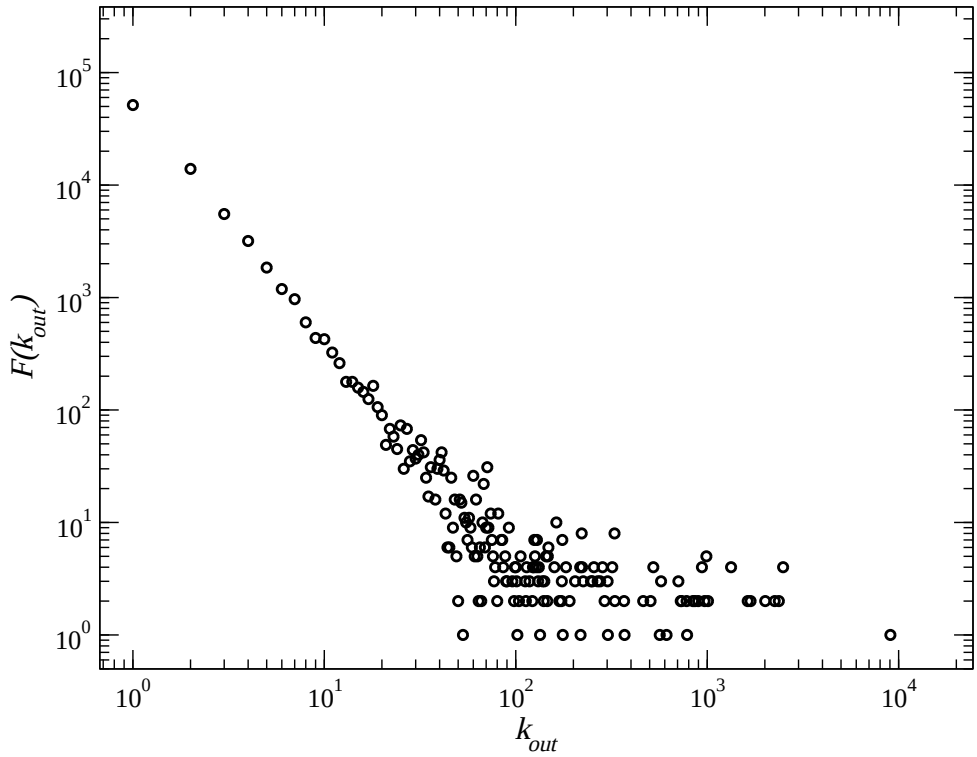


Figura 3.6: Frequência de mudança (F) de estado entre os sítios da rede de dependências a partir de sua conectividade de saída (k^{out}).

Para uma visualização mais apurada dos principais pacotes foi medida a freqüência (F) na qual cada sítio mudou o valor de seu rótulo durante a simulação de arrefecimento. Observa-se que a conectividade de saída, equação 2.2 está bem correlacionada com tal freqüência, figura 3.6. Isso ressalta a concordância entre as principais comunidades obtidas entre os métodos descritos acima. A tabela 3.1 mostra os dados para as principais comunidades obtidas com o *annealing*.

Observando-se a separação das comunidades pelos dois métodos é possível definir estratégias de organização para os times de desenvolvedores Debian. Citando os pacotes fundamentais das comunidades mais povoadas e detectadas em níveis baixos no método espectral, figura 3.5; que estão de acordo àquelas detectadas com o método de arrefecimento.

O *libc6* como esperado – praticamente todos os programas dependem deste pacote – define a principal comunidade da rede, pacotes mais dependentes das bibliotecas padrões da linguagem C. O *libc6-dev* formou uma segunda comunidade no mesmo nível que a *libc6*, mostrando que os recursos de desenvolvimento são pontos bastante relevantes nesse sistema, pois tal biblioteca

armazena os recursos necessários para compilar programas que usem a biblioteca padrão C.

Tecnologias de desenvolvimento posteriores ao C ganham corpo também nesse sistema, observando a comunidades definidas pelos pacotes *libglib*, biblioteca gráfica que proporciona mais flexibilidade à linguagem C, e as linguagens *perl* e *python* linguagens interpretadas bastante flexíveis.

Pacotes de gerenciamento interno do sistema como o *debconf* e *common-lisp-controller* formam estruturas bem definidas, visto que o primeiro possui um nível baixo de detecção espectral e este último não sofreu mudança de comunidade na simulação de arrefecimento.

O gerenciamento gráfico de janelas também é uma peça marcante de um sistema operacional, na simulação de arrefecimento acaba por enaltecer esse recurso disponibilizado pela biblioteca *libx11-6*, sendo destacada também no método espectral.

3.4 Conclusão

O método de maximização da função de modularidade confirma a existência de agrupamentos que surgem nessas estruturas, visto que as redes sociais mostram grupos bem definidos, tando para o clube de karatê quanto para os baleais. Esse tipo de estudo portanto pode ser útil em sistemas do mesmo tipo em escalas maiores.

A detecção de comunidades em sistemas tecnológicos também abre uma grande margem de aplicações. Para a rede UFF a detecção desses módulos poderá ser de grande utilidade no uso da administração e planejamento da rede, visto que a função modularidade leva em conta a propagação de informação, e esse modelo configuracional pode ser alterado para detecção de estruturas que não necessariamente se afastem do aleatório, segundo termo da equação 3.1.

No caso de sistemas grandes, como foi o caso de estudo do presente capítulo, podemos mostrar um passo posterior à criação de pacotes que seriam os diversos núcleos formados no sistema operacional por conta dessa dinâmica de programação por dependências. Nesse método

Tabela 3.1: Tamanho das principais comunidades (N) separadas usando o *simullated annealing* para maximização da função de modularidade; e frequência de mudança de estado (f) dos pacotes fundamentais, em relação ao pacote que mais mudou durante a simulação, *toolbar-fancy*, 98 vezes.

N (%)	pacote	k_{out}	f (%)
13.9	libx11-6	2490	0,04
10.2	libc6	9057	0,01
9.2	perl	1684	0,02
9.2	debconf	842	0,02
7.7	libglib2.0-0	1333	0,04
4.7	python	785	0,01
3.1	libglib1.2	368	0,02
2.6	libsdl1.2debian	247	0,03
2.5	libc6-dev	370	0,01
2.0	apache	86	0,02
2.0	libx11-dev	90	0,01
1.9	libncurses5	613	0,01
1.7	libglib2.0-dev	88	0,02
1.5	r-base-core	89	0,03
1.4	java2-runtime	144	0,02
1.4	tetex-bin	132	0,04
1.1	tcl8.4	112	0,03
1.0	libgl1	218	0,01
1.0	emacs21	134	0,01
0.9	common-lisp-controller	146	0
0.8	x11-common	175	0,06
0.7	libgnustep-base1.13	74	0,06
0.7	iceweasel	72	0,04
0.6	libmono-corlib1.0-cil	88	0,03
0.6	xfonts-utils	92	0,04
18.0	demais pacotes	–	–

a conectividade de saída acabou sendo o parâmetro que melhor define o sítio dentro das comunidades, figura 3.6, visto que tais estruturas possuem os sítios mais conectados como vértices centrais.

Os métodos de separação em comunidades definiram as principais estruturas como sendo as responsáveis por fornecer os recursos padrões da principal linguagem que o sistema foi escrito, recursos de desenvolvimento, flexibilidade, gerenciamento interno e gerenciamento gráfico. Ao observar-mos as comunidades de nível mais baixo do método espectral, figura 3.5 e as mais populosas obtidas com o método de arrefeciemento, tabela 3.1.

4 Conclusão

O presente estudo oferece uma base ao entendimento da formação de estruturas modulares em grandes sistemas, passando pela análise das características básicas da estrutura de rede do sistema operacional Debian GNU/Linux.

Observa-se uma estrutura livre-de-escala em termos da precedência de instalação dos pacotes, com expoente de lei de potência $\gamma_k = 1,7$, e $\gamma_c = 1,85$ para a distribuição de pacotes danificados, indicando a robustez do sistema para falhas aleatórias, que em média não se propagaram por níveis muito profundos na estrutura, veja figura inserida 2.3.

A fragilidade do sistema medida por meio da centralidade de entroncamento, mostra que os pacotes mais vulneráveis (centralidade alta, portanto dependentes de diversos recursos) são menos populosos, definindo uma estrutura bem heterogênea.

É possível observar ainda agrupamentos de dependência mútua, que seriam falhas no sistema de dependências, pacotes com informações de instalação ambígua. Observado que o coeficiente de agrupamento do tipo ciclo ainda é superior ao caso aleatório. Nesse tipo de estrutura temos os aglomerados de pacotes definidos em torno de pacotes básico, dado o alto coeficiente de agrupamento do tipo saída.

A otimização da função modularidade mostrou-se eficiente para a classificação social de indivíduos, como no caso do clube de karatê e dos baleais, ainda é possível usar uma hierarquização da estrutura conforme a evolução da detecção, dendrograma da figura 3.5.

Para o tratamento de comunidades, observa-se que diversas estruturas surgem em torno de pacotes básicos, sendo que os principais elementos para o caso do Debian GNU/Linux são

tecnologias de desenvolvimento C padrão, flexibilizações dessa tecnologia (*perl*, *glib*, etc.) e gerenciamento interno e de janelas.

APÊNDICE A – Busca em Largura

O *Breadth First Search* (*BFS*), ou busca em largura, permite o exame de todos os vértices do grafo, definindo qualquer componenete totalmente conectada, daí seu uso para a verificação dos danos, bem como cálculo de menor caminho entre sítios, servindo de base para o algoritmo de *betweenness*.

O *BFS* trabalha usando uma busca exaustiva por todos os sítios alcançáveis a partir de um sítio raíz *i*. O pseudo-código ilustra como ocorre essa busca:

início
    $rotulo[i] \leftarrow 0, i \in V$;
    $Q \leftarrow$ fila vazia;
    insere $i \rightarrow Q$;
    enquanto *Q não-vazia* **faça**
        retira $v \leftarrow Q$;
        para cada *z vizinho de v* **faça**
            se $rotulo[z] \neq 0$ **então**
                insere $z \rightarrow Q$;
                $rotulo[z] \leftarrow 1$;
            fim
        fim
        $rotulo[v] \leftarrow 2$
    fim
fim

Algoritmo 1: Busca em Largura

Percebe-se que os sítios são percorridos portanto em ordem crescente de distância partindo do sítio raiz. É possível detectar todas as componentes conexas da rede ao empregar tal algoritmo por todos os sítios da rede que não possuem rótulos 1 ou 2 ainda, bem como determinar o menor caminho entre todos os pares de sítios da rede, usando a atribuição de que a distância será infinita para sítios que não pertencem à mesma componente conexa.

Uma aplicação do BFS muito usada nos problemas de percolação foi o algoritmo criado por P. L. Leath em 1976 [14]. Usa-se basicamente o BFS para verificar se um cluster de percolação foi formado durante uma simulação. A ilustração mais básica desse uso é a percolação numa rede quadrada, onde após definir a estrutura da rede com condições de contorno periódicas o cluster é formado com uma busca em largura a partir de um sítio central (distante da periferia da rede), marcando os sítios como existente conforme uma probabilidade p, parâmetro de ordem no problema de percolação:

início
  $rotulo[i] \leftarrow 0, i \in V$;
  $existe[i] \leftarrow 0, i \in V$;
  $Q \leftarrow$ fila vazia;
  insere $i \rightarrow Q$;
  enquanto *Q não-vazia* ou *percolação concluída* **faça**
    retira $v \leftarrow Q$;
    para cada *z vizinho de v* **faça**
      se $rotulo[z] \neq 0$ e $existe[z] \neq 0$ **então**
        insere $z \rightarrow Q$;
        $rotulo[z] \leftarrow 1$;
        $n \leftarrow$ número aleatório gerado a partir de uma distribuição uniforme;
        se $n < p$ **então** $existe[z] = 1$;
      fim
    fim
    $rotulo[v] \leftarrow 2$
  fim
fim

Algoritmo 2: Formando um possível *cluster* de percolação.

A condição de percolação geralmente pode ser tomada verdadeira quando sítios de lados opostos passarem a existir por intermédio dos sítios mais internos, isto é, ultrapassar o comprimento L da rede.

Tal algoritmo é bastante usado para redes em que a probabilidade crítica de poercolação p_c é conhecida, podendo assim gerar facilmente configurações de cluster percolante próximas ao ponto fixo p_c.

O processo de contato [16] também pode ser estudado com essa ferramenta, entendendo o crescimento de cluster como a propagação de uma infecção no tempo a partir de um único indivíduo. O caso da medida de danos, deste trabalho, segue essa mesma interpretação, emprega-

se tal algoritmo com uma probabilidade de percolação igual à 1, essa medida é portanto a quantidade de valores 1 armazenados no vetor *existe*.

APÊNDICE B – Cálculo da centralidade de entroncamento

Para o cálculo dessa centralidade será necessário o espaço de armazenamento $\mathcal{O}(n+m)$, armazenando sítios e ligações, e o tempo de processamento será de $\mathcal{O}(n \cdot m)$, pois todos os caminhos serão percorridos a cada vértice. O pseudocódigo ilustra bem a definição usada no capítulo 1:

De forma mais simples, esse algoritmo irá calcular todos os potenciais comunicativos $b_{ij}(k)$ por meio do vetor $\sigma[k]$ a partir de todas geodésicas (num total de *m* ligações) que partem de uma fonte *s* pertencente à rede (*n* sítios), execução no tempo $\mathcal{O}(m \cdot n)$. Após percorrer todos possíveis menores caminhos de *s* a soma de $C_B(k)$ é implementada fazendo o caminho inverso.

início
  $C_B(v) \leftarrow 0, v \in V$;
  para cada $s \in V$ **faça**
    $S \leftarrow$ pilha vazia;
    $P[w] \leftarrow$ lista vazia, $w \in V$;
    $\sigma[t] \leftarrow 0, t \in V$; $\sigma[s] \leftarrow 1$;
    $d[t] \leftarrow -1, t \in V$; $d[s] \leftarrow 0$;
    $Q \leftarrow$ fila vazia;
    insere $s \rightarrow Q$;
    enquanto *Q não-vazia* **faça**
      sai da fila $v \leftarrow Q$;
      entra na pilha $v \rightarrow S$;
      para cada *vizinho w de v* **faça**
        // *w é encontrado a primeira vez*
        se $d[w] < 0$ **então**
          entra na fila $w \rightarrow Q$;
          $d[w] \leftarrow d[v] + 1$;
        fim
        // *o menor caminho para w via v é*
        se $d[w] = d[v] + 1$ **então**
          $\sigma[w] \leftarrow \sigma[w] + \sigma[v]$;
          entra na lista $v \rightarrow P[w]$;
        fim
      fim
    fim
    $\delta[v] \leftarrow 0, v \in V$;
    // S retorna os sítios em ordem decrescente do tamanho do caminho partindo de s. **enquanto** S **faça**
      extrai da pilha $w \leftarrow S$;
      para $v \in P[w]$ **faça** $\delta[v] \leftarrow \delta[v] + \frac{\sigma[v]}{\sigma[w]} \cdot (1 + \delta[w])$ **se** $w \neq s$ **então**
      $C_B[w] \leftarrow C_B[w] + \delta[w]$
    fim
  fim
fim

Algoritmo 3: Centralidade de entroncamento para grafos sem peso nas ligações [4]

ANEXO A – Método de potências

Esse é um método iterativo para obter um par de autovalor e autovetor, baseado no teorema A.0.1:

Teorema A.0.1. *Assumindo que a matriz $n \times n$ $\mathbf{A}$ tem n autovalores distintos λ_1, λ_2, ..., λ_n, em ordem decrescente de magnitude:*

$$|\lambda_1| > |\lambda_2| \geqslant |\lambda_3| \geqslant ... \geqslant |\lambda_n| \tag{A.1}$$

Se $\mathbf{X}_0$ é esolhido apropriadamente, então a seqüência $\{\mathbf{X}_k = [x_1^{(k)} x_2^{(k)} ... x_n^{(k)}]\}$ e $\{c_k\}$ gerada recursivamente por

$$Y_k = \mathbf{A}\mathbf{X}_k \tag{A.2}$$

e

$$\mathbf{X}_{k+1} = \frac{1}{c_{k+1}}\mathbf{Y}_k, \tag{A.3}$$

onde

$$c_{k+1} = x_j^{k+1} \, e \, x_j^{(k)} = \max_{1 \leqslant i \leqslant n} \{|x_i^{(k)}|\}, \tag{A.4}$$

irá convergir para o autovetor dominante $\mathbf{V}_1$ e autovalor λ_1, respectivamente. Isto é:

$$\lim_{k \to \infty} \mathbf{X}_k = \mathbf{V}_1 \, e \, \lim_{k \to \infty} c_k = \lambda_1. \tag{A.5}$$

Prova. Desde que $\mathbf{A}$ tenha n autovalores, existem então n correspondentes autovetores (caso não-degenerado) $\mathbf{V}_j$, com $j = 1, 2, ..., n$. O vetor inicial $\mathbf{X}_0$ pode ser escrito como uma

combinação linear:

$$\mathbf{X}_0 = b_1\mathbf{V}_1 + b_2\mathbf{V}_2 + ... + b_n\mathbf{V}_n. \tag{A.6}$$

Assumindo que o vetor $\mathbf{X}_0 = [x_1\, x_2\, ...\, x_n]$ foi escolhido de maneira que $b_1 \neq 0$. Também assume-se que as coordenandas de $\mathbf{X}_0$ são escaladas de forma que $\max_{1\leqslant j\leqslant n}\{|x_j|\} = 1$. Podemos encontrar o vetor $\mathbf{AX}_0$ por um processo de normalização

$$\begin{aligned}
\mathbf{Y}_0 &= \mathbf{AX}_0 = \mathbf{A}(b_1\mathbf{V}_1 + b_2\mathbf{V}_2 + ... + b_n\mathbf{V}_n) \\
&= b_1\mathbf{AV}_1 + b_2\mathbf{AV}_2 + ... + b_n\mathbf{AV}_n \\
&= b_1\lambda\mathbf{V}_1 + b_2\lambda_2\mathbf{V}_2 + ... + b_n\lambda_n\mathbf{V}_n \\
&= \lambda_1\left(b_1\mathbf{V}_1 + b_2\left(\frac{\lambda_2}{\lambda_1}\right)\mathbf{V}_2 + ... + b_n\left(\frac{\lambda_n}{\lambda_1}\right)\mathbf{V}_n\right)
\end{aligned} \tag{A.7}$$

e

$$\mathbf{X}_1 = \frac{\lambda_1}{c_1}\left(b_1\mathbf{V}_1 + b_2\left(\frac{\lambda_2}{\lambda_1}\right)\mathbf{V}_2 + ... + b_n\left(\frac{\lambda_n}{\lambda_1}\right)\mathbf{V}_n\right). \tag{A.8}$$

Após k iterações chegamos à

$$\begin{aligned}
\mathbf{X}_k &= \frac{1}{c_1c_2...c_k}\mathbf{A}^k\mathbf{X}_0 \\
&= \frac{\lambda_1^k}{c_1c_2...c_k}\left(b_1\mathbf{V}_1 + b_2\left(\frac{\lambda_2}{\lambda_1}\right)^{k-1}\mathbf{V}_2 + ... + b_n\left(\frac{\lambda_n}{\lambda_1}\right)^{k-1}\mathbf{V}_n\right),
\end{aligned} \tag{A.9}$$

temos que os termos dependentes em $\mathbf{V}_2$ até $\mathbf{V}_n$ são nulos no limite $k \to \infty$, visto ques os autovalores estão ordenados em ordem crescente, equação A.1, ou seja:

$$\lim_{k\to\infty} b_j\left(\frac{\lambda_j}{\lambda_1}\right)^k\mathbf{V}_j = 0 \text{ para cada } j = 2,\, 3,\, ...,\, n. \tag{A.10}$$

Logo $\mathbf{X}_k$ converge para um vetor paralelo à $\mathbf{V}_1$, e como é requerido que ambos sejam normalizados com a maior coordenada igual à 1 para $\mathbf{X}_k$, constante de proporção de $\mathbf{X}_k$ em relação a $\mathbf{V}_1$ será:

$$\lim_{k\to\infty}\frac{b_1\lambda_1^k}{c_1c_2...c_k} = 1. \tag{A.11}$$

Temos que esse limite não muda ao usar a recursividade num passo a menos:

$$\lim_{k\to\infty} \frac{b_1 \lambda_1^{k-1}}{c_1 c_2 \ldots c_{k-1}} = 1. \tag{A.12}$$

Ao dividirmos as equações A.11e A.12 temos:

$$\lim_{k\to\infty} \frac{b_1 \lambda_1^{k}/(c_1 c_2 \ldots c_{k})}{b_1 \lambda_1^{k-1}/(c_1 c_2 \ldots c_{k-1})} = \lim_{k\to\infty} \frac{\lambda_1}{c_k} = 1 \Rightarrow \lim_{k\to\infty} c_k = 1 \tag{A.13}$$

Sendo assim podemos a partir de um vetor trivial $\mathbf{X}_0 = [1\,1\,..\,1]$ que não seja autovetor da matriz em questão, obter o autovalor de maior amplitude e o autovetor correspondente:

$$\mathbf{AV} = \lambda \mathbf{V}$$

Referências Bibliográficas

[1] Reka Albert and Albert L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1), 2002.

[2] Neil W. Ashcroft and David N. Mermin. *Solid State Physics*. Brooks Cole, January 1976.

[3] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, October 1999.

[4] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[5] Damien Challet and Andrea Lombardoni. Bug propagation and debugging in asymmetric software structures. *Phys. Rev. E*, 70(4):046109, Oct 2004.

[6] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of Networks: From Biological Nets to the Internet and Www (Physics)*. Oxford University Press, March 2003.

[7] Tânia Tomé e Mário José de Oliveira. *Dinâmica Estocástica e Irreversibilidade*, pages 124–125. Edusp, 2001.

[8] P. Erdös and A. Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[9] Giorgio Fagiolo. Clustering in complex directed networks. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 76(2), 2007.

[10] Miguel A. Fortuna and Carlos J. Melián. Do scale-free regulatory networks allow more expression than random ones? *Journal of Theoretical Biology*, 247(2):331–336, July 2007.

[11] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[12] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proc Natl Acad Sci U S A*, 99(12):7821–7826, June 2002.

[13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[14] P. L. Leath. Cluster shape and critical exponents near percolation threshold. *Physical Review Letters*, 36(16):921+, April 1976.

[15] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Physical Review Letters*, 100(11), 2008.

[16] Thomas M. Liggett. *Stochastic Interacting Systems: Contact, Voter and Exclusion Processes (Grundlehren der mathematischen Wissenschaften)*. Springer, November 1999.

[17] T. Maillart, D. Sornette, S. Spaeth, and G. Von Krogh. Empirical tests of zipf's law mechanism in open source linux distribution. *Physical Review Letters*, 101, Junho 2008.

[18] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, October 2002.

[19] Andrea Montanari and Federico Ricci-Tersenghi. Cooling-schedule dependence of the dynamics of mean-field glasses. *Phys. Rev. B*, 70(13):134406, Oct 2004.

[20] M. E. J. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132+, Junho 2001.

[21] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[22] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[23] Frederick Reif. *Fundamentals of Statistical and Thermal Physics (McGraw-Hill Series in Fundamentals of Physics)*. McGraw-Hill Science/Engineering/Math, 1 edition, June 1965.

[24] Ray Solomonoff and Anatol Rapoport. Connectivity of random nets. *Bulletin of Mathematical Biology*, 13(2):107–117, June 1951.

[25] Eugene H. Stanley. *Introduction to Phase Transitions and Critical Phenomena (International Series of Monographs on Physics)*. Oxford University Press, USA, julho 1987.

[26] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[27] Gang Yan, Tao Zhou, Jie Wang, Zhong-Qian Fu, and Bing-Hong Wang. Epidemic spread in weighted scale-free networks. *Chinese Physics Letters*, 22:510, 2005.