

Melhorando a Precisão e Repetibilidade de Experimentos no PlanetLab

Lauro Luis Costa, Luis C. E. Bona, Elias P. Duarte Jr.

¹Departamento de Informática – Universidade Federal do Paraná
Caixa Postal 19018 – 81531-980 – Curitiba – PR – Brazil

{llcosta, bona, elias}@inf.ufpr.br

Abstract. *Current Internet users and applications require high levels of performance, security, scalability and mobility. These characteristics were not predicted at the time of its creation. Large-scale testbeds like PlanetLab have been developed to allow the design and realistic evaluation of protocols and architectures to supply the new Internet demands. However, it is particularly hard to guarantee the precision and repeatability of the results obtained. In this work, we present an experimental evaluation of the impact of using subsets of nodes selected with a node connectivity monitoring tool on the precision and repeatability of the results obtained. Applications with different profiles were executed. Results show that the selection strategy based on k-cores reduces the variation on the results obtained by up to 27% and resulted on an average execution time up to 28% faster compared to other alternatives.*

Resumo. *Usuários e aplicações da Internet atualmente demandam características de desempenho, segurança, escalabilidade e mobilidade que não foram previstas no momento de sua criação. Testbeds de larga escala, como o PlanetLab, são peças chave para o desenvolvimento e avaliação de novas aplicações e arquiteturas que atendam a estas demandas. No entanto, um dos principais desafios no PlanetLab é garantir a precisão e repetibilidade dos resultados obtidos. Neste trabalho, apresentamos uma avaliação experimental do impacto da utilização de subconjuntos de nodos selecionados por uma ferramenta de monitoramento de conectividade na precisão e repetibilidade dos resultados obtidos. Foram realizados experimentos utilizando aplicações com diferentes perfis de utilização executadas por subconjuntos de nodos obtidos por estratégias de seleção distintas. A estratégia de seleção de nodos k-core reduziu a variação dos resultados obtidos em até 27% e obteve médias de tempo de execução até 28% mais baixa que uma das estratégias alternativas.*

1. Introdução

Desde seu surgimento a Internet vem crescendo e evoluindo rapidamente, de uma rede acadêmica para uma ampla plataforma de informação, interação social e comércio, afetando organizações e indivíduos nas mais diversas formas. As aplicações desenvolvidas, e seu uso, mudaram drasticamente a natureza do tráfego na rede. As novas aplicações demandam características de desempenho, segurança, escalabilidade e mobilidade que não foram previstas na criação da Internet [Pan et al. 2011]. Os *testbeds* de larga escala são

essenciais para a validação de novos protocolos e serviços, entre os principais exemplos de *testbeds* estão as redes GENI¹, FIRE², FIBRE³ e PlanetLab⁴ [Chun et al. 2003].

A rede do *testbed* PlanetLab é formada por uma coleção de máquinas mantidas por instituições de pesquisa. Estas máquinas estão amplamente distribuídas ao redor do mundo, sendo atualmente cerca de 1333 nodos em 634 locais diferentes. O usuário acessa um *slice*, que é uma abstração para recursos disponíveis nos nodos da rede. Este é um ambiente altamente instável [Duarte Jr. et al. 2010], apresentando grandes variações no tempo de resposta e largura de banda disponível para a comunicação entre os nodos, e também variações na disponibilidade de recursos em cada nodo. Apesar destas variações refletirem o comportamento da própria Internet, muitas vezes pesquisadores precisam de maior controle sobre estas variações.

Não é trivial encontrar um grupo de nodos estáveis entre si no PlanetLab. Neste trabalho, utilizamos a ferramenta de seleção de nodos estáveis baseado no monitoramento da comunicação par-a-par proposto em [Bona et al. 2012]. Diferentemente do CoMon [Park and Pai 2006] e SWORD [Albrecht et al. 2008], que monitoram informações de cada nodo individualmente, esta ferramenta de seleção monitora a estabilidade da conexão par-a-par entre todos os nodos da rede.

Este trabalho apresenta uma avaliação experimental do impacto da seleção de nodos na precisão dos experimentos. Também pretende-se demonstrar que o compartilhamento dos parâmetros de seleção de nodos utilizados na ferramenta de seleção é suficiente para a obtenção de um conjunto de nodos com características de estabilidade similar, contribuindo para a repetibilidade e reprodutibilidade dos experimentos. Nos experimentos realizados neste trabalho, os nodos selecionados pela ferramenta utilizada apresentaram melhor desempenho e maior consistência nos resultados obtidos. A média do tempo de execução das aplicações foi até 26% mais baixa que o resultado utilizado por um conjunto de nodos selecionados com uma estratégia alternativa. A variação entre os resultados obtidos durante um período de 40 dias foi avaliada, e os nodos selecionados pela ferramenta de monitoramento obtiveram um coeficiente de variação mais baixo quando comparados aos resultados utilizando nodos da estratégia alternativa.

Os resultados obtidos nos experimentos realizados também demonstraram que a ferramenta de monitoração da estabilidade das conexões fim-a-fim é adequada tanto para aplicações com intensa troca de mensagens quanto aplicações distribuídas com intenso uso de CPU. Os resultados demonstram ainda que a ferramenta é capaz de selecionar conjuntos de nodos com características de estabilidade equivalente, apenas utilizando os mesmos parâmetros de seleção.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 apresenta trabalhos relacionados, a Seção 3 descreve nossa proposta de utilização da ferramenta de seleção de nodos estáveis para melhorar a precisão e repetibilidade de experimentos no PlanetLab. A Seção 4 apresenta uma avaliação experimental da utilização dos subconjuntos de nodos selecionados, seguida da conclusão na Seção 5.

¹<http://www.geni.net>

²<http://www.ict-fire.eu>

³<http://fibre-ict.eu>

⁴<https://www.planet-lab.org>

2. Trabalhos Relacionados

O planejamento de experimentos é uma etapa fundamental no processo de avaliação de desempenho de um sistema através da experimentação [Barros Neto and Scarminio 2002]. Mesmo quando um experimento é repetido sob condições tão semelhantes quanto possível, é comum a ocorrência de flutuações nos resultados de cada repetição. Essa variação é chamada *erro experimental* ou *ruído* [Box et al. 2005].

Os autores do PlanetLab [Spring et al. 2006] afirmam que os resultados dos experimentos no PlanetLab não são reproduzíveis, e que o *testbed* foi projetado para oferecer condições reais de rede e não um ambiente controlável. Entretanto, no caso de aplicações que executam por longos períodos de tempo, os autores sugerem que pesquisadores devem ser capazes de detectar padrões no comportamento da rede e entender o desempenho e confiabilidade de seus experimentos. No caso de aplicações de curta duração, os autores sugerem a seleção de nodos utilizando a ferramenta CoMon.

O CoMon [Park and Pai 2006] é um serviço de monitoramento desenvolvido para o PlanetLab que oferece aos seus usuários informações referentes aos nodos da rede. Cada nodo da rede monitora diversas informações, incluindo *uptime*, se o nodo está respondendo a tentativas de conexão SSH (*Secure Shell*), uso de processador, uso de memória, uso de disco, entre outros. As informações obtidas pelos nodos são mantidas em um servidor central. O usuário pode fazer consultas neste servidor, utilizando diversos parâmetros, e obter uma lista de nodos que apresentem características desejadas para a realização de um experimento, ou ainda nodos problemáticos a serem evitados. O projeto CoMon esteve operacional e disponível para consultas desde 2004, mas foi encerrado em 2012 devido à dificuldades de compatibilidade e manutenção do sistema.

O SWORD [Albrecht et al. 2008] é uma ferramenta de seleção de nodos no PlanetLab que permite localizar um conjunto de nodos que atendam a uma lista de parâmetros desejados. O SWORD se comunica com outras ferramentas de monitoramento do PlanetLab, incluindo o CoMon, para obter informações sobre os nodos. Além das características de cada nodo, como carga do processador, uso de memória, disco e outros, o usuário pode requisitar nodos que atendam parâmetros de comunicação entre si, como largura de banda e tempo de resposta. A interface da ferramenta oferece ao usuário uma lista com diversos parâmetros relacionados aos recursos disponíveis nos nodos e à conexão entre os nodos. O usuário escolhe quais parâmetros são relevantes em sua consulta e define valores mínimos e máximos para estes parâmetros. Devido à indisponibilidade do CoMon, atualmente não é possível selecionar nodos utilizando a ferramenta SWORD.

MyOps⁵ é uma ferramenta de monitoramento do próprio PlanetLab. Esta ferramenta é responsável por verificar se os nodos da rede estão funcionando corretamente e enviar notificações automáticas aos responsáveis caso algum problema seja detectado. Se um nodo fica *offline* por um determinado período de tempo ou é detectada a presença de um *firewall* bloqueando as principais portas, uma notificação é enviada à instituição responsável pelo nodo para que as devidas medidas sejam aplicadas e o nodo volte a funcionar de maneira adequada. O MyOps disponibiliza um portal com uma lista pública do estado dos nodos monitorados e permite ao usuário executar buscas por nodos e obter

⁵<http://monitor.planet-lab.org/monitor/node> e <http://monitor.planet-lab.eu/monitor/node>

informações relacionadas, como estado (*online* ou *offline*), estado do SSH e *uptime*.

A execução de experimentos repetíveis em condições realísticas para redes sem fio é abordada em [Lenders and Martonosi 2009]. Experimentos com comunicação sem fio são difíceis de repetir com precisão e modelos de simulação abstratos utilizam parâmetros teóricos não realísticos. Os autores apresentam modelos para a camada física e MAC de redes sem fio baseados em medições realizadas em redes reais, para serem utilizados em simulações e emulações.

Um *framework* para a automatização do projeto e execução de experimentos chamado Weevil é apresentado em [Y. et al. 2008]. O trabalho foca-se na automatização da geração de cargas de trabalho e controle de parâmetros para a obtenção de resultados representativos e repetíveis. Os autores ressaltam a necessidade de controle sobre o ambiente de experimentação e as dificuldades de se obter tal controle no PlanetLab.

3. Seleção de Nodos Estáveis para a Precisão e Repetibilidade de Experimentos no PlanetLab

Esta seção descreve de forma sucinta a ferramenta de seleção de nodos utilizada neste trabalho [Bona et al. 2012]. A arquitetura da ferramenta de seleção de nodos baseada na monitoração de estabilidade das interações fim-a-fim é composta por três módulos: módulo de monitoramento, módulo servidor e módulo cliente. Estes módulos são responsáveis, respectivamente, pelo monitoramento das interações entre os nodos, armazenamento centralizado das informações de monitoramento e interface de interação com o usuário.

Cada nodo do PlanetLab executa um *daemon* de monitoramento. Estes *daemons* se comunicam entre si, enviando e respondendo mensagens utilizando um *socket* UDP. Cada *daemon* deve enviar, periodicamente, mensagens para outros nodos. O tempo até a resposta de cada mensagem (RTT) é observado e armazenado. Um valor de *timeout* é utilizado e caso uma resposta não seja obtida antes do *timeout*, o nodo é considerado indisponível. É importante notar que o RTT é medido na camada de aplicação, portanto o tempo de resposta é influenciado não apenas pelas características de rede, mas também pela carga de processamento no nodo. O módulo servidor é responsável por armazenar os dados obtidos pelo módulo de monitoramento e oferecer dados ao módulo cliente. O servidor armazena em seu banco de dados as informações obtidas na última hora de monitoramento e versões sumarizadas destes dados.

O módulo cliente oferece ao usuário uma interface onde ele pode selecionar um intervalo de tempo e um limiar de RTT, para obter como resposta, uma lista de pares de nodos que apresentaram um tempo de resposta entre si abaixo do limiar selecionado durante o período de tempo desejado. A interface exibe a lista de pares de nodos obtidos e também permite ao usuário selecionar um subconjunto destes nodos utilizando diferentes estratégias de seleção.

Uma lista de pares de nodos que apresentam tempo de resposta entre si abaixo de um certo limiar em um período de tempo é na verdade uma lista de arestas que forma um grafo, onde cada nodo é representado por um vértice, e uma aresta ligando dois vértices indica que o RTT entre os nodos representados está abaixo de um certo limiar no intervalo de tempo selecionado. A presença de um vértice representando um nodo neste grafo não significa que o nodo faz parte de um subconjunto de nodos estáveis. É desejável encontrar

um subconjunto de nodos que apresente um alto grau de conectividade entre si, ou seja, um subgrafo onde os vértices possuem um alto grau de conectividade. Para isto, diferentes estratégias de seleção podem ser utilizadas, como a estratégia do Grau Mínimo, Maior Grau Mínimo, *k-Core* e Clique Estável. Os experimentos realizados pelos autores em [Bona et al. 2012] demonstraram que a melhor estratégia a ser utilizada é o *k-Core*.

4. Resultados Experimentais

Esta seção descreve os resultados experimentais obtidos na avaliação da repetibilidade de experimentos no *testbed* PlanetLab. A métrica utilizada para avaliar precisão dos experimentos realizados é o *coeficiente de variação* (desvio padrão dividido pela média). O coeficiente de variação é proporcional à média, portanto é adequado para a comparação da variação entre o resultados de amostras com médias diferentes.

Foram implementadas três aplicações distribuídas para serem executadas no ambiente do PlanetLab. As aplicações escolhidas apresentam diferentes características de uso de recursos, variando de intensa troca de mensagens e reduzido uso de CPU a intenso uso de CPU e reduzida troca de mensagens. Foram obtidos subconjuntos de nodos utilizando a estratégia de seleção descrita na Seção 3 e também através de uma estratégia alternativa. Para cada aplicação, a precisão dos resultados obtidos utilizando cada subconjunto de nodos é comparada utilizando o coeficiente de variação.

Os experimentos foram organizados em *rodadas de experimentos* e *séries*. Uma série é composta por 30 rodadas de experimentos e cada rodada de experimentos consiste na execução das três aplicações utilizando cada um dos subconjuntos de nodos. Uma rodada de experimentos se completa em aproximadamente 150 minutos, a próxima rodada inicia-se automaticamente logo ao fim da anterior, sem nenhuma restrição de horário ou dia da semana. As séries são iniciadas manualmente e têm duração de aproximadamente 3 dias. No total, foram executadas 300 rodadas de experimentos organizadas em 10 séries, sendo a primeira série iniciada em 20 de abril de 2014 e a última em 25 de maio de 2014.

Esta seção está organizada da seguinte maneira. A Subseção 4.1 descreve as estratégias e os parâmetros utilizados na seleção dos subconjuntos de nodos utilizados para a realização dos experimentos. A Subseção 4.2 apresenta detalhes das aplicações utilizadas nos experimentos e os resultados obtidos. A Subseção 4.3 apresenta uma análise geral dos resultados dos experimentos realizados.

4.1. Seleção dos Subconjuntos de Nodos

Apesar de o PlanetLab ser formado por mais de mil nodos, o número de nodos acessíveis via SSH durante a execução destes experimentos foi muito mais baixo, usualmente em torno de 400 nodos. Para a execução dos experimentos, foram utilizados quatro subconjuntos de 150 nodos. Dois subconjuntos foram selecionados utilizando a estratégia de seleção de nodos apresentada na Seção 3, chamados *core-150ms* e *core-150ms-everyround*. Os outros dois subconjuntos, obtidos utilizando uma estratégia alternativa, são chamados *ping.c3sl* e *ping.uk*.

Os subconjuntos *k-core* (*core-150ms* e *core-150ms-everyround*) foram selecionados utilizando os seguintes parâmetros: 150 nodos, dados de monitoramento obtidos na última hora, limiar do tempo de resposta de 150ms e estratégia de seleção *k-core*. Uma seleção de nodos utilizando estes parâmetros usualmente resulta em um

subconjunto de nodos com grau mínimo próximo a 120, significando que todos os nodos do subconjunto têm conexão com RTT menor que 150ms para pelo menos outros 120 nodos dentro do subconjunto selecionado.

O subconjunto `core-150ms` foi obtido antes do início dos experimentos, no dia 20 de abril de 2014. Este subconjunto permanece inalterado durante a execução de todos os experimentos. Já o subconjunto `core-150ms-everyround` é obtido novamente no início de cada *rodada de experimentos*. Os nodos são obtidos sempre utilizando os mesmos parâmetros de seleção.

Durante uma das séries de experimentos realizados, a composição do subconjunto `core-150ms-everyround` foi observada. Em um período de aproximadamente três dias, tempo necessário para completar a série, um total de 262 nodos distintos fizeram parte do subconjunto `core-150ms-everyround` em ao menos uma das 30 rodadas de experimento. Destes 262 nodos, 59 foram constantemente selecionados em todas as rodadas e 123 foram selecionados mais de 20 vezes. Por outro lado, 42 nodos foram selecionados 5 vezes ou menos e apenas 11 foram selecionados uma única vez.

Para comparação, selecionamos outros dois subconjuntos de nodos com uma estratégia alternativa, utilizando os valores de tempos de resposta obtidos pela troca de mensagens *Echo Request* e *Echo Reply* do protocolo ICMP, implementada pelo comando *ping*. Nesta estratégia, um nodo central é escolhido e é construída uma lista com os tempos de resposta obtido pela média de cinco execuções do comando *ping* de cada nodo para o nodo central escolhido. Os nodos que obtiverem os valores de tempo de resposta mais baixos são selecionados. Ambos os subconjuntos foram selecionados antes do início dos experimentos, em 20 de abril de 2014, e permanecem inalterados durante as execuções de todos os experimentos. A ferramenta CoMon monitorava o tempo de resposta entre os nodos e seu servidor, esta informação permitia ao usuário selecionar um subconjunto de nodos com os menores tempos de resposta para um servidor central, de forma muito similar a estratégia alternativa descrita.

O primeiro subconjunto de nodos obtidos através desta estratégia, chamado `ping.c3sl`, utilizou como nodo central o nodo `planetlab2.c3sl.ufpr.br`, um dos dois nodos mantidos no PlanetLab pelo Centro de Computação Científica e Software Livre na Universidade Federal do Paraná. O segundo subconjunto, chamado `ping.uk`, utilizou como nodo central o `planetlab2.xeno.cl.cam.ac.uk`. Este nodo foi escolhido por estar constantemente entre os nodos estáveis selecionados utilizando a estratégia de seleção *k-core*, portanto este subconjunto deve conter muitos dos nodos também presentes no subconjunto `core-150ms`.

O subconjunto `ping.uk` é bastante semelhante ao subconjunto `core-150ms`, dos 150 nodos selecionados, 100 estão presentes nos dois subconjuntos. Já o subconjunto `ping.c3sl` é bastante diferente dos outros subconjuntos, apenas 11 de seus nodos também estão presentes no subconjunto `ping.uk` e 27 estão presentes no subconjunto `core-150ms`. Portanto, o subconjunto `ping.uk` é bastante similar aos subconjuntos obtidos a partir da estratégia *k-core*, enquanto o subconjunto `ping.c3sl` é bastante diferentes dos outros três subconjuntos.

4.2. Aplicações Executadas e Resultados

Os experimentos foram realizados no período entre 20 de abril e 28 de maio de 2014. Os experimentos foram executados de forma serializada, alternando os subconjuntos de nodos e as aplicações utilizadas. É importante destacar que, apesar de nossos experimentos serem realizados individualmente, a todo momento existem *slices* pertencentes a outros usuários concorrendo pelos recursos dos nodos.

Para a realização dos experimentos foram utilizadas três aplicações. Duas destas aplicações foram implementadas utilizando o *framework* Map-Reduce [Dean and Ghemawat 2004] e a terceira aplicação utiliza o protocolo *BitTorrent*⁶. As duas aplicações Map-Reduce foram desenvolvidas utilizando o Mincemeatpy⁷, uma implementação *leve* do Map-Reduce em Python. O Mincemeatpy permite o desenvolvimento de aplicações simples e não requer a instalação e configuração complexa antes da execução de um experimento.

Uma aplicação Mincemeatpy possui um único nodo *master* e vários *workers*. O nodo *master* organiza as funções *Map* e *Reduce* como tarefas e os *workers* se conectam ao *master* e executam tarefas assinaladas. Nesta implementação, os *workers* não se comunicam entre si, toda comunicação é gerenciada pelo nodo *master*.

As subseções seguintes apresentam mais detalhes sobre as características de cada aplicação e os resultados obtidos nos experimentos realizados com cada uma delas. A primeira subseção apresenta a aplicação *Triangular.py*, a subseção seguinte apresenta a aplicação *Bit.py* e a terceira subseção apresenta a aplicação *Torrent*.

4.2.1. Aplicação Map-Reduce 1: *Triangular.py*

A primeira aplicação desenvolvida utilizando o Mincemeatpy, *Triangular.py*, retorna uma lista com os N primeiros números triangulares. O n -ésimo número triangular T_n é dado por $(n^2 + n)/2$, para n pertencente ao conjunto dos números naturais. Com os parâmetros selecionados para esta sequência de experimentos, a aplicação calcula os primeiros 8192 números triangulares. A implementação desta aplicação não é a forma mais eficiente de se encontrar números triangulares, mas tem como característica o baixo uso de CPU e intensa troca de mensagens. Para calcular os primeiros 8192 números triangulares, a função *map* emite mais de 33 milhões de pares (chave,valor)

Os gráficos na Figura 1 apresentam os resultados obtidos com a execução da aplicação *Triangular.py*. Foram realizadas 300 rodadas de experimentos, organizadas em 10 séries de 30 repetições da aplicação para cada subconjunto. O eixo y indica o tempo de execução em segundos e o eixo x indica a data de execução de cada série de experimentos. A linha azul cruzando o gráfico representa a média do tempo de execução destas 300 repetições, e a sombra cinza o intervalo de confiança no nível de 95%. A linha verde apresenta a média e intervalo de confiança no nível de 95% para cada série de 30 repetições. Cada x vermelho representa o tempo de execução de um experimento.

Observando a distribuição dos x vermelhos, nota-se a presença de diversos valores muito distantes da média nos quatro gráficos, como por exemplo os dois resultados acima

⁶http://www.bittorrent.org/beps/bep_0003.html

⁷<https://github.com/michaelfairley/mincemeatpy>

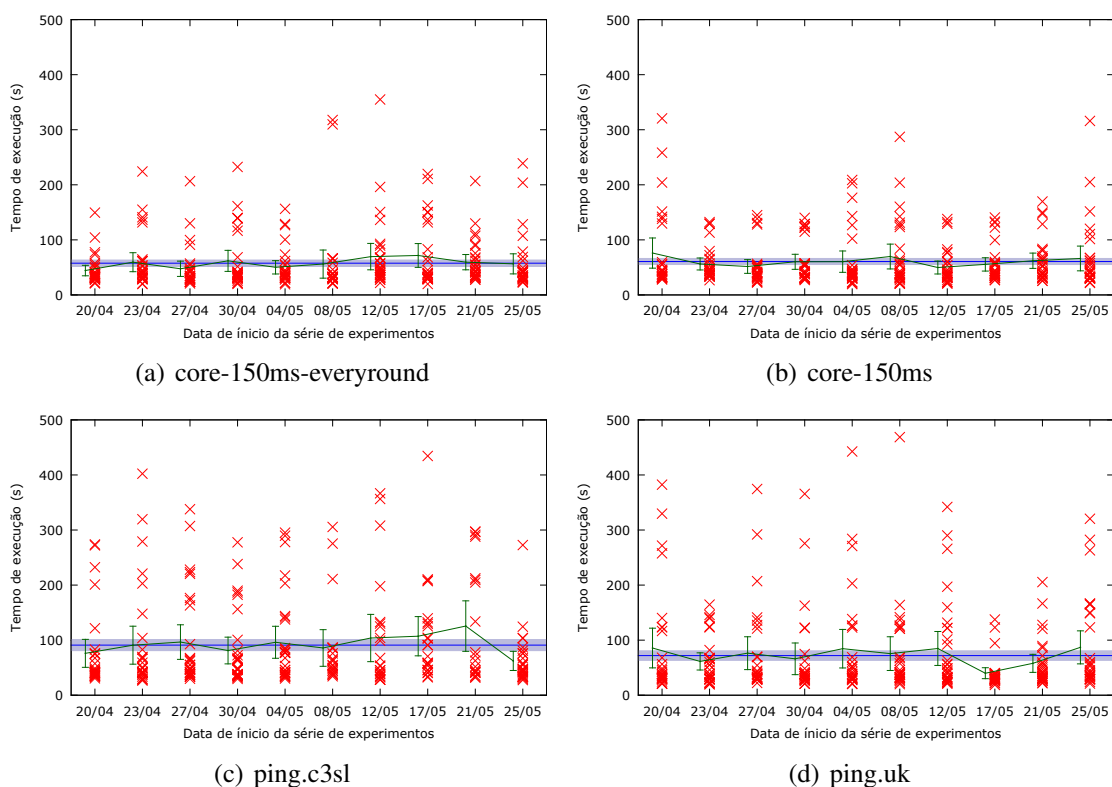


Figura 1. Tempos de execução da aplicação *Triangular.py*.

de 300 segundos obtidos pelo subconjunto *core-150ms-everyround* na série iniciada no dia 08/05. Porém, os valores distantes da média aparecem com maior frequência para os subconjuntos obtidos com a estratégia *ping*. Observando-se a linha verde, comparando as médias obtidas em cada série de 30 repetições e a média geral obtida por todas as 300 repetições do experimento, a distância entre estas médias é menor nos resultados obtidos com os subconjuntos *k-core* comparados aos resultados obtidos com os subconjuntos *ping*.

Subconjunto	core-150ms everyround	core-150ms	ping.c3sl	ping.uk
Média (s)	52,79	56,36	85,08	65,59
Desvio Padrão (s)	37,26	36,65	71,57	60,09
Mínimo (s)	21,02	21,69	29,13	20,95
Máximo (s)	206,89	202,69	307,26	320,42
Coefficiente de variação	0,705	0,650	0,841	0,916

Tabela 1. Sumarização dos resultados para a aplicação *Triangular.py*.

A Tabela 1 apresenta informações referentes aos tempos de execução de todas as 300 repetições da aplicação *Triangular.py* com cada subconjunto de nodos, excluindo-se 5% dos dados mais discrepantes, os 2,5% resultados mais altos e mais baixos obtidos. Todos os subconjuntos apresentaram um coeficiente de variação bastante alto, mas a estratégia de seleção *k-core* contribuiu para o aumento da precisão, obtendo coeficientes de variação mais baixos, comparados aos subconjuntos *ping*. Observando a última linha da tabela, o coeficiente de variação dos subconjuntos *k-core* é de 0,705 e 0,650, enquanto os

subconjuntos *ping* obtiveram 0,841 e 0,916.

Já observando-se as médias de tempo de execução, é possível perceber que os subconjuntos *k-core* obtiveram um tempo de execução médio mais baixo que os subconjuntos *ping*. Considerando os resultados eliminando 5% dos dados discrepantes, as médias de tempo de execução dos subconjuntos *k-core* foram de 52,79s e 56,36s, contra as médias de 85,08s e 65,59s obtidas pelos subconjuntos *ping*.

Em geral, os subconjuntos obtidos com a estratégia *k-core* resultaram em tempo de execuções mais baixos e resultados mais consistentes do que os subconjuntos obtidos com a estratégia *ping*. O subconjunto `core-150ms-everyround` obteve uma média de tempo de execução mais baixa que o subconjunto `core-150ms`. A seleção de novos nodos no início de cada rodada de experimento força a escolha de novos nodos para o experimento, o que deveria implicar em aumento do coeficiente de variação, no entanto o subconjunto `core-150ms-everyround` obteve um coeficiente de variação próximo ao do subconjunto `core-150ms`, demonstrando a capacidade da estratégia *k-core* de selecionar subconjuntos de nodos com características de conectividade semelhante ao longo do tempo.

4.2.2. Aplicação Map-Reduce 2: *Bit.py*

A segunda aplicação desenvolvida foi inspirada no problema resolvido por *mineradores* da cripto moeda Bitcoin⁸. O Bitcoin, lançado em 2009, é uma moeda com criação e transferência baseada em protocolos de código aberto de criptografia. A segurança e confiabilidade desta moeda vem da dificuldade de se validar uma transação incorreta. As transações de Bitcoins são organizadas em blocos e validadas por uma rede P2P. A validação destas transações requer grande poder de processamento, mas gera recompensa financeira aos usuários envolvidos, chamados *mineradores*.

A validação de um bloco de transações ocorre utilizando a função *hash* SHA-256 [Eastlake 3rd 2001]. De forma simplificada, o objetivo de um minerador de Bitcoin é, dado um valor de entrada que representa o bloco de transações e informações relacionadas ao bloco anterior, encontrar um valor v tal que $\text{SHA-256}(\text{entrada} + v) < h$, sendo h um valor em hexadecimal de 256 bits, usualmente com muitos zeros à esquerda. A aplicação implementada resolve uma versão simplificada deste problema. Com os parâmetros selecionados, durante uma execução da aplicação, a função SHA-256 é calculada para quase 7 bilhões de valores de v . Destes bilhões de valores para v , mais de 6 mil resultaram em 5 zeros à esquerda na *hash* resultante, mais de 300 com 6 zeros, 36 com 7 zeros e apenas 2 com 8 zeros.

Os gráficos na Figura 2 apresentam os resultados obtidos com a execução da aplicação *Bit.py*, seguindo a mesma organização utilizada nos gráficos da aplicação anterior. Observando a distribuição dos x vermelhos, nota-se a presença de diversos valores muito distantes da média nos resultados dos subconjuntos utilizando a estratégia *ping*. Os resultados dos subconjuntos obtidos utilizando a estratégia *k-core* apresentam menor dispersão. Por exemplo o gráfico do subconjunto `core-150ms-everyround` apresenta alguns resultados mais distantes da média nos dias 4 e 17 de maio, já os dois subconjuntos *ping*

⁸<https://bitcoin.org/en/>

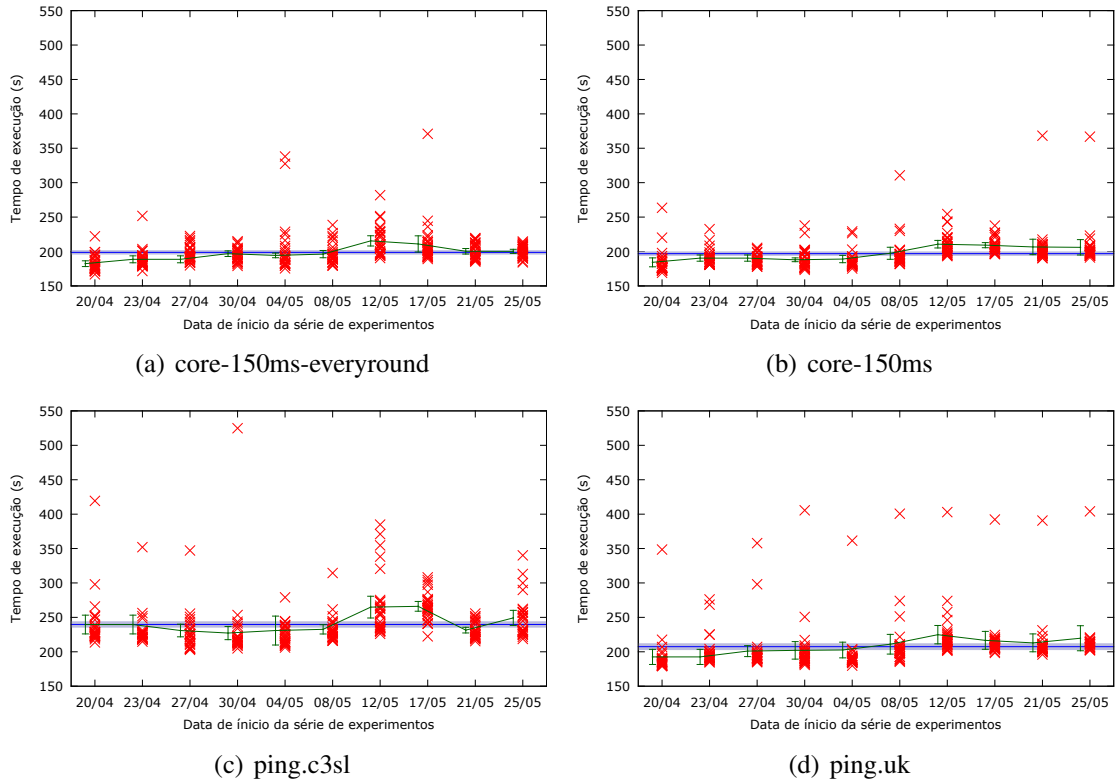


Figura 2. Tempos de execução da aplicação *Bit.py*.

apresentam algum resultados distante da média em todas as séries de experimentos.

Observando-se a linha verde nos quatro gráficos, nota-se um comportamento comum aos quatro subconjuntos, com resultados melhores nas primeiras séries de experimentos, e uma piora nos experimentos realizados após o dia 12/05. Esta alteração indica alguma variação no desempenho da rede do PlanetLab, afetando todos os nodos. No entanto, o impacto deste comportamento foi mais suave nos resultados dos subconjuntos *k-core*. Ainda sobre os subconjuntos *k-core*, destaca-se um intervalo de confiança menor, tanto para cada série de experimentos (em verde), quanto para as 300 repetições combinando os resultados das 10 séries (sombra cinza).

Subconjunto	core-150ms everyround	core-150ms	ping.c3sl	ping.uk
Média (s)	196,86	195,02	236,39	202,82
Desvio Padrão (s)	13,21	13,09	22,14	18,86
Mínimo (s)	174,89	175,31	206,37	180,95
Máximo (s)	238,54	237,70	338,23	348,51
Coefficiente de variação	0,067	0,067	0,093	0,092

Tabela 2. Sumarização dos resultados para a aplicação *Bit.py*.

A Tabela 2 apresenta os resultados para a aplicação *Bit.py*, seguindo a mesma organização utilizada na tabela anterior. Esta aplicação resultou em coeficientes de variação mais baixos nos resultados dos quatro subconjuntos, mas a estratégia de seleção *k-core* obteve maior precisão com coeficientes de variação relativamente mais baixos. Obser-

vando a última linha da tabela, o coeficiente de variação dos subconjuntos *k-core* é de 0,067 para ambos, enquanto os subconjuntos *ping* obtiveram 0,093 e 0,092.

Já observando-se as médias de tempo de execução, é possível perceber que os subconjuntos *k-core* obtiveram um tempo de execução médio mais baixo que os subconjuntos *ping*. Considerando os resultados eliminando 5% dos dados discrepantes, as médias de tempo de execução dos subconjuntos *k-core* foram de 195,86s e 195,02s, contra as médias de 236,39s e 202,82s obtidas pelos subconjuntos *ping*.

Os subconjuntos *core-150ms* e *core-150ms-everyround* exibiram comportamento bastante similar entre si e superiores aos subconjuntos *ping.c3sl* e *ping.uk* na média de tempo de execução. O coeficiente de variação foi baixo em todos os subconjuntos e ligeiramente inferior para os subconjuntos *k-core*.

4.2.3. Aplicação 3: *Torrent*

A terceira aplicação utilizada para os experimentos apresentados neste trabalho utiliza uma implementação do protocolo *BitTorrent*, o *Transmission*⁹, software padrão nas distribuições Ubuntu atuais. O protocolo *BitTorrent* suporta o compartilhamento de arquivos em redes P2P. A primeira versão do protocolo foi apresentada em 2001 e, atualmente, é uma das aplicações mais comuns para transferência de arquivos.

Para a execução dos experimentos com *torrents* apresentados nesta Seção, foi utilizado um arquivo gerado com dados aleatórios de tamanho 27MB. Este arquivo está, inicialmente, disponível em apenas um *host*, o tempo entre o início do experimento e a obtenção de uma cópia completa do arquivo foi observada em cada nodo. A métrica utilizada para avaliar a execução do experimento foi a média do tempo observado nos 80 primeiros nodos a completarem a transferência do arquivo.

Os gráficos na Figura 3 apresentam os resultados obtidos com a execução da aplicação *Torrent*, seguindo a mesma organização utilizada nos gráficos das aplicações anteriores. Observando-se a distribuição dos *x* vermelhos, é possível notar a grande dispersão das amostras para os subconjuntos da estratégia *ping*. Os subconjuntos *k-core* apresentam amostras mais concentradas e próximas da média geral obtida, com exceção da série realizada no dia 20/04 utilizando o subconjunto *core-150ms*, que apresentou uma dispersão nas amostras incomum às outras séries.

Subconjunto	core-150ms everyround	core-150ms	ping.c3sl	ping.uk
Média (s)	98,46	97,09	240,25	126,37
Desvio Padrão (s)	15,84	15,07	47,67	21,63
Mínimo (s)	75,83	75,93	158,77	88,06
Máximo (s)	137,97	142,30	354,43	188,35
Coeficiente de Variação	0,161	0,155	0,198	0,171

Tabela 3. Sumarização dos resultados para a aplicação *Torrent*.

A Tabela 3 apresenta os resultados para a aplicação *Torrent*, seguindo a mesma organização utilizada nas tabelas anteriores. Observando-se o coeficiente de variação, é

⁹<http://www.transmissionbt.com>

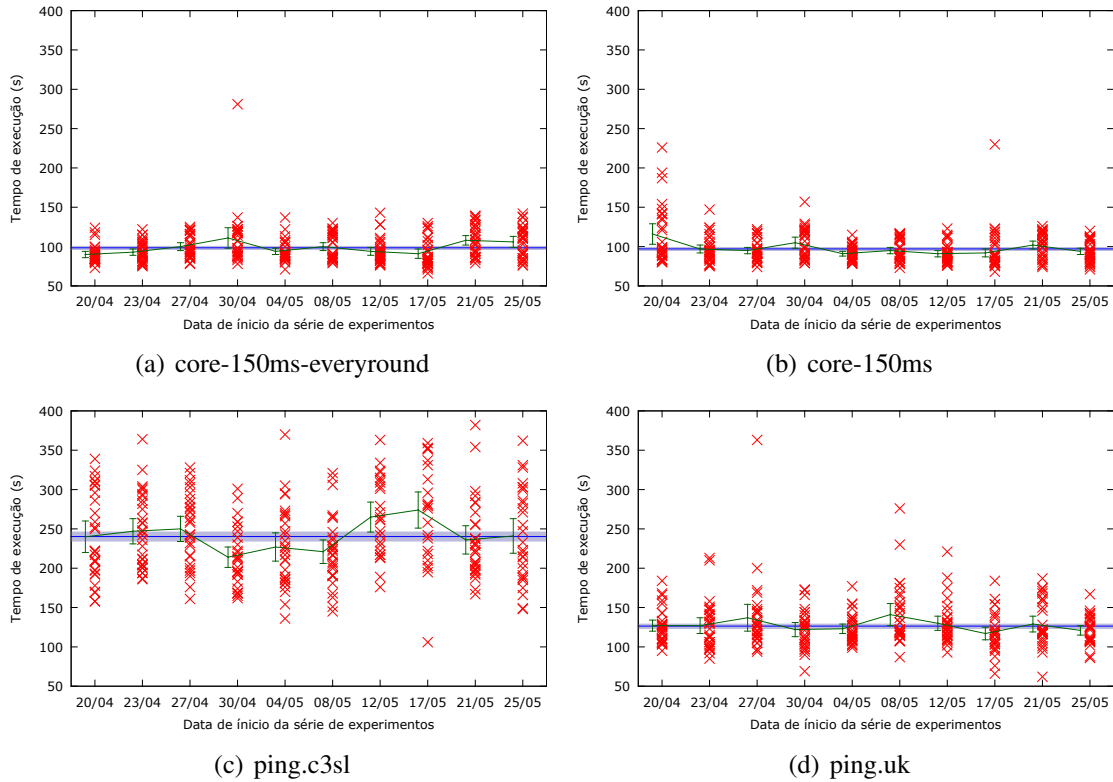


Figura 3. Tempos de execução da aplicação Torrent.

possível notar que os subconjuntos *k-core* obtiveram um coeficiente de variação próximos entre si, 0,161 e 0,155, e superiores aos subconjuntos *ping*, 0,198 e 0,171.

A média de tempo obtida pelos subconjuntos *k-core* também são muito semelhantes entre si, 98,46s e 97,09s, superiores à média de 126,37s obtida pelo subconjunto *ping.uk*. Já o subconjunto *ping.c3sl* apresentou um desempenho muito inferior, com uma média de 240,25s, evidenciando uma baixa conectividade entre os nodos do subconjunto.

4.3. Discussão

Em todos os experimentos apresentados, os subconjuntos selecionados pela estratégia *k-core* obtiveram médias de tempo de execução mais baixas que as obtidas pelos subconjuntos selecionados pela estratégia *ping*. O subconjunto *core-150ms-everyround* foi em média 24% mais rápido que o subconjunto *ping.uk* na aplicação *Triangular.py*, 3% na aplicação *Bit.py* e 28% na aplicação *Torrent*. Comparada ao subconjunto *ping.c3sl*, o subconjunto *core-150ms-everyround* foi em média 61% mais rápido na aplicação *Triangular.py*, 20% mais rápido na aplicação *Bit.py* e quase duas vezes e meia mais rápido na aplicação *Torrent*.

A estratégia *k-core* também foi mais eficiente para minimizar o impacto de fatores externos nos resultados dos experimentos, reduzindo o coeficiente de variação dos resultados obtidos. O subconjunto *core-150ms* obteve um coeficiente de variação de 0,65 na aplicação *Triangular.py*, contra 0,916 do subconjunto *ping.uk* e 0,841 do subconjunto *ping.c3sl*. Os coeficientes obtidos na aplicação *Bit.py* também foram menores,

0,067 do subconjunto `core-150ms` contra 0,092 e 0,093 dos conjuntos `ping.uk` e `ping.c3sl`. O coeficiente de variação obtido pelo subconjunto `core-150ms` é novamente mais baixo na aplicação *Torrent*, 0,155, enquanto `ping.uk` e `ping.c3sl` obtiveram respectivamente 0,191 e 0,171.

Comparando os dois subconjuntos obtidos pela estratégia *k-core*, os resultados obtidos são bastante similares, com médias de tempo de execução equivalentes em todas as aplicações, coeficientes de variação idêntico na aplicação *Bit.py* e bastante próximos nas aplicações *Triangular.py* e *Torrent*. Os resultados obtidos pelo subconjunto `core-150ms` demonstram a estabilidade dos nodos selecionados pela estratégia *k-core*. Durante um período de 40 dias, o subconjunto permaneceu fixo e manteve as propriedades de conectividade entre os nodos, obtendo resultados com baixa variação durante o período de experimentação.

A estratégia de seleção de nodos `core-150ms-everyround` inclui um número maior de nodos no experimento. Como foi observado em um período de três dias, a seleção de 150 nodos executada antes do início de cada uma das 30 rodadas de experimento de uma série incluiu um total de 262 nodos diferentes. É esperado que a utilização de diferentes nodos resulte em uma maior variação dos resultados, no entanto, os resultados obtidos pelo subconjunto `core-150ms-everyround` foram muito similares ao subconjunto fixo `core-150ms`. Portanto, a estratégia de seleção de nodos estáveis é capaz de selecionar subconjuntos de nodos com características de estabilidade semelhantes apenas utilizando os mesmos parâmetros de seleção, contribuindo para a reprodutibilidade dos experimentos.

5. Conclusão

Novas soluções, arquiteturas e serviços para a Internet precisam ser testados em um ambiente realista e de larga escala. O PlanetLab é um *testbed* de escala planetária, com mais de mil nodos espalhados ao redor do mundo, o qual oferece aos pesquisadores um ambiente real para a execução de experimentos. No entanto, a execução de experimentos no PlanetLab pode atingir um alto grau de complexidade, especialmente pela existência de instabilidades e variações de desempenho nos nodos e na própria rede. Este trabalho apresentou uma avaliação experimental do impacto do uso de uma ferramenta de seleção na precisão dos resultados obtidos em experimentos realizados no PlanetLab.

Os resultados obtidos confirmaram que o PlanetLab é um ambiente instável e é muito difícil obter repetibilidade nos experimentos realizados. No entanto, estas variações não atingem todos os nodos da mesma maneira e com a mesma intensidade. A utilização de subconjuntos de nodos obtidos pela estratégia de seleção *k-core* reduziu as variações observadas nos resultados obtidos, aumentando a precisão e contribuindo para a repetibilidade dos experimentos. Além disso, os subconjuntos de nodos selecionados pela ferramenta de monitoração obtiveram melhores médias de tempo de execução para as aplicações testadas, demonstrando a capacidade da estratégia de seleção de encontrar um subconjunto de nodos com melhor conectividade entre todos os pares, confirmando os resultados originais obtidos pelo autor da ferramenta.

Os experimentos realizados neste trabalho demonstraram que a ferramenta de seleção de nodos baseada no monitoramento das interações fim-a-fim é eficiente em obter subconjuntos de nodos estáveis para a realização de experimentos. Porém, existem outras

informações sobre os nodos que a ferramenta de seleção não foi projetada para monitorar. Durante a execução dos experimentos, foi detectada especialmente a falta de informações relacionadas à disponibilidade de memória e espaço em disco dos nodos. Um trabalho futuro importante seria a realização de novos experimentos para avaliar a incorporação da monitoração de outros parâmetros à estratégia de seleção, permitindo uma filtragem de nodos selecionados pela estratégia *k-core*.

Referências

- Albrecht, J., Oppenheimer, D., Vahdat, A., and Patterson, D. (2008). Design and implementation trade-offs for wide-area resource discovery. *ACM Transactions on Internet Technology (TOIT)*, 8(4):18.
- Barros Neto, B. and Scarminio, I. (2002). *Como Fazer Experimentos: pesquisa e desenvolvimento na ciência e indústria*. Editora da Unicamp.
- Bona, L., Duarte Jr., E., and Garrett, T. (2012). Monitoring pairwise interactions to discover stable wormholes in highly unstable networks. In Korakis, T., Zink, M., and Ott, M., editors, *Testbeds and Research Infrastructure. Development of Networks and Communities*, volume 44 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 146–161. Springer Berlin Heidelberg.
- Box, G., Hunter, W., and Hunter, J. (2005). *Statistics for experimenters: an introduction to design, data analysis, and model building*. Wiley series in probability and mathematical statistics. Wiley.
- Chun, B., Culler, D., Roscoe, T., Bavier, A., Peterson, L., Wawrzoniak, M., and Bowman, M. (2003). Planetlab: an overlay testbed for broad-coverage services. *ACM SIGCOMM Computer Communication Review*, 33(3):3–12.
- Dean, J. and Ghemawat, S. (2004). Mapreduce: Simplified data processing on large clusters. In *OSDI 2004*, pages 137–150.
- Duarte Jr., E., Garrett, T., Bona, L., Carmo, R., and Zuge, A. (2010). Finding stable cliques of planetlab nodes. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 317–322. IEEE.
- Eastlake 3rd, D. (2001). RFC 6234: Secure hash algorithm.
- Lenders, V. and Martonosi, M. (2009). Repeatable and realistic experimentation in mobile wireless networks. *Mobile Computing, IEEE Transactions on*, 8(12):1718–1728.
- Pan, J., Paul, S., and Jain, R. (2011). A survey of the research on future internet architectures. *Communications Magazine, IEEE*, 49(7):26–36.
- Park, K. and Pai, V. S. (2006). Comon: A mostly-scalable monitoring system for planetlab. *SIGOPS Oper. Syst. Rev.*, 40(1):65–74.
- Spring, N., Peterson, L., Bavier, A., and Pai, V. (2006). Using planetlab for network research: Myths, realities, and best practices. *SIGOPS Oper. Syst. Rev.*, 40(1):17–24.
- Y., W., Carzaniga, A., and Wolf, A. (2008). Four enhancements to automated distributed system experimentation methods. In *Software Engineering, 2008. ICSE '08. ACM/IEEE 30th International Conference on*, pages 491–500.