COPPE/UFRJ

SOBRE MEDIDAS DE DESEMPENHO DA INTERNET PARA O USO EM APLICAÇÕES DE REDES

Antonio Augusto de Aragão Rocha

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Rosa Maria Meri Leão
Edmundo Albuquerque de Souza e Silva

Rio de Janeiro
Abril de 2010

SOBRE MEDIDAS DE DESEMPENHO DA INTERNET PARA O USO EM APLICAÇÕES DE REDES

Antonio Augusto de Aragão Rocha

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Rosa Maria Meri Leão, Dr.

Prof. Edmundo Albuquerque de Souza e Silva, Ph.D.

Prof. José Ferreira de Rezende, Dr.

Prof. Daniel Ratton Figueiredo, Ph.D.

Prof. Célio Vinicius Neves de Albuquerque, Ph.D.

Prof. Artur Ziviani, Dr.

RIO DE JANEIRO, RJ – BRASIL

ABRIL DE 2010

Rocha, Antonio Augusto de Aragão

Sobre medidas de desempenho da Internet para o uso em aplicações de redes/Antonio Augusto de Aragão Rocha. – Rio de Janeiro: UFRJ/COPPE, 2010.

XIX, 173 p.: il.; 29, 7cm.

Orientadores: Rosa Maria Meri Leão

Edmundo Albuquerque de Souza e Silva

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2010.

Referências Bibliográficas: p. 158 – 173.

1. Avaliação de desempenho. 2. Medições em redes. 3. Aplicações peer-to-peer. 4. Atraso em um sentido. 5. Capacidade de transmissão. 6. Disponibilidade. 7. Tempo de download. I. Leão, Rosa Maria Meri *et al.*. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

À toda minha família,
em especial a Fabianne e meu
filho Matheus.

Agradecimentos

O término dese trabalho só foi possível devido ao apoio de uma série de pessoas que me acompanharam ao longo dos últimos anos. Assim, se faz necessáio agradecer a todos que direta ou indiretamente me auxiliaram na conclusão do trabalho.

Obrigado a toda minha família, pelo amor, carinho e compreensão que sempre tiverem comigo. Um agradecimento especial aos meus avós Raimundo e Marysses. Essa conquista jamais seria possível sem o apoio deles, pois nunca mediram esforços para possibilitar que eu lutasse por meus objetivos. Muito obrigado por tudo, serei eternamente grato a vocês!

Agradeço a minha esposa também pelo amor, apoio e paciência que me deu ao longo desses anos. Como não poderia deixar de ser, agradeço também ao meu filho Matheus, por me servir de inspiração nos momentos finais da tese.

Obrigado aos meus orientadores, Edmundo e Rosa, pela oportunidade de trabalho e por toda sabedoria que me passaram. Thanks also to professors Don F. Towsley and Arun Venkataramani for the support during my internship at UMass-Amherst. Obrigado também aos membros da banca Arthur, Célio, Rezende e Daniel pelas revisões e comentários sobre o trabalho.

Não posso deixar de agradecer, ainda, a todos os amigos da família LAND/UFRJ e da UMass. Obrigado Bernardo, Beto, Ana, GD, Luiz, Fabrício, Allyson, Flavio, Hugo, Ed, Fernando, Watanabe, e outros. Um agradecimento especial Carol! Thanks Bruno, Sadoc, Antonio, Bruno Gaúcho, André, Marcelo, Yu Gu, Michael Zink, Bin Li, Ramin, Pablo, Boulat, Vicky, Sookhyun.

Por fim, agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo suporte financeiro.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

SOBRE MEDIDAS DE DESEMPENHO DA INTERNET PARA O USO EM APLICAÇÕES DE REDES

Antonio Augusto de Aragão Rocha

Abril/2010

Orientadores: Rosa Maria Meri Leão
Edmundo Albuquerque de Souza e Silva

Programa: Engenharia de Sistemas e Computação

Os serviços mais populares da Internet deixaram de ser exclusivamente aqueles tradicionais. Usuários estão cada vez mais interessados em serviços como multimídia e aplicações P2P. No entanto, serviços como multimídia possuem estreitos requisitos quanto ao desempenho da rede. A crescente demanda por essas aplicações tem motivado o desenvolvimento de novas técnicas de medição para coleta de estatísticas na Internet. Já as aplicações P2P são, sem dúvida, as mais populares dentre todas aquelas da "nova geração". Compreender as características desse modelo de aplicação, com objetivo de melhorar o desempenho de sistemas (por exemplo, tempo de download e disponibilidade) e/ou reduzir o custo (como economia no consumo de banda), é um importante tópico de pesquisa na área de redes.

Esta tese versa sobre a avaliação de medidas de desempenho da Internet para o uso de aplicações na rede. O texto discorre sobre as principais contribuições alcançadas por este trabalho, que são: (i) uma nova técnica de medição ativa não cooperativa para estimar a média e a variância da distribuição do atraso unidirecional; (ii) uma técnica de medição fim-a-fim para inferir a taxa de transmissão de uma máquina conectada através de uma rede sem fio; e, (iii) soluções para aumentar a disponibilidade e reduzir o custo da disseminação de conteúdos em aplicações P2P.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

ON INTERNET MEASUREMENT PERFORMANCE FOR USING ON NETWORK APPLICATIONS

Antonio Augusto de Aragão Rocha

April/2010

Advisors: Rosa Maria Meri Leão
Edmundo Albuquerque de Souza e Silva

Department: Systems Engineering and Computer Science

The most popular Internet services are no more longer the traditional ones. Users are now more interested in services such as multimedia and P2P applications. However, services such as multimedia have narrow network performance requirements. The growing demand for these applications motivates the development of new network measurement techniques for estimating statistics on the Internet. P2P applications are the most popular among all those from "new generation". Thus, to understand the characteristics of this type of application, aiming at improving the system's performance (for instance, download time and availability) and/or reducing the costs (such as savings in bandwidth), is an important topic in network research.

This thesis focuses on the evaluation of Internet measurement performances for using applications. The text describes the main contributions achieved by this work, which are: (i) a new non-cooperative technique for measuring the mean and variance of one-way delay; (ii) an end-to-end technique to infer the transmission rate of a machine connected via a IEEE 802.11 link; and, (iii) solutions to increase availability and reduce the cost for content dissemination using P2P applications.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Glossário

RTT - *Rount Trip Time - Atraso de ida-e-volta.*

OWD - *One-way Delay - Atrado unidirecional.*

Jitter - *Variação do atraso.*

Sondas - *Pacotes usados em medições ativas (Probes).*

Skew - *Diferença na taxa de crescimento dos relógios.*

Offset - *Diferença entre os instantes de tempo de dois relógios.*

P2P - *Peer-to-peer.*

IP - *Internet Protocol.*

TCP - *Transmission Control Protocol.*

UDP - *User Datagram Protocol.*

ICMP - *Internet Control Message Protocol.*

MTU - *Maximum Transmission Unit.*

TTL - *Time to live.*

HTTP - *Hypertext Transfer Protocol.*

FTP - *File Transfer Protocol.*

SNMP - *Simple Network Management Protocol.*

CBR - *Constant Bit Rate - Taxa Constante de Bits.*

Gbps - *Giga bits por segundo.*

Kbps - *Kilo bits por segundo.*

SA - *Sistemas Autonomos.*

ISP - *Internet Service Provider - Provedores de Serviços de Internet.*

NAT - *Network Address Translator.*

QoS - *Quality of service - Qualidade de Serviço.*

PMF - *Probability Mass Function.*

PDF - *Probability Density Function.*

CDF - *Cumulative Distribution Function.*

MSE - *Mean Square Error.*

HMM - *Hidden Markov Model.*

LAN - *Local Area Network - Rede Local.*

WLAN - *Wireless Local Area Network - Rede Local Sem-fio.*

WiMax - *Worldwide Interoperability for Microwave Access.*

Wifi - *Wireless Fidelity.*

DCF - *Distributed Coordination Function.*

DIFS - *DCF Interframe Space.*

SIFS - *Short Interframe Space.*

ACK - *Acknowledgment.*

ISDN - *Integrated Services Digital Network.*

CDMA - *Code Division Multiple Access.*

EVDO - *Evolution Data Optimized (Only).*

UMTS - *Universal Mobile Telecommunications System.*

HSDPA - *High-Speed Downlink Packet Access.*

ITU - *International Telecommunication Union.*

IEEE - *Institute of Electrical and Electronics Engineers.*

IETF - *Internet Engineering Task Force.*

RFC - *Request for comments.*

IPPM WG - *IP Performance Metrics Working Group.*

CAIDA - *Cooperative Association for Internet Data Analysis.*

UFRJ - *Universidade Federal do Rio de Janeiro.*

UNIFACS - *Universidade Salvador.*

UMASS - *University of Massachusetts.*

Palavras Iniciais

ESTA tese versa sobre a avaliação de medidas de desempenho da Internet para o uso de aplicações. O texto descreve as contribuições alcançadas por este trabalho, que estão relacionadas a: (i) técnicas de medição fim-a-fim para a obtenção de métricas de interesse em redes de computadores; e, (ii) análise de disponibilidade e custo para a disseminação de conteúdos em aplicações peer-to-peer na Internet.

Os trabalhos desenvolvidos nesta tese foram realizados em duas etapas distintas. A primeira etapa, que contempla as contribuições relacionadas ao item (i) citado acima, ocorreu exclusivamente na Universidade Federal do Rio de Janeiro, sob a orientação dos professores Rosa Maria Meri Leão e Edmundo A. de Souza e Silva. A segunda etapa, que contempla as contribuições do item (ii), teve início durante o período de estágio de doutoramento (doutorado sanduíche) do aluno, realizado na Universidade de Massachusetts-Amherst, sob a co-orientação do professor Donald F. Towsley, e se estendeu após o seu retorno ao Brasil. Durante a segunda etapa, o aluno integrou um grupo de pesquisa em aplicações P2P daquela universidade e alguns dos trabalhos realizados pelo grupo teve participação ativa dos professores orientadores brasileiros.

Se faz necessário destacar que, embora algumas das publicações obtidas pelo autor desta tese sejam em co-autoria com outros alunos de doutorado da instituição estrangeira, os trabalhos desenvolvidos por cada um no grupo de pesquisa foi bem delimitado e sem sobreposições. Os resultados obtidos por cada um deles são parte das contribuições de sua respectiva tese de doutorado. Portanto, as contribuições relacionadas ao item (ii), relatadas em uma das seções a seguir, fazem parte do trabalho desenvolvido exclusivamente pelo aluno autor desta tese.

Capítulo 1

Introdução

ESTE capítulo discorre a respeito de conceitos fundamentais relacionados ao tema de trabalho desta tese. Na primeira seção é apresentada uma breve descrição sobre a arquitetura, as aplicações e as limitações da Internet (1.1). Serão definidas algumas das principais medidas de desempenho de rede e a importância de estimar essas medidas para as aplicações (1.2). Em seguida, serão descritas as motivações, os objetivos e o resumo das contribuições desta tese (1.3). Por fim, é apresentada a estrutura definida para os demais capítulos deste trabalho (1.4).

1.1 Redes de computadores e Internet: arquitetura, aplicações e limitações

A popularidade das redes de computadores, especialmente das redes baseadas na arquitetura TCP/IP [1], cresceu significativamente nas últimas décadas. Conhecida como a "rede das redes", a Internet hoje é uma imensa rede, organizada em milhares de sistemas autônomos sob diferentes controles administrativos, conectando milhões de diferentes dispositivos eletrônicos, e utilizada por mais de um bilhão e meio de usuários. Segundo dados publicados pela "Internet System Consortium", em julho de 2008, já passavam de 600 milhões o número de terminais conectados à grande rede [2]. A "Internet World Stats" [3] estima que, só na última década, o número de usuários em todo o mundo subiu de 248 milhões para 1.5 bilhões, sendo que no Brasil esse número passou de 5 para 68 milhões de usuários. Mas qual o motivo para esse crescimento da Internet? Obviamente, não existe uma única razão,

mas, certamente, um dos principais fatores, definido ainda no desenvolvimento da Internet, contribuiu significativamente para esse rápido crescimento: a arquitetura simples e descentralizada.

O princípio adotado no desenvolvimento da arquitetura da Internet foi de um modelo simples e descentralizado de conectividade "fim-a-fim". Esse modelo, analisado por Saltzer, Reed e Clark em [4], prevê que a complexidade do sistema de comunicação fique a cargo das estações finais da rede, ou o mais próximo possível delas, sem a existência de entidades centrais de controle. Ortogonalmente diferente do paradigma seguido pelas redes de comutação por circuito, no modelo de conectividade "fim-a-fim", tradicionalmente adotados em redes de comutação por pacotes, o núcleo da rede não faz distinção do tráfego gerado por diferentes aplicações e opera simplesmente como um meio de transporte neutro no encaminhamento dos pacotes. Apenas tarefas simples como endereçamento e encaminhamento dos pacotes são feitas pelos equipamentos no núcleo da rede (roteadores), enquanto que serviços como controles de fluxo e congestionamento, estabelecimento de conexão, resolução de nomes, dentre outros, ficam a critério das aplicações executadas nas estações localizadas nas bordas da rede. Dessa forma, os requisitos necessários para um terminal conectar-se à Internet são mínimos, permitindo que dispositivos de recursos limitados (como PDA's, celulares, sensores, dentre outros) se comuniquem com equipamentos bem mais sofisticados (tais como, grandes servidores e supercomputadores).

O crescimento da popularidade da Internet, na última década, foi acompanhado por um aumento significativo no número de aplicações disponíveis na grande rede. Já faz algum tempo que os serviços mais populares deixaram de ser exclusivamente aqueles tradicionais, como correio eletrônico, Web, acesso remoto e transferência de arquivo. Os usuários, acessando à Internet com taxas de transmissão cada vez mais altas, estão agora interessados também em serviços como voz sobre IP (VoIP), vídeo sob demanda ou em tempo real, aplicações P2P(*peer-to-peer*), jogos "on-line", dentre outros. Ao contrário das aplicações tradicionais que são elásticas [1], alguns desses novos serviços possuem estreitos requisitos quanto ao desempenho da rede.

[1] são chamadas de *elásticas* as aplicações menos sensíveis ao atraso e mais intolerantes à perda de pacotes na rede.

Por exemplo, para que usuários do Skype[5] ou FreeMeeting[6, 7] possam utilizar o serviço de VoIP oferecidos por estas aplicações de forma satisfatória, a taxa de perda e o atraso dos pacotes dessas aplicações não podem ser muito altos. Do contrário, a qualidade do som e a interatividade da conversa serão insatisfatórias.

As aplicações P2P são, sem dúvida, as mais populares dentre todas da "nova geração". Recentes estudos apresentados em [8] indicam que as aplicações P2P (como BitTorrent[9], Emule[10], PPLive[11] e Sopcast[12]) são responsáveis por mais da metade do tráfego gerado atualmente na Internet, em todas as regiões monitoradas no mundo. A fração do tráfego originado de aplicações P2P, em relação ao tráfego total medido em diferentes pontos na Internet, foi de 65% na América do Sul, 70% no Leste Europeu e aproximadamente 55% nas demais regiões da Europa.

As aplicações P2P revolucionaram o modelo de disseminação de conteúdo na Internet. Os sistemas P2P possuem diversas vantagens em relação ao modelo cliente/servidor e aparecem como principal opção para a distribuição de conteúdo digital que visam as melhorias de desempenho (por exemplo, menor tempo de download e maior disponibilidade), redução de custos para grandes servidores (como economia no consumo de banda) e aumento da escalabilidade. A tendência é que cada vez mais empresas de entretenimento como a CNN, Netflix, Rhapsody e Globo utilizem soluções P2P que explorem a capacidade ociosa de seus clientes para auxiliar na disseminação do conteúdo pela Internet. No entanto, devido ao grande volume de tráfego gerado por essas aplicações, elas são frequentemente apontadas como as maiores responsáveis pela deterioração do desempenho experimentado por outras aplicações na rede. Provedores de Serviços de Internet (*ISP's*) têm tentado reduzir, sem muito sucesso, o tráfego P2P de seus clientes [13]. O bloqueio ou redução artificial do tráfego de usuários tem também atraído comentários negativos da mídia, direcionados aos ISPs [14, 15].

Embora a arquitetura simples e descentralizada tenha possibilitado o rápido crescimento da Internet, essa característica resultou também em limitados serviços oferecidos pelo sistema às aplicações. Algumas dessas limitações são:

- As aplicações não são informadas pela rede a respeito das medidas de desempenho (por exemplo, largura de banda disponível, atraso e taxa de perda) no caminho entre as duas máquinas;

- As aplicações também não sabem detalhes sobre as características do caminho de rede até a máquina remota. Não têm conhecimento da capacidade de transmissão dos enlaces ou tamanho da memória de armazenamento nas filas dos roteadores ao longo do caminho de rede, nem se a máquina remota está conectada à Internet por uma conexão de alta ou baixa capacidade de transmissão, ou mesmo se a largura de banda dos enlaces entre as estações finais satisfazem os requisitos daquela aplicação;

- O serviço oferecido é do tipo "melhor esforço". Não provê garantias de que os pacotes das aplicações com maior restrição de desempenho terão algum tipo de prioridade, em relação aos pacotes concorrentes gerados por aplicações elásticas. Nem mesmo há garantias de que os pacotes das aplicações serão entregues ao destino.

Soluções para garantir a qualidade de serviço da rede foram temas de inúmeras pesquisas em um passado recente. No entanto, problemas como a complexidade e o custo da implementação em larga escala impedem a implantação de serviços como Intserv[16] e Diffserv[17] em uma escala global na Internet. Garantias de serviço são oferecidos por provedores a clientes que tenham interesse em pagar pela reserva de recursos (por exemplo, taxas mínimas de transmissão e máximas de descarte), mas as garantias são apenas para dentro do próprio domínio daquela operadora. Os administradores de sistemas autônomos não têm controle sobre os recursos e nem conhecimento sobre as condições de desempenho fora de seus domínios. Por isso, a maioria das sessões de aplicações distribuídas executadas na Internet ocorrem sem reservas de recursos da rede e são regidas apenas pelo serviço de "melhor esforço".

1.2 Por que medir o desempenho da rede?

Em se tratando de aplicações distribuídas, o desempenho da rede é fundamental para a eficiência do funcionamento de algumas aplicações. Diferentes aplicações exigem distintos requisitos de desempenho da rede. Devido à inexistência na Internet de meios automáticos para garantir a reserva de recursos da rede, ou que ao menos forneçam informações sobre o desempenho da rede, realizar medições e analisar os

resultados de desempenho obtidos são tarefas fundamentais para algumas aplicações, além de importante também para usuários e provedores.

1.2.1 Algumas importantes medidas de desempenho da rede

Um grupo de trabalho formado pelo *IETF (Internet Engineering Task Force)*, denominado *IPPM WG (IP Performance Metrics Working Group)* [18], dedica-se ao estudo e à definição de importantes métricas de desempenho relacionadas à qualidade e confiabilidade das aplicações em redes. Algumas outras medidas de desempenho importantes, não definidas formalmente pelo *IPPM*, são amplamente utilizadas na literatura. Aqui estão as definições para algumas das principais medidas de desempenho em redes de computadores:

Atraso

Trata-se de uma classe de medidas de desempenho que representa o tempo necessário para uma informação ser transmitida e se propagar pela rede. As três medidas de desempenho utilizadas para avaliar o atraso na rede são: o *Atraso em um sentido (ou unidirecional)*, que é o tempo que um pacote leva para percorrer um caminho de rede entre a origem e o destino; o *Atraso de ida-e-volta*, que é o tempo que leva para um pacote percorrer o caminho de ida até uma máquina receptora e retornar à máquina de origem; e *Variação do atraso (Jitter)*, que é a diferença entre o intervalo da chegada de dois pacotes consecutivos e o intervalo das respectivas transmissões.

Capacidade

A capacidade também representa uma classe de medidas de desempenho. Essa classe está relacionada à habilidade do sistema de transmitir dados pela rede. Diversas medidas de desempenho estão relacionadas a essa classe, algumas delas são: *Largura de banda disponível*, que é a fração não utilizada da capacidade de um enlace, ou dentre todos os enlaces ao longo do caminho, dependendo do objetivo final da medida; *Vazão (*Throughput*)*, que representa o número total de pacotes enviados em um determinado intervalo de tempo; e, *Capacidade de transmissão em redes wireless*. Apesar de entendermos que a capacidade de transmissão de um enlace cabeado não possa ser definido como uma medida de desempenho, mas sim como uma car-

acterística da rede, no caso de uma rede sem fio a consideramos como tal. Essa definição justifica-se pelo fato de que a capacidade de transmissão adotada por um dispositivo 802.11 pode variar a depender das condições no meio de propagação (tais como, relação sinal ruído e taxa de colisão).

Tempo de *download*

É o tempo necessário para que um usuário receba por completo um determinado conteúdo (um arquivo, por exemplo). O tempo de *download* de um arquivo está diretamente associado à métrica vazão. Por exemplo, para uma transferência de dados feita por fluxo TCP, o tempo de *download* é igual a $\frac{S}{T}$, onde S é o tamanho do arquivo e T a vazão alcançada pela conexão TCP. A vazão decorrente do fluxo TCP é também uma medida de desempenho muito utilizada, em geral chamada de *BTC (Bulk Transfer Capacity)*.

Perda (*descarte*)

Três medidas de desempenho são associadas à perda de pacotes em redes de computadores: (i) Taxa de perda, representada pela fração do número de pacotes perdidos em relação ao total de pacotes enviados em um intervalo de tempo; (ii) Distribuição de perdas consecutivas, que estima a distribuição do número total de pacotes perdidos em sequência.

Utilização

É a razão do tempo em que um determinado serviço esteve ocupado dividido pelo tempo total de observação. Essa medida pode ser computada para qualquer serviço desejado. O serviço pode ser, por exemplo, um enlace de comunicação. Neste caso, a utilização representa a fração de tempo em que o enlace esteve ocupado transmitindo dados.

Disponibilidade

É o percentual de tempo que um determinado serviço fica disponível em relação ao tempo total de observação. Por exemplo, a disponibilidade de um arquivo, oferecido por um sistema P2P, é dada pela fração de tempo em que todo o conteúdo deste

arquivo (isto é, 100% das partes deste arquivo), esteve disponível para *download* dos usuários interessados.

1.2.2 Exemplos práticos para o uso das medidas...

... por aplicações:

- *Adaptação automática às condições de desempenho da rede:* aplicações multimídia podem, por exemplo, estimar a largura da banda disponível na rede e ajustar as taxas de envio de dados ou alterar a codificação de áudio e vídeo de suas transmissões. O Skype, por exemplo, implementa um algoritmo próprio de controle de congestionamento que tenta ajustar a taxa de transmissão de dados de sua aplicação à largura de banda disponível na rede [19]. O FreeMeeting oferece ao usuário diferentes opções de *codecs* de áudio com o objetivo de alcançar a melhor qualidade possível para seus usuários [7]. Ajuste das taxas de codificação (ou nos algoritmos de congestionamento implementados pela aplicação) podem também levar em consideração a taxa de transmissão do enlace do cliente, quando este estiver conectado à Internet por meio de uma rede local sem fio [20]. Informações sobre o atraso em um sentido e a taxa de perda dos pacotes também podem ser úteis para que as aplicações multimídia de tempo real ajustem seus mecanismos de codificação e/ou correção de erro [21, 22];

- *Escolha de rotas overlay:* Skype e outras aplicações P2P formam redes *overlays* e utilizam máquinas de outros usuários da aplicação como retransmissores (*relay*) para encaminhar os pacotes da aplicação, quando a comunicação direta entre as duas máquinas originais não é possível ou apresenta qualidade inferior [23]. Como o atraso unidirecional é uma métrica fundamental para a eficiência da interatividade de aplicações VoIP, as escolhas das rotas *overlay* podem levar em consideração os resultados dessa métrica de desempenho.

... por usuários:

- *Criação e validação de modelos:* os resultados de medições são constantemente utilizados para auxiliar na modelagem de sistemas. Em [21], por exemplo,

medidas obtidas na Internet foram usadas para validar um modelo de previsão de perdas de pacotes e avaliar o desempenho do algoritmo de correção de erro em aplicações VoIP. Experimentos de medições para computar o tempo de *download* e a disponibilidade de arquivos medidos em *swarms* do BitTorrent foram usados para validar modelos analíticos em [24, 25, 26];

- *Escolha das aplicações (ou equipamentos):* diferentes condições de desempenho da rede podem justificar o uso e a aquisição de uma aplicação (ou equipamento). Medir o desempenho da rede pode auxiliar usuários a tomarem decisões mais adequadas;

- *Verificação de cumprimento dos acordos de serviços:* clientes podem utilizar ferramentas de medições para verificar o cumprimento, por parte dos provedores, dos acordos de serviços (e vice-versa, provedores podem monitorar clientes para comferir cumprimento de contratos).

... por provedores:

- *Identificar e implementar soluções para problemas na rede:* é comum o uso de medições por parte dos provedores para identificar eventuais problemas ou pontos de falha na rede. Ferramentas como Ping [27] e Traceroute [28], que medem o atraso e a taxa de perda, são amplamente utilizadas por provedores nessa tarefa [29, 30, 31]. Sistemas distribuídos de larga escala, como o iPlane[32] e Hubble [33, 34], também usam medidas de desempenho (como latência, largura de banda disponível e taxa de perda) estimadas entre diversos pontos da rede para criar um grande mapa de desempenho da Internet e, possivelmente, auxiliar na identificação de problemas como *buracos negros*[2] [34];

- *Melhorar o desempenho em redes locais sem fio:* a existência de dispositivos operando a uma taxa de transmissão muito baixa em uma rede local sem fio pode comprometer a qualidade dos demais usuários da WLAN. Através de

[2]Em [33, 34], os autores definem *buraco negro* como sendo uma região da rede com problemas de alcançabilidade na Internet. Embora existam rotas anunciadas pelo BGP até essas regiões críticas, pacotes originados de alguns diferentes pontos da Internet se perdem ao longo do caminho.

medições, os administradores de rede podem identificar esses casos e tomar as devidas providências para evitar a degradação da qualidade da rede. Problemas desse tipo são tratados em [35, 36], por exemplo;

- *Dimensionar a rede:* previsão de tráfego permite que provedores evitem saturamento dos seus enlaces, possibilitando um planejamento antecipado da capacidade da sua rede, evitando também uma degradação na qualidade do serviço oferecido aos seus clientes [37, 38]. Através da análise do tráfego, por exemplo o histórico do tráfego de acesso a um conjunto de servidores, é possível dimensionar apropriadamente a rede em estudo, prevendo os recursos necessários para manter o serviço oferecido dentro dos limites desejáveis;

- *Reduzir custos:* medidas de desempenho como disponibilidade e tempo de *download* são de grande utilidade para provedores que usam os sistemas P2P para distribuição de seus conteúdos. Para conteúdos com alta disponibilidade, provedores podem reduzir seus custos (com diminuição do tráfego em seus enlaces e menor gasto de energia em seus servidores) deixando a tarefa de disseminar o conteúdo por conta dos clientes do sistema P2P, sem afetar o tempo de *download* do usuário [24].

1.3 Motivações, objetivos e contribuições da tese

O desenvolvimento de técnicas de medições que permitam conhecer melhor as características da rede e a análise do desempenho de aplicações na Internet sob diferentes perspectivas são dois importantes tópicos de pesquisa, dentro da comunidade de redes na atualidade. Apenas através das medições é possível estimar as características de desempenho da rede. Embora algumas métricas sejam triviais de serem obtidas, outras medidas requerem algoritmos e/ou dispositivos sofisticados para serem estimadas. A análise experimental da operação das aplicações na Internet permite também compreender melhor o estado atual da rede e, possivelmente, melhorar o desempenho do serviço oferecido.

Os objetivos definidos nesta tese são: (i) desenvolver novas técnicas de medições para estimar algumas métricas de desempenho fundamentais para o funcionamento de aplicações em redes; (ii) analisar, por meio de medições em larga escala, o desem-

penho de aplicações para disseminação de conteúdo na Internet e identificar soluções eficientes para aumentar a disponibilidade do conteúdo e/ou reduzir o custo para os provedores. Escolhemos nesta tese uma aplicação alvo: a aplicação peer-to-peer BitTorrent.

Esses objetivos foram alcançados com as seguintes contribuições:

1. **Uma técnica de medição ativa não cooperativa para estimar a média e a variância da distribuição do atraso em um sentido na Internet;**

   Computar o atraso unidirecional dos pacotes na rede não é trivial, pois requer algoritmos sofisticados caso as máquinas envolvidas na medição não possuam seus relógios perfeitamente sincronizados. O problema torna-se ainda mais complexo quando o analista não tem acesso à máquina localizada no final do caminho. O primeiro conjunto de contribuições desta tese está relacionado à proposta de uma nova técnica de medição ativa que lida com ambos os problemas (falta de acesso e falta de sincronismo), permitindo que um analista estime a média e a variância da distribuição do atraso em um sentido. Para contornar o problema da falta de acesso à máquina remota, foram desenvolvidas duas variações da técnica, uma faz uso do campo IPID do cabeçalho de pacotes IP e a outra utiliza spoofing *dos pacotes IP. É possível destacar a validação exaustiva nesta etapa do trabalho: a avaliação da técnica desenvolvida através de simulações; resultado de experimentos reais executados na Internet para avaliação e validação dos algoritmos; e, a análise quantitativa do erro causado pelo método.*

2. **Uma técnica para estimar a taxa de transmissão de enlaces em uma rede local sem fio IEEE 802.11;**

As redes locais sem fio (WLANs), baseadas nos padrões IEEE 802.11 [39], têm se tornado uma das formas mais populares de acesso à Internet. As taxas de transmissão alcançadas pelos padrões 802.11a/b/g [40] podem variar de valores relativamente altos (54Mbps) até valores significativamente muito baixos (1 ou 2Mbps), dependendo das características do meio de propagação. O segundo conjunto de contribuições desta tese refere-se ao desenvolvimento de uma técnica simples e acurada para estimar a taxa de transmissão (capacidade em bits por segundo) do enlace no último salto em um caminho de rede, quando este encontra-se conectado à Internet através de uma rede local sem fio IEEE 802.11. A técnica consiste em uma extensão do método tradicional de pares de pacotes, adaptado para computar a dispersão dos pacotes decorrente da capacidade de transmissão do enlace no último salto. A técnica leva em consideração aspectos como o overhead *causado pelo protocolo IEEE 802.11, a existência de tráfego concorrente, a possibilidade de enlaces de capacidade inferior ao longo do caminho de rede e a variação automática da taxa de transmissão do enlace sem fio. A análise de resultados obtidos por simulações e experimentos realizados na Internet, utilizados para validar a técnica, destacam-se também como contribuições desta tese.*

3. **Estudo de soluções para aumentar a disponibilidade e reduzir o custo na distribuição de conteúdo através de aplicações peer-to-peer na Internet.**

O uso de sistemas P2P para disseminação de conteúdo tem algumas vantagens bem conhecidas em comparação ao método mais tradicional utilizando uma aplicação cliente/servidor. Tais sistemas contam com a capacidade não utilizada dos clientes da rede para possibilitar uma economia de banda do servidor, um tempo menor de download para o usuário e uma maior escalabilidade para a aplicação. No entanto, outras questões são inerentes a essa arquitetura: arquivos pouco populares têm problemas de indisponibilidade no sistema e a disseminação de conteúdos muito populares continuam sendo extremamente custosos para servidores de conteúdo. Experimentos realizados utilizando o BitTorrent nos levou a duas descobertas no mínimo intrigantes a respeito da disseminação de conteúdo na Internet por meio de sistemas P2P. A primeira é que distribuir arquivos agrupados (por exemplo, todos os arquivos agrupados em um único ZIP ou em um ISO) pode aumentar significativamente a disponibilidade dos arquivos e até mesmo reduzir o tempo total de download de conteúdo. A segunda descoberta é a possibilidade de reduzir a (quase) zero o custo de um servidor para disseminar conteúdos muito populares, isso sem afetar o desempenho (tempo de download) para o usuário. O terceiro conjunto de contribuições desta tese são os seguintes: (i) uma análise, através de simulações do protocolo BitTorrent, sobre as implicações da popularidade de um conteúdo na sua disponibilidade entre os Leechers do swarm, custo para disseminação e desempenho experimentado pelos usuários; (ii) avaliação experimental dos benefícios da prática de agrupamento de arquivos na disseminação de conteúdo, que comprovam a possibilidade de aumentar significativamente a disponibilidade e reduzir o tempo total de download do conteúdo se os arquivos foram distribuídos na forma agrupada; e, (iii) estudo de soluções para reduzir a (quase) zero os custos para um provedor disseminar um conteúdo através de sistemas P2P.

1.4 Organização da tese

Os demais capítulos desta tese estão estruturados da seguinte forma. O Capítulo 2 discorre sobre a revisão bibliográfica dos trabalhos relacionados. O Capítulo 3 apresenta as soluções de técnicas não cooperativas para estimar a média e a variância da distribuição do atraso de pacotes em um único sentido. No Capítulo 4 é apresentada a técnica para estimar a taxa de transmissão em uma rede local sem fio 802.11. O Capítulo 5 apresenta a análise sobre as implicações da popularidade de *swarms* P2P, as validações do aumento da disponibilidade com a disseminação de arquivos agrupados, e as soluções de redução do custo para a distribuição de conteúdos. O Capítulo 6 aborda as considerações finais desta tese, com um sumário das principais contribuições, além de algumas deliberações sobre problemas em aberto e possíveis trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

NESSE capítulo é apresentada uma revisão bibliográfica das técnicas de medição em redes (2.1) e dos trabalhos de avaliação de desempenho de aplicações P2P para a disseminação de conteúdo na Internet (2.2).

2.1 Uma revisão sobre medição em redes

Um dos primeiros trabalhos de medição em larga escala na Internet foi desenvolvido por Vern Paxson em 1997 [41]. No trabalho, Paxson apresentou uma infra-estrutura de monitoramento e novas técnicas de medições. Na ocasião, mais de 20.000 conexões TCP foram monitoradas e as coletas foram utilizadas para analisar medidas de desempenho na Internet. O estudo desenvolvido revelou o dinamismo de medidas de desempenho relacionadas ao atraso, à perda e à capacidade das conexões fim-a-fim na rede. Desde então, novas técnicas, infraestruturas e estudos de medições de desempenho têm sido temas de inúmeras pesquisas na comunidade de redes [42].

Devido à vasta bibliografia existente na literatura, a revisão apresentada nesta seção limita-se à descrição dos trabalhos de maior relevância para as principais contribuições desta tese. Além de fundamentos básicos em medições, serão abordados os métodos de medições que formam o estado da arte em soluções para estimativa do atraso unidirecional e inferência da capacidade de transmissão em redes locais sem fio. Revisões mais amplas sobre trabalhos relacionados a outras medidas de desempenho estão presentes em [43, 44].

2.1.1 Fundamentos básicos de medições

Os métodos de medições existentes são classificados como *ativos*, *passivos* ou *híbridos*. Nos métodos *passivos*, o tráfego enviado por aplicações em execução na rede é observado em pontos de medição, muitas vezes com o auxílio de equipamentos apropriados (tais como, placas DAG[45], dispositivos Ipoque[46] ou AirPcap[47]) e/ou *softwares* específicos (por exemplo, Tcpdump[48], Wireshark[49] ou Netflow[50]). Em alguns casos, informações geradas pelas próprias aplicações ou protocolos de rede podem ser utilizadas pelos métodos de medição, dispensando neste caso a necessidade de outros equipamentos ou *softwares* específicos. Na forma *ativa*, um tráfego extra de pacotes de controle, denominados sondas ou *probes*, é inserido na rede. As sondas são enviadas a partir de máquinas fontes escolhidas e coletadas, após percorrer um caminho de rede, pelas próprias fontes ou por uma ou mais máquinas receptoras. Mais recentemente, foram propostas algumas técnicas *híbridas* de medições nas quais informações obtidas passivamente são utilizadas para a execução de medições ativas [33]. Nas três formas de medição, após a coleta das informações obtidas do tráfego observado passivamente ou das sondas extras injetadas na rede, algoritmos especiais são aplicados às coletas para extrair as medidas de desempenho desejadas.

Existem vantagens e desvantagens quando comparadas as formas passiva e ativa de medição. Enquanto a forma passiva permite obter medidas, sem gerar uma sobrecarga na rede com pacotes de controle, a ativa oferece maior flexibilidade aos métodos de medição. Determinadas métricas de desempenho da rede só são possíveis de serem estimadas quando utilizadas técnicas ativas. Isso porque, os algoritmos apropriados, aplicados às coletas para estimar as medidas de interesse, requerem que os pacotes tenham tamanhos predefinidos e sejam transmitidos em intervalos de tempo específicos. Dois exemplos são os algoritmos para estimar o atraso unidirecional e para computar medidas de capacidade. Detalhes desses algoritmos serão discutidos mais adiante (nas Subseções 2.1.3 e 2.1.4).

Em geral, as medições têm como propósito caracterizar o desempenho de apenas um enlace da rede ou de um caminho de rede entre dois pontos. No primeiro caso, a métrica em questão representa o desempenho de um equipamento em um ponto específico da rede, como por exemplo a taxa de perda ou a largura de banda

disponível de um enlace. No segundo caso, a métrica refere-se não a um enlace específico, mas sim ao caminho fim-a-fim existente entre os pontos de medição, formado por dois ou mais enlaces.

Na Internet atual, os caminhos de ida e volta entre duas máquinas podem ser assimétricos. Isto é, as capacidades dos roteadores em um sentido podem ser diferentes das capacidades dos roteadores no sentido oposto, ou ainda, as sequências de roteadores percorridos em cada direção podem ser distintas. Mesmo quando a seqüência de roteadores for a mesma e a capacidade deles simétrica, os caminhos podem apresentar características de desempenho completamente diferentes devido à assimetria do tráfego (e consequentemente do tamanho das filas) dos roteadores. Por isso, medir os caminhos de forma independente permite identificar o desempenho da rede em cada um dos sentidos.

As técnicas de medições fim-a-fim se distinguem, então, quanto à habilidade de estimar o desempenho do caminho "em um único sentido" ou do caminho de "ida e volta" percorrido pelos pacotes na rede. Quando as máquinas de origem e destino das sondas são distintas (ou na forma passiva de medição, quando o tráfego enviado pela aplicação é monitorado tanto na origem quanto no destino dos pacotes), a medida de desempenho é computada "em um único sentido", também chamada de unidirecional. No caso em que as sondas enviadas não são coletadas pela máquina alvo e sim replicadas de volta à máquina de origem (ou na medição passiva, caso os pacotes de solicitações e respostas, enviados e recebidos pelas aplicações, sejam monitorados apenas na máquina de origem), a métrica estimada refere-se ao desempenho no caminho de "ida e volta" percorrido pelos pacotes.

Medir o desempenho no caminho de "ida e volta", quando comparado à medição unidirecional, em geral, é mais simples. Estimar o atraso e a taxa de perda na ida e volta dos pacotes na rede, por exemplo, é trivial utilizando ferramentas de medições ativas como o Ping. Isso porque, é comum nas máquinas conectadas à Internet estar habilitada a função de *ICMP echo reply* em resposta ao recebimento de um *ICMP echo request*[51]. Na forma passiva é também possível computar essas métricas apenas monitorando os pacotes de solicitação e respostas pertencente aos fluxos TCP em uma única máquina, por exemplo. No entanto, as técnicas de medição existentes para computar medidas como atraso, largura de banda disponível e taxa de perda em

um sentido normalmente necessitam da execução de processos na máquina remota. Informações como chegadas com sucesso e instantes de chegada dos pacotes devem ser coletadas na máquina de destino, para que os algoritmos definidos pelas técnicas de medição possam estimar as métricas "em um único sentido".

Recentemente, pesquisas têm sido dedicadas ao desenvolvimento de novas técnicas de medição que possibilitem estimar as características de desempenho dos caminhos de rede em um único sentido, sem a necessidade de privilégios especiais de acesso a uma máquina remota. Denominadas técnicas de medições não cooperativas, elas compensam a falta de acesso à máquina remota, para coleta de informações sobre a chegada dos pacotes, explorando características do protocolo IP. As técnicas de medição não cooperativas são, particularmente, de grande relevância para uma das contribuições desenvolvidas nesta tese e serão discutidas na próxima seção (2.1.2).

2.1.2 Técnicas de medição não cooperativas

Medições com restrição de acesso à máquina remota é uma questão que tem sido contornada por novas técnicas explorando características inerentes ao protocolo IP. Por exemplo, a partir de informações contidas no campo de identificação do cabeçalho IP *(IPID)* de pacotes *ICMP echo reply* enviados por uma máquina alvo qualquer da Internet, propostas existentes possibilitam computar a taxa de perda em um sentido [52, 53], a fração da chegada de pacotes fora de ordem em um caminho unidirecional [52, 54], e as diferenças entre os atrasos de duas máquinas fonte para uma máquina alvo [55]. Outras propostas utilizam IP *spoofing*[1] para lidar com a falta de acesso a um dos pontos de medição na estimativa do desempenho da rede [56, 57].

Explorando o IPID em medições não cooperativas

O IPID é um campo de identificação existente no cabeçalho de pacotes do protocolo IP [58]. Este campo fornece uma identificação que é utilizada pelo processo de fragmentação e remontagem de datagramas na Internet. Ocupando 16-bits do cabeçalho IP, este identificador, juntamente com outras informações contidas também no cabeçalho IP, possibilitam a remontagem dos datagramas que tenham sido fragmentados para transmissão.

[1] IP *spoofing* consiste no envio de pacotes IP utilizando endereços de remetentes falsificados.

Embora a utilização do IPID na fragmentação e remontagem dos datagramas seja um padrão na Internet, o padrão não define uma regra quanto ao uso do identificador. A forma como os valores de identificação do datagrama IP são incrementados, por exemplo, depende da implementação do sistema operacional. Diversos sistemas programam o IPID como um simples contador global. Isso inclui as máquinas servidas com sistemas operacionais Windows, Freebsd, Mac OS e Linux até a versão 2.2 do kernel. As versões mais atuais do Linux, Solaris e Openbsd implementam um contador pseudo-aleatório para cada fluxo.

Um simples experimento, com sondas geradas de duas máquinas fonte quaisquer para uma mesma máquina alvo remota, permite identificar que tipo de implementação no IPID é utilizada pelo sistema operacional deste alvo. A Figura 2.1 ilustra dois logs obtidos com a ferramenta Tcpdump executada no roteador de saída da rede do laboratório LAND[2]. (Para possibilitar o registro do campo IPID no log do Tcpdump, sondas foram geradas com tamanho superior a 1480 *bytes*, forçando a fragmentação dos datagramas na fonte.) O primeiro log mostra pacotes de *ICMP echo reply* destinados a duas máquinas diferentes, em resposta a sondas de *ICMP echo request*, previamente enviadas à uma máquina com sistema operacional Windows XP. O outro log mostra os pacotes *echo reply* gerados por uma máquina com sistema operacional Linux de kernel 2.6. No primeiro log, é possível verificar o crescimento global dos valores do IPID gerados pela máquina remota. Já no segundo log, existe um crescimento apenas nos valores do IPID relativos a cada fluxo. (Por uma questão de segurança, os nomes reais das máquinas foram aqui substituídos por nomes fictícios.) Ferramentas para auditoria de segurança de rede utilizam técnicas semelhantes que exploram essa característica do IPID para identificar, em uma máquina remota, o seu sistema operacional [59] ou detectar ocorrências de ataques de *port scan* [60].

Outros trabalhos têm explorado os valores coletados do campo IPID para a obtenção de características da rede. Em [55] é apresentado um estudo de técnicas de inferência de várias medidas com uso do IPID. No artigo, os autores definem três categorias de aplicações para as técnicas existentes: medição de atividade do tráfego

[2]O Laboratório de Modelagem/Análise e Desenvolvimento de Sistemas de Computação e Comunicação (LAND) está localizado no Programa de Engenharia de Sistemas e Computação da COPPE, na Universidade Federal do Rio de Janeiro (UFRJ) - `http://www.land.ufrj.br`

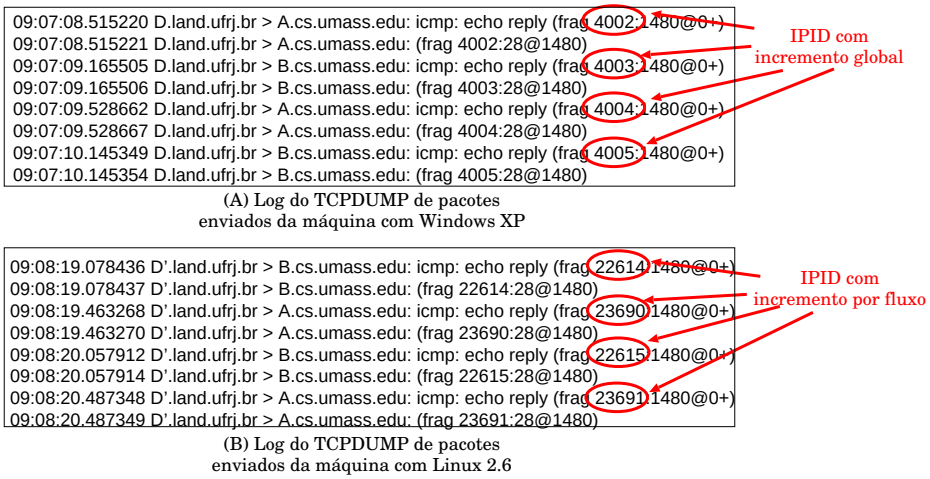


Figura 2.1: Logs do Tcpdump executado no roteador de saída da rede.

[60]; agrupamento de fontes [61, 60]; e, identificação de perda, duplicação e chegada fora de ordem [52, 54]. Além desta classificação, os autores de [55] ainda propõe três novas técnicas para o uso do IPID, uma referente a cada classe definida.

Observando a variação do IPID de sondas recebidas por uma máquina fonte, é possível estimar o tráfego em um dado intervalo de tempo, desde que a máquina destino implemente um contador global para o IPID. Em [60], é apresentada uma proposta para estimar o volume de tráfego para um servidor através de medições ativas. Sondas são enviadas para a máquina alvo e capturados os IPIDs dos pacotes de resposta. Seja $IPID(i)$ o valor de IPID capturado da sonda i e $T(i)$ o instante de chegada destas respostas. O número de requisições recebidas por um servidor, entre os instantes de tempo $T(i)$ e $T(i+1)$, é igual a $\Delta IPID(i)$ e equivale à diferença dos valores $IPID(i)$ e $IPID(i+1)$. Como o campo IPID possui um tamanho máximo de 16 bits, essa e outras técnicas que explorem o campo IPID devem levar em consideração que o incremento do valor deste identificador retorna a zero ao atingir 2^{16}.

Uma abordagem semelhante à [60] é apresentada em [55] para estimar o volume de tráfego de um servidor. A diferença entre as propostas [60] e [55] é que a segunda técnica utiliza medição híbrida, ao invés de medição ativa, para observação do IPID gerado pelo servidor medido. A vantagem deste método, em relação ao anterior, é a redução significativa da sobrecarga na rede, uma vez que boa parte dos pacotes utilizados para computar a medida de interesse são coletados passivamente do roteador de saída da rede. Em contrapartida, é necessária permissão para execução

de uma aplicação para a coleta de pacotes no roteador do canal de saída da rede deste servidor. Além disso, no método apresentado em [55], sondas extras ainda são enviadas para lidar com o problema de retorno a zero do contador de IPID e, por isso, a técnica é classificada como híbrida.

O campo IPID foi explorado também em propostas para identificar o número de servidores utilizados por um sistema de balanceamento de carga [55, 60] e o número de máquinas por detrás de um serviço *NAT(Network Address Translator)* [61]. Os métodos supõem que dois pacotes gerados por uma mesma máquina em um curto intervalo de tempo devem apresentar um valor pequeno para o $\Delta IPID$. Se cada servidor do sistema de balanceamento de carga possui um contador global independente, pacotes gerados por um servidor possuem uma sequência do IPID diferente da sequência dos pacotes gerados por outro servidor. Observando valores coletados do IPID, as técnicas de [60, 55] tentam identificar essas independências entre as sequências e estimar o número de servidores utilizados para o balanceamento de carga. Embora essa técnica tenha sido sugerida em [60], apenas em [55] foi apresentado um algoritmo apropriado para estimar o número de servidores. Técnica semelhante é utilizada em [61] para detectar máquinas utilizando servidores NAT para acesso à Internet e contabilizar o número de máquinas em atividade utilizando um mesmo servidor.

Recentemente, alguns trabalhos propuseram novas técnicas que possibilitam medir características de desempenho da rede, a partir dos valores de IPID existentes nos pacotes recebidos de uma máquina alvo. Essas técnicas permitem identificar, dentre outras medidas, a taxa de perda e chegadas fora de ordem [52, 54]. Embora as sondas utilizadas pelas técnicas sejam geradas e coletadas na mesma máquina, os valores do IPID obtidos da máquina remota permitem a estimativa destas métricas em cada um dos sentidos. Em geral, essas técnicas utilizam mensagens de *ICMP echo request e reply*.

Para compreender como é possível identificar a ocorrência e o sentido da reordenação de dois pacotes, considere os possíveis casos ilustrados na Figura 2.2. Se dois pacotes ($P1$ e $P2$, por exemplo), enviados por uma máquina fonte para uma máquina alvo (denotadas na figura como máquinas A e D, respectivamente), não foram reordenados em qualquer um dos sentidos, o pacote replicado de $P1$ deve

chegar à máquina A antes de $P2$ e o valor do IPID da resposta de $P1$ deve ser inferior à de $P2$, como mostra a ilustração (A) da Figura 2.2. (Obviamente, desconsiderando a questão do retorno a zero, após alcançado o valor máximo do campo IPID.) No entanto, se a resposta de $P2$ apresentar valor de IPID inferior e chegar à máquina A primeiro que a resposta de $P1$, isso indica que houve uma reordenação no sentido de ida dos pacotes (vide ilustração (B) da Figura 2.2). Caso, a resposta de $P2$ chegue antes da resposta de $P1$, mas com o valor de IPID maior, isso caracteriza uma reordenação no sentido de volta. E, por fim, se o pacote replicado de $P1$ chegar antes da resposta de $P2$. porém com o IPID superior ao de $P2$, como mostra ilustração (D) da Figura 2.2, isso significa que os pacotes foram reordenados tanto no sentido de ida, quanto no sentido de volta. Esse algoritmo foi proposto em [52] para identificar, numa coleta das sondas, as reordenações ocorridas em cada um dos sentidos.

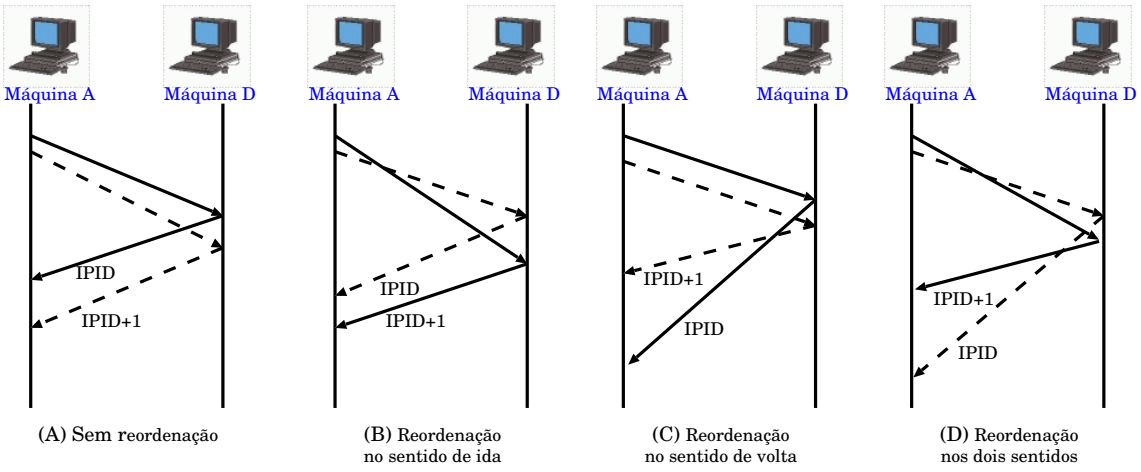


Figura 2.2: Detecção do sentido da reordenação.

Abordagem semelhante foi utilizada em [52] para determinar o sentido (caminho de ida ou de volta) da ocorrência de uma perda, explorando também os valores do IPID contidos nas sondas replicadas pela máquina remota. Para detectar o sentido da perda de uma sonda, são observados os valores do IPID de outras sondas recebidas com sucesso e que foram enviadas da mesma origem em instantes próximos de tempo.

Suponha que não tenha chegado à máquina A a resposta da n-ésima sonda, de uma série enviada da máquina fonte A para a máquina alvo D. A técnica proposta para identificar o sentido da perda analisa o IPID recebido nas respostas das sondas enviadas exatamente antes e exatamente depois a essa n-ésima sonda. Se os valores

IPID das respostas de n-1 e n+1 forem consecutivos, então a perda ocorreu no sentido de ida; caso contrário, a perda da n-ésima sonda ocorreu no caminho de volta. A Figura 2.3 ilustra esses dois casos.

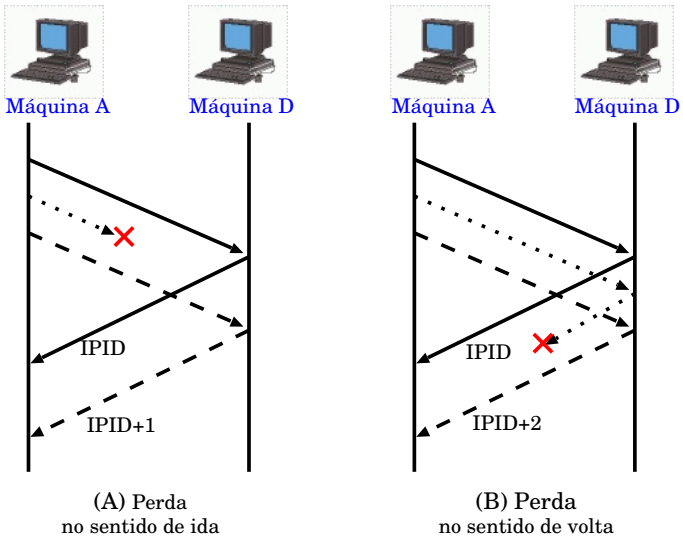


Figura 2.3: Detecção do sentido da perda.

Em [55] foi proposta uma técnica para determinar a diferença entre os atrasos de sondas enviadas de máquinas fontes para uma máquina alvo. As Figuras 2.4(a) e (b) ajudam a compreender a técnica. Considere duas máquinas A e B, com relógios sincronizados por GPS, gerando sondas para uma máquina remota D a intervalos constantes iguais a δ_A e a δ_B, respectivamente, sendo $\delta_B \ll \delta_A$. Um pacote enviado por A chegará à máquina D entre dois pacotes consecutivos de B. Ao receber os pacotes, a máquina alvo, que não precisa estar com seu relógio sincronizado com as demais, replica as sondas imediatamente para as máquinas de origem, incluindo no campo IPID os valores referentes ao contador global desta máquina. Intuitivamente, se uma sonda enviada por A retornou à máquina de origem com um valor de IPID entre os valores deste campo de duas sondas enviadas por B, então a sonda de A chegou em um instante de tempo entre as duas sondas de B, como ilustra a Figura 2.4(a).

Seja n_A (n_B) o número total de sondas enviadas por A (B) desde o instante inicial de geração τ_A (τ_B). Suponha que o n_A-ésimo pacote enviado por A chegue a D entre os pacotes n_B e n_B+1 enviados por B. Sejam d_{AD} e d_{BD} os atrasos experimentados pelos pacotes de A para D e de B para D, respectivamente. Então, conforme

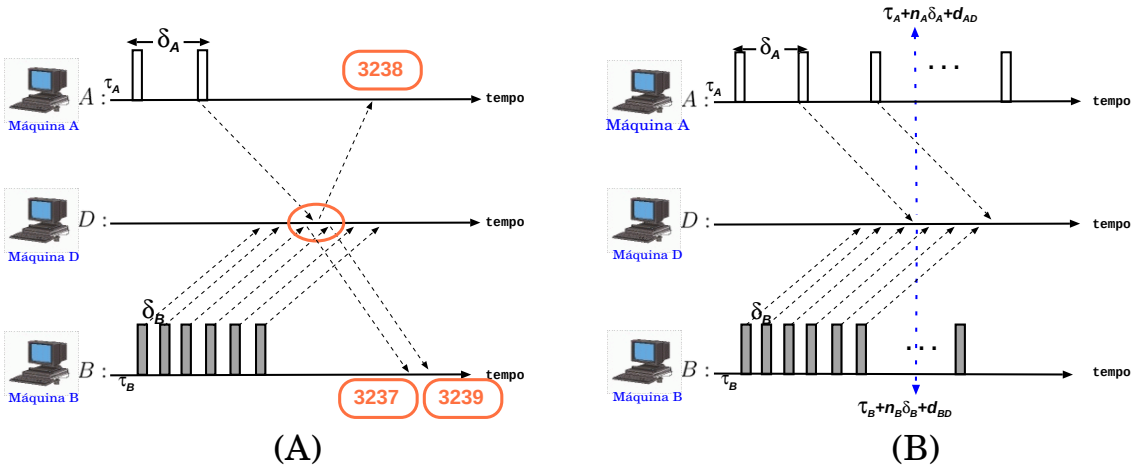


Figura 2.4: Técnica para determinar a diferença entre os atrasos de sondas enviadas de máquinas fontes para uma máquina alvo.

definido em [55] e ilustrado na Figura 2.4(b), $\tau_B + d_{BD} + n_B\delta_B \leq \tau_A + d_{AD} + n_A\delta_A \leq \tau_B + d_{BD} + (n_B + 1)\delta_B$.

Note que os limites máximo e mínimo dependem de δ_B. Logo, quanto menor o valor de δ_B mais estreita é a diferença entre os limites inferior e superior. Dessa forma, para δ_B pequeno, a diferença entre os atrasos em um sentido pode ser estimada pelos instantes de envio das sondas:

$$d_{AD} - d_{BD} \approx \tau_B - \tau_A + n_B\delta_B - n_A\delta_A. \tag{2.1}$$

Usando IP Spoofing em medições não cooperativas

O padrão definido para o protocolo IP não prevê autenticação dos pacotes encaminhados na rede. Portanto, os roteadores na Internet encaminham os pacotes independente do endereço IP de origem contido nos cabeçalhos. Assim, nada impede que uma aplicação inclua arbitrariamente um IP falso no campo de endereço de origem no cabeçalho do pacote e transmita-o pela Internet. Independente do valor presente no campo de origem do endereço IP, esse pacote será encaminhado normalmente ao longo do caminho de rede e entregue à máquina endereçada pelo IP de destino contido no pacote. O artifício de incluir endereços falsos nos pacotes transmitidos pela rede é chamado de *IP spoofing*.

O *IP spoofing* é amplamente utilizado em conjunto com outras técnicas de ataque na Internet, como por exemplo os ataques de DOS (*denial-of-service)*, para ocultar

a verdadeira fonte da operação maliciosa. No entanto, recentemente esse artifício passou a ser utilizado também em técnicas de medição ativa.

Em [56], por exemplo, a operação de *IP spoofing* é utilizada por uma técnica para estimar a taxa de perda unidirecional, em um caminho de rede que não pode ser medido diretamente. Suponha que o objetivo é computar a taxa de perda do caminho entre as máquinas A e B via roteador S, sendo que esse roteador S não é parte da rota original de A para B. Na solução apresentada por Zhao et. al[56], sondas de *ICMP echo request* são enviadas de A para S com o endereço de origem falso de B. As mensagens de *ICMP echo reply* são replicadas de S para B e a taxa de perda do caminho A-B via S pode então ser computado.

O *IP spoofing* é também utilizado em uma técnica para estimar o retardo introduzido pelos roteadores na geração de mensagens de controle *ICMP TE (Time Exceeded)*[57]. No padrão definido para o protocolo ICMP[51], mensagens TE são enviadas por roteadores em resposta a pacotes recebidos com o TTL (*Time to Live)* expirado. No entanto, alguns roteadores são configurados para retardar propositalmente essas mensagens de ICMP. Govindan e Paxson, em [57], definiram então um método que possibilita computar o retardo introduzido por roteadores antes de enviar as mensagens de *ICMP TE*. A técnica proposta utiliza *IP spoofing* nas sondas.

Para medir o retardo introduzido em um roteador R que encontra-se no caminho entre duas máquinas A e B pelo método em [57], pacotes são enviados por A contendo o endereço falso de origem B para a própria máquina B e com o TTL limitado a um valor que irá expirar em R. Os pacotes percorrem o caminho entre A e B, mas ao chegar em R têm o TTL expirado. Mensagens ICMP TE são geradas por R e, eventualmente, retardadas por ele antes de serem enviadas. Essas mensagens são endereçadas à máquina B devido ao endereço falso incluído por A na mensagem original. Ao chegar em B é possível computar o atraso unidirecional de A para B, somado ao retardo introduzido pelo roteador R à mensagem ICMP TE. Diminuído o atraso unidirecional de uma mensagem regular de ICMP, enviada de A para B, que não teve o TTL expirado em R, é possível estimar o retardo introduzido pelo roteador para o envio das mensagens de controle ICMP TE.

Soluções foram desenvolvidas com o objetivo de evitar operações de *IP spoofing* na Internet. Essas soluções são baseadas na instalação de filtros de ingresso ou

egresso de pacotes nos canais de acesso à rede. No entanto, as duas abordagens apresentam problemas. Resultados apresentados em [62] de experimentos em larga escala, executados na Internet, sugerem que uma grande parte das máquinas são vulneráveis a *IP spoofing*.

O método de filtragem de ingresso rejeita pacotes vindos de fora da rede e que tenham como endereço IP de origem um valor referente ao segmento de endereçamento pertencente à rede interna. A Figura 2.5(A) ilustra esse modelo de filtragem. O pacote enviado pela máquina A (cujo endereço IP real é 1.1.1.1) foi enviado para a máquina B (de endereço IP 2.2.2.2), fingindo ter sido gerado pela máquina S (com o endereço IP 2.2.2.1) que encontra-se na mesma rede de B. Nesse cenário, se a filtragem estiver sendo feita no ingresso, esse pacote será descartado antes de entrar na rede 2.2.2.0/24. No entanto, esse tipo de filtro não é eficiente, pois o atacante (no exemplo citado acima, a máquina A) pode contornar essa restrição, simplesmente, utilizando como endereço de origem o IP de um segmento de rede diferente da máquina alvo (por exemplo, 3.3.3.3).

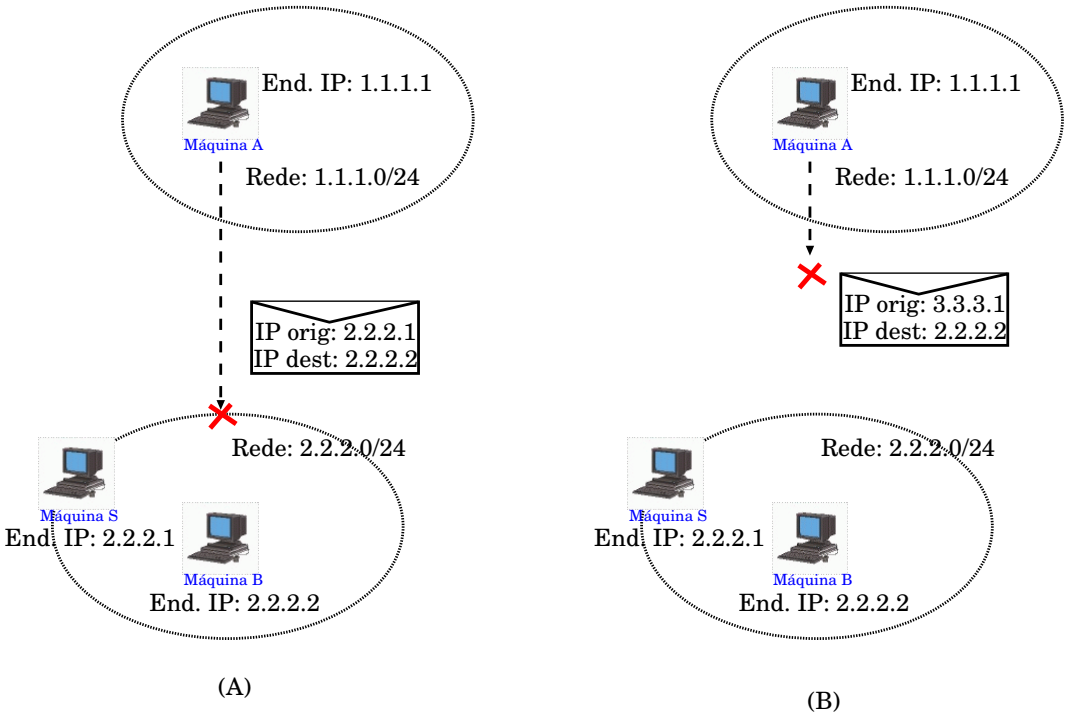


Figura 2.5: Filtragem de pacotes: (a) Ingresso; (b) Egresso.

A abordagem mais eficiente é o filtro de egresso. Nesse caso, os pacotes são descartados já pelos roteadores de saída da rede, caso o endereço IP de origem

seja diferente do segmento de rede ao qual pertence aquele roteador. No exemplo ilustrado Figura 2.5(B), o pacote enviado pela máquina A (cujo IP real é 1.1.1.1), contendo um endereço de origem falso (por exemplo, 2.2.2.1 ou 3.3.3.3), não será encaminhado para a Internet. Isso acontece porque, o filtro de egresso, localizado no roteador de saída daquela rede, descarta qualquer pacote que deva ser encaminhado para fora da rede e que tenha no campo IP de origem um endereço que não pertença ao segmento de rede 1.1.1.0/24. Apesar da eficiência, os filtros de egresso não são largamente implementados na Internet. Provedores e administradores não têm grande incentivo para habilitar um serviço que impõe certa sobrecarga em seus equipamentos e não traz qualquer proteção para a sua própria rede.

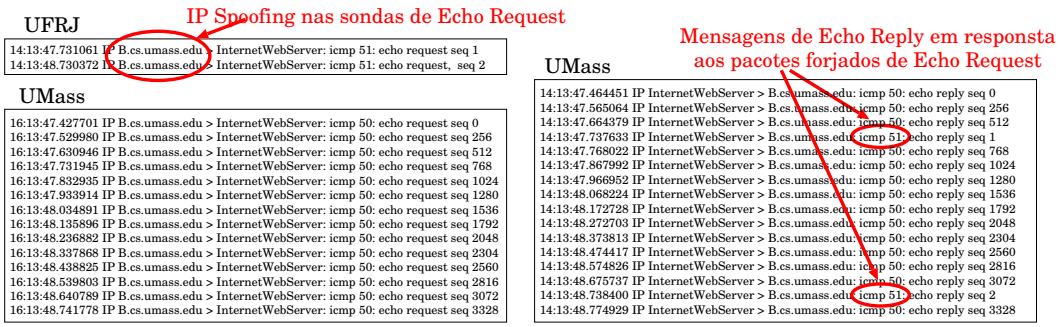


Figura 2.6: Logs obtidos rodando a ferramenta TCPDUMP nas máquinas da UFRJ e da UMass.

A Figura 2.6 mostra três logs obtidos com a ferramenta Tcpdump durante experimentos reais executados na Internet. No experimento, as máquinas A e B, localizadas, respectivamente, nos laboratórios LAND/UFRJ e CNRG/UMass-Amherst[3], enviam mensagens de *ICMP echo request* para um popular servidor Web da Internet. As mensagens de *echo request* enviadas estão registradas nos logs apresentados na Figura 2.6(A). O log coletado na UFRJ mostra que o *IP spoofing* foi feito pela máquina A (localizada na UFRJ), quando as sondas são enviadas com o endereço de origem da máquina B (localizada na UMass). Assim, todas as mensagens de *ICMP echo reply* geradas pelo servidor Web, em resposta às mensagens de *echo request* enviadas por A e B, foram direcionadas à máquina da UMass, como mostra o log

[3]Laboratório do grupo de pesquisa em redes da University of Massachusetts - Amherst `http://www-net.cs.umass.edu`

da Figura 2.6(B). Diferentes tamanhos foram definidos para as mensagens geradas por A (51 *bytes*) e por B (50 *bytes*). Isso permite distinguir no log as respostas para as mensagens da UFRJ e da UMass, pois os pacotes de *ICMP echo reply* mantêm o mesmo tamanho das mensagens de *echo request* originais. (Mais uma vez, por uma questão de segurança, os nomes reais das máquinas foram substituídos por nomes fictícios.)

2.1.3 Problemas para estimar o atraso unidirecional

Embora o atraso de ida e volta, o *Jitter* e a diferença dos atrasos unidirecionais entre máquinas fontes para uma mesma máquina alvo sejam medidas úteis para algumas aplicações, a medida de desempenho atraso unidirecional encontra também um número grande de aplicações. Por outro lado, essa medida é bem mais difícil de ser estimada. A não ser que dispositivos específicos para sincronização de relógios como *GPS(Global Positioning System)* sejam utilizados pelas máquinas envolvidas, medir o atraso entre duas máquinas na Internet não é trivial. O problema torna-se ainda mais complexo quando não se tem acesso a todas as máquinas da medição. Ou seja, quando é necessária uma medição não cooperativa.

Os problemas para estimar o atraso em um sentido de pacotes, quando não é garantida a sincronia dos relógios das máquinas envolvidas na medição, já vêm sendo discutidos há algum tempo na literatura. O cálculo do atraso em um sentido requer um tratamento especial às diferenças existentes entre os relógios dessas máquinas e algumas soluções já foram propostas [63, 64, 65, 66, 67, 68, 69, 70, 71]. No entanto, todas as técnicas existentes, até então, na literatura, que estimam esta métrica, necessitam de permissão para execução do processo coletor na máquina remota, onde são computadas as informações referentes às chegadas das sondas. O único trabalho existente na literatura, em que é proposta uma técnica não cooperativa para estimar o atraso unidirecional, foi apresentado em [72] com uma versão estendida em [73]. Essa técnica faz parte das contribuições principais desta tese e está detalhada no Capítulo 3. Abaixo são descritos os problemas gerais para estimar o atraso em um sentido quando se tem acesso às máquinas envolvidas na medição.

A Figura 2.7 mostra o resultado de medições feitas entre duas máquinas (uma localizada na UFRJ e outra na UMass), em que N sondas foram enviadas nos dois

sentidos. Em cada sentido, uma sequência $\Omega := [v_i = (i, d_i) : i = 1, \ldots, N]$ foi gerada a partir das sondas coletadas no destino, onde i equivale ao número de sequência da i-ésima sonda enviada e d_i ao atraso obtido pela simples diferença entre os tempos de envio e recebimento da sonda i.

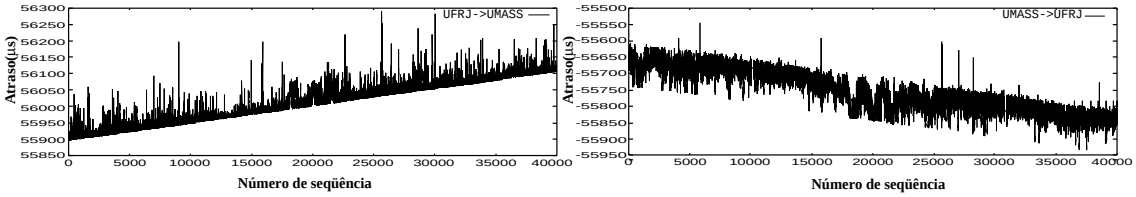


Figura 2.7: Atraso de pacotes entre máquinas com relógios não sincronizados.

O primeiro problema, chamado de *Skew*, é resultante da diferença na taxa de crescimento dos relógios das máquinas. Considerando que os relógios não são atômicos, a taxa do relógio em uma máquina pode ser maior ou menor do que na outra. Em consequência, o resultado do cálculo do atraso entre duas máquinas sofre um crescimento ou decrescimento constante. Quando o experimento é executado por um tempo maior que poucos segundos, o erro causado pela diferença nas taxas de crescimento dos relógios é significativo e causa um crescimento ou decrescimento na sequência de atrasos computados das sondas, como mostrado na Figura 2.7.

O segundo problema, chamado de *Offset*, surge em consequência dos relógios das máquinas envolvidas na medição possuírem valores distintos no início da medição. O valor dessa diferença é somado ou diminuído do valor real do atraso, resultando até mesmo em valores negativos para as estimativas d_i.

Algoritmos para remoção do *Skew* e *Offset*

Soluções foram propostas para remover das coletas os valores causados pelos problemas de *Skew* [64, 65, 68] e *Offset* [63, 64, 66, 67].

Todos os algoritmos, existentes para remoção do *Skew* [64, 65, 68], têm como objetivo estimar uma função linear, que esteja abaixo e mais próxima possível de todos os pontos em Ω, para representar a tendência de crescimento ou decrescimento em uma coleta. A diferença entre os métodos está basicamente na definição da função objetivo definida em cada uma das propostas. Um exemplo de função objetivo, definida em [65], é dado por: minimizar a soma das distâncias verticais entre os

vértices v_i e a reta da função linear.

Em [65], Moon, Skelly e Towsley propõem o uso de um algoritmo de programação linear para estimar a função linear. Além de proporem o novo método, fazem uma comparação entre esse e o proposto por Paxson [64]. Na avaliação dos algoritmos, é demonstrado um fraco desempenho no quesito robustez por parte da proposta de Paxson, sendo verificado que, em caso de altos valores do *Skew*, o algoritmo falha na estimativa desse parâmetro. Uma avaliação dos algoritmos e uma nova proposta é também apresentada por Zhang et al. em [68]. Os autores provam que sua proposta possui uma menor complexidade computacional do que a feita por Paxson, e menor ou igual do que a proposta de Moon, Skelly e Towsley.

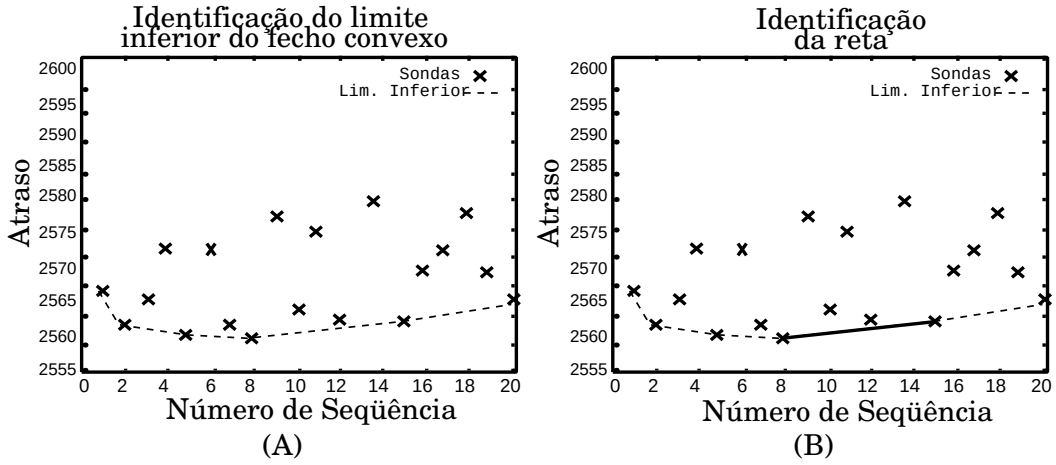


Figura 2.8: Funcionamento dos algoritmos para remoção do Skew.

A proposta de Zhang et al., exposta em [68], é baseada na estimativa do fecho convexo da sequência coletada Ω. O fecho convexo de um conjunto de pontos em duas dimensões consiste no menor polígono convexo formado por um subconjunto desses pontos, onde todos os outros pontos deste conjunto se encontram na parte interior do polígono. Os pontos pertencentes a esse subconjunto equivalem aos vértices do polígono. O limite inferior (superior) de um fecho é formado pelos vértices inferiores (superiores) do polígono entre o ponto de menor valor na dimensão "x" até o ponto de maior valor na dimensão "x".

No primeiro passo do algoritmo apresentado em [68], é determinado o limite inferior do fecho convexo de Ω, conforme ilustrado na Figura 2.8(A). A reta que cobre exatamente o ponto médio da coleta é a solução para o seguinte problema de

otimização: minimizar a área entre a curva formada pelos vértices v_i e uma função linear qualquer. Por exemplo, se N sondas forem geradas a intervalos de tempo determinísticos, o ponto médio é igual a $N/2$. A Figura 2.8(B) ilustra a reta obtida para a coleta Ω. Uma vez obtida a solução do problema de otimização, estimar a inclinação da função linear é trivial. Seja $y = f(x)$ a reta estimada e v_i e v_j dois pontos desta reta onde v_i é o vértice inicial. A inclinação desta reta em relação ao eixo das abscissas é dada por $\alpha = (d_j - d_i)/(j - i)$ e representa a diferença entre as taxas de crescimento dos relógios envolvidos na medição. O valor do atraso sem *Skew* pode ser calculado por: $atraso\_sem\_Skew_i = d_i - ((g_i - g_1) * \alpha)$, onde g_i e g_1 são os instantes de geração da sonda i e da primeira sonda, respectivamente.

Uma nova sequência γ é, então, gerada após o cálculo do atraso sem *Skew* para todas as N sondas recebidas. É importante perceber que os valores do d_i, computados nessa nova sequência, equivalem ao valor real do atraso somado (ou diminuído) do *Offset* inicial da coleta. Isso porque, os relógios não se encontravam sincronizados no início da medição. Para estimar o valor real do atraso unidirecional é necessário estimar e remover da coleta o valor referente ao *Offset*.

Algumas soluções para estimar o *Offset* entre duas máquinas estão definidas na literatura [63, 64, 66, 67]. No entanto, apenas a proposta apresentada em [67] considera a possibilidade de capacidades de transmissão assimétricas nos dois sentidos. Isto é, as capacidades de transmissão dos enlaces ao longo do caminho de ida podem ser diferentes das capacidades no caminho de volta.

Para estimar o *Offset* entre duas máquinas, o algoritmo de [67] requer o envio de sequências de sondas, de diferentes tamanhos, simultaneamente nos dois sentidos (por exemplo, uma sequência de sondas da máquina A para a máquina B e uma sequência de B para A). O método pressupõe que a distância percorrida pelas sondas, enviadas em cada um dos sentidos, são aproximadamente as mesmas; assim, a diferença entre os tempos de propagação de A para B e de B para A é desprezível ($T_{AB}^{prop} - T_{BA}^{prop} \approx 0$). Das sequências coletadas em cada um dos sentidos, são selecionadas as sondas que obtiveram o menor atraso, para cada tamanho usado na geração. Essas amostras de atraso selecionadas equivalem às sondas que supostamente não entraram em fila durante todo o caminho percorrido ($T^{fila} = 0$). Neste caso, o atraso de uma sonda selecionada é igual ao tempo de propagação no caminho

somado ao seu tempo de transmissão ($d_{AB} = T_{AB}^{prop} + T_{AB}^{tx}$ e $d_{BA} = T_{BA}^{prop} + T_{BA}^{tx}$).

A partir dos valores dos atrasos das sondas selecionadas de diferentes tamanhos, para cada um dos sentidos, são obtidas duas retas, como mostra a Figura 2.9. Considerando que o atraso das sondas obedece uma função linear (em relação ao tempo de transmissão), estima-se o atraso sofrido por uma sonda supostamente de tamanho nulo, caso tal sonda pudesse ser enviada. Pela Figura 2.9 é fácil verificar que, como o atraso varia linearmente com o tamanho da sonda transmitida, o ponto de interseção entre o eixo das ordenadas e a reta obtida usando os menores valores de atraso, para aquele sentido, é uma estimativa do atraso sofrido por uma sonda de tamanho nulo.

Sejam d_{AB}^{nulo} e d_{BA}^{nulo} os atrasos de uma sonda de tamanho nulo enviada da máquina A para a B e da máquina B para a A, respectivamente. Então, $d_{AB}^{nulo} = O + T_{AB}^{prop}$, e $d_{BA}^{nulo} = -O + T_{BA}^{prop}$ onde, T_{AB}^{prop} e T_{BA}^{prop} são os tempos de propagação entre A e B e entre B e A, respectivamente (supostamente igual nos dois sentidos), e O é o valor do *Offset*. Logo, $d_{AB}^{nulo} - d_{BA}^{nulo} = 2O$. Portanto, o *Offset* é obtido subtraindo os valores d_{AB}^{nulo} e d_{BA}^{nulo} estimados, e dividindo o resultado por dois.

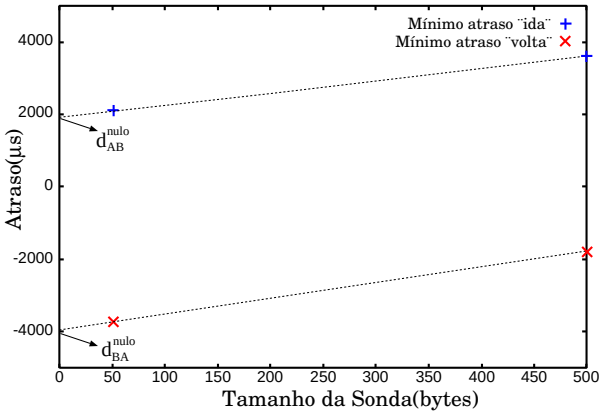


Figura 2.9: Atraso das sondas de tamanhos variados.

Um *framework* para estimar o atraso unidirecional

Em [71] foi definido um *framework* para estimar o atraso em um sentido. As técnicas propostas em [68] para remoção do *Skew* e de [67] para remoção do *Offset* foram implementadas no módulo de medição ativa da ferramenta *TANGRAM-II* [74, 75, 76, 77]. Do nosso conhecimento, a ferramenta *TANGRAM-II* é a única que permite

a estimativa do atraso em um sentido sem que as máquinas envolvidas na medição estejam com seus relógios sincronizados.

A ferramenta exige acesso à máquina alvo e gera tráfego seguindo os padrões definidos pelos algoritmos: sondas são enviadas a intervalos determinísticos, nas duas direções, e de tamanhos variados. As sondas são coletadas no destino e, após o término da coleta, algoritmos são executados para remoção de *Skew* e remoção de *Offset*. As Figuras 2.10(A) e (B) ilustram os atrasos unidirecionais computados para uma sequência de sondas coletadas após a execução dos algoritmos para remoção do *Skew* e do *Offset*, respectivamente. Em [71] é também apresentada uma série de resultados experimentais realizados com a ferramenta *TANGRAM-II* para caracterizar a distribuição do atraso unidirecional computado entre máquinas localizadas nos laboratórios LAND(COPPE/UFRJ), CNRG(UMass-Amherst) e NUPERC(UNIFACS).

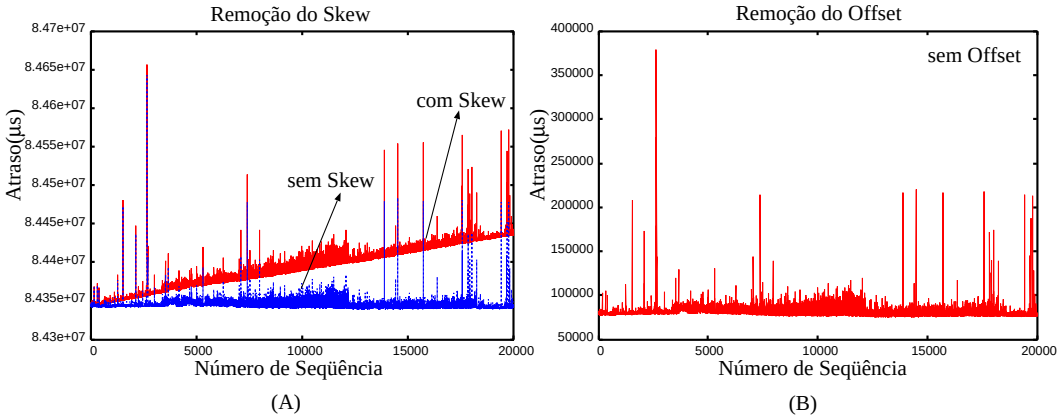


Figura 2.10: Atraso estimado por uma medição da ferramenta *TANGRAM-II*.

2.1.4 Medições fim-a-fim para estimar capacidade

Capacidade de contenção (ou capacidade do gargalo), capacidade de transmissão dos enlaces de um caminho e largura de banda disponível são algumas das medidas associadas à capacidade de transmissão em redes de computadores. Diversos métodos foram propostos para estimar essas e outras métricas relacionadas. Dentre os métodos mais conhecidos estão: (i) *One-packet*, implementado pelas ferramentas Pathchar[78] e Clink[79], que tem como objetivo estimar a taxa de transmissão de todos os enlaces presentes no caminho de rede medido [80]; (ii) *Mult-packet*, uma

variação da técnica *One-packet* desenvolvida por Lai e Baker em [81], que também tem como finalidade estimar a capacidade de transmissão dos enlaces de um caminho; (iii) *Pares de pacotes* (ou *Packet-pairs*), que é amplamente utilizado na literatura para estimar a capacidade de contenção e outras métricas relacionadas; e, (iv) *Trem de pacotes* (ou *packet-train*), que é uma extensão da técnica de *Pares de pacotes*, desenvolvida por Dovrolis et al. em [82], e é utilizada por ferramentas como Pathrate[83] e Pathload[84] para medir, respectivamente, a capacidade de contenção e a largura de banda disponível em um caminho de rede.

Descrições mais detalhadas sobre o funcionamento de cada um desses métodos podem ser encontrados em diversos trabalhos da literatura [43, 44]. O CAIDA[4] mantém uma página *web* com descrições e ponteiros para algumas ferramentas de medições de capacidade [85] disponíveis na Internet. O foco a seguir será apenas para o método de pares de pacotes e suas variações, pois são os mais relacionados às contribuições apresentadas nesta tese.

Medições de capacidade com pares de pacotes

O método de pares de pacotes consiste na emissão de dois pacotes de mesmo tamanho e de uma mesma origem, separados por um intervalo de tempo bem próximo de zero. Os pacotes atravessam o mesmo caminho na rede até chegarem a um único destino, onde são coletados. A partir da coleta destes pacotes é possível identificar algumas características do caminho de rede atravessado pelo par, como a capacidade de contenção.

A suposição principal da técnica é que a dispersão entre os pacotes do par, identificada na coleta, é causada pela menor capacidade de transmissão ao longo do caminho. Os pacotes, que são gerados de uma mesma origem e separados por intervalos de tempo bem próximos de zero, possuem o espaçamento entre eles mantido até que passem por um enlace com capacidade de transmissão inferior à do emissor. Essa dispersão, causada pelo tempo de transmissão deste enlace (superior aos tempos experimentados nos enlaces anteriores) é mantida até o destino dos pacotes, a menos que seja encontrado, ao longo do restante do caminho, um outro enlace com

[4]Cooperative association for Internet data analysis (CAIDA) é um programa de cooperação para medições de desempenho e análise de dados na Internet.

uma capacidade ainda menor. A Figura 2.11 ilustra a causa da dispersão entre os pacotes em sua recepção.

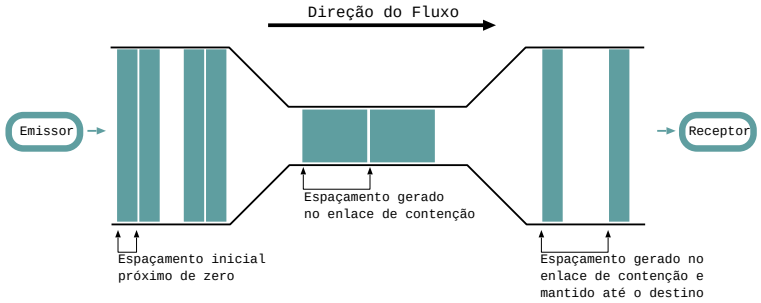


Figura 2.11: Ilustração do funcionamento do método Pares de Pacotes com a dispersão imposta pelo enlace de menor capacidade.

Com o valor do intervalo de tempo entre as chegadas e o tamanho dos pacotes, é possível estimar a capacidade de contenção. Seja T o intervalo de tempo entre as chegadas dos dois pacotes dado em segundos, e seja B o tamanho dos pacotes dado em *bits*. A capacidade de contenção, representada em *bits* por segundo, pode ser obtida a partir da divisão do tamanho do pacote pelo intervalo de tempo entre as chegadas: $C = \frac{B\ bits}{T\ segundos}$.

Estimar a capacidade do enlace de contenção, com base na dispersão entre as chegadas dos pares de pacotes, foi originalmente ilustrado em [86], mas no trabalho apresentado por Jacobson não foi considerada a existência de tráfego concorrente. Resultados das avaliações feitas do método de pares de pacotes, como os apresentados em [82, 87, 88, 89, 90, 91], demonstram que o estado da rede durante a medição é fator crucial para a precisão da estimativa. As condições atuais da rede, como de alto tráfego concorrente, podem influenciar negativamente as medições de tal forma que resultados errados sejam estimados.

A influência causada pelo tráfego concorrente pode ser caracterizada de duas formas: (i) a presença de pacotes em frente aos pares na fila dos roteadores, após já terem passado pelo nó de contenção do caminho, pode ocasionar uma redução na dispersão existente entre os pacotes. Como conseqüência, a capacidade de contenção é superestimada; (ii) a inserção de tráfego concorrente entre os dois pacotes do par. Este evento pode resultar em um acréscimo da dispersão dos pacotes e causar uma estimativa inferior à capacidade real de transmissão do enlace de contenção. Para melhorar a precisão da estimativa, pode ser utilizada uma série de pares e gerado

um histograma das capacidades estimadas por todos os pares. A capacidade de contenção estimada para o experimento equivale àquela que apresenta o maior valor de probabilidade no histograma obtido.

Keshav, em [92], foi o primeiro a usar o método para medir a capacidade de contenção, levando em consideração a existência de um tráfego concorrente. Bolot também utilizou os pares de pacotes para medir a capacidade de um canal inter-continental em [93]. Seguiram-se diversas propostas e ferramentas que utilizaram o método, ou variações dele, para estimar a mesma métrica ou outras medidas baseadas no envio de pares de pacotes.

Em [91], Rocha et al. apresentam uma variação da técnica de pares de pacotes, em que apenas os pares selecionados são utilizados para computar a capacidade de contenção. A seleção dos pares é feita baseada no atraso unidirecional sofrido pelo primeiro pacote do par. O objetivo desta seleção é usar apenas pares que, supostamente, sofreram pouca ou nenhuma influência do tráfego concorrente durante a travessia do caminho na rede.

A ferramenta CapProbe, apresentada em [94], também propõe uma seleção dos pares de pacotes utilizados para computar a capacidade de contenção baseada no atraso sofrido pelas sondas. Uma diferença desta técnica para a apresentada por Rocha et al. em [91] é que a primeira refere-se à métrica do caminho de ida e volta, enquanto que a outra mede a capacidade de contenção unidirecional. Os parâmetros utilizados para a seleção dos pares de pacotes também são diferentes. A seleção feita pelo CapProbe tem como parâmetro a soma dos atrasos sofridos pelo primeiro e pelo segundo pacote do par, enquanto que no *TANGRAM-II* a seleção é feita baseada apenas no atraso da primeira sonda do par.

Medições de capacidade em redes 802.11

A partir das considerações mencionadas acima, viu-se que a dispersão dos pares de pacotes em uma rede cabeada é causada pela variação da capacidade de transmissão dos enlaces. No entanto, em um caminho de rede, onde exista enlaces sem fio (por exemplo, se o último salto tratar-se de uma *WLAN*), essa dispersão pode ser consequência não só da taxa de transmissão da camada física, mas também do *overhead* do padrão 802.11. Portanto, a equação $C = \frac{B\ bits}{T\ segundos}$ não pode ser usada

para estimar a taxa de transmissão do enlace sem fio. No Capítulo 4 desta tese serão discutidos detalhes das características inerentes aos padrões do protocolo 802.11 e os desafios para o uso de pares de pacotes em redes 802.11.

A medida de desempenho obtida, através de ferramentas como Pathrate, CapProbe e *TANGRAM-II*, quando aplicadas a um caminho de rede que apresente salto(s) sem fio, depende do cenário existente. Se o enlace de menor capacidade em todo o caminho não tratar-se do salto sem fio e a dispersão dos pacotes do par for ocasionada por um enlace cabeado, então a medida obtida é mesmo uma estimativa da capacidade de contenção. No entanto, se o enlace de menor capacidade do caminho está no salto sem fio, então a medida obtida é a taxa (ou capacidade) de transmissão *efetiva* do enlace 802.11. Note que devido às características do protocolo 802.11, a medida obtida não é a taxa de transmissão desse dispositivo, mas sim a capacidade de transmissão *efetiva* do enlace sem fio 802.11. Se a medição for executada na ausência de tráfego concorrente, essa taxa de transmissão efetiva equivale à vazão máxima alcançada por um fluxo neste salto. Caso a medição seja feita com a existência de tráfego concorrente, a medida não necessariamente será igual à vazão máxima.

O primeiro trabalho a considerar características do protocolo 802.11 para medições de capacidade com pares de pacotes foi apresentado em [35] e uma versão extendida em [36]. Esses trabalhos descrevem e avaliam uma técnica proposta para estimar a taxa de transmissão de enlaces em uma rede local sem fio e faz parte das contribuições desta tese, apresentadas no Capítulo 4.

Em trabalhos anteriores já foram utilizadas técnicas de medições fim-a-fim para estimar algumas métricas relacionadas à capacidade em caminhos onde o último salto é uma rede 802.11 [94, 95]. Em [94], por exemplo, medições foram executadas em um caminho onde o enlace de menor capacidade estava no último salto e este era uma *WLAN*. No trabalho foi utilizada a ferramenta CapProbe e, portanto, foi medida a capacidade efetiva do enlace sem fio.

No trabalho apresentado em [95] é proposta uma ferramenta, chamada ProbeGap, que tem como objetivo estimar a largura de banda disponível na rede de acesso do último salto. Naquele trabalho, medições são feitas em ambientes de acesso por *Cable Modem* ou *WLAN*. O trabalho apresentou também resultados obtidos com

a ferramenta PathRate para estimar a capacidade efetiva de enlaces sem fio, em diversos cenários, variando a taxa de transmissão e o tráfego concorrente. Os resultados obtidos com a ferramenta PathRate serviram para auxiliar na avaliação dos resultados obtidos com a ferramenta proposta (ProbeGap) para estimar a largura de banda disponível.

2.2 Avaliação de desempenho de aplicações P2P para distribuição de conteúdo na Internet

Na seção anterior foram descritas medidas de desempenho úteis para diversas aplicações. Esta seção, agora, é dedicada a uma aplicação específica (a aplicação peer-to-peer) e a utilidade de medição para estudar características importantes desses sistemas.

Peer-to-peer é um modelo de arquitetura de sistemas distribuídos, que tem como característica fundamental a descentralização das funções, onde cada entidade do sistema opera como cliente e servidor ao mesmo tempo. Embora a computação peer-to-peer seja aplicável a inúmeros sistemas, certamente as aplicações para distribuição de conteúdo são as mais populares. O BitTorrent[9], por exemplo, é uma das aplicações para disseminação de conteúdo mais bem sucedidas da Internet. Estudo recente, apresentado em [8], sugere que o tráfego gerado por clientes BitTorrent já representa mais de um terço de todo tráfego passante nas redes de diversos provedores na Internet. Parte desse sucesso se deve à alta escalabilidade e robustez inerente à arquitetura P2P, que permite aos usuários distribuir conteúdo para milhares de outros usuários de maneira eficiente.

Entender as vantagens do modelo de distribuição de conteúdo, através de aplicações P2P, em comparação ao modelo tradicional cliente/servidor, é o objetivo da próxima subseção (2.2.1). Alguns trabalhos da literatura dedicados à análise de disponibilidade e custo para disseminação de conteúdo, através de aplicações P2P, são discutidos em seguida (subseções 2.2.2 e 2.2.3).

2.2.1 Aplicações P2P vs. Cliente/servidor

Para compreender as vantagens do uso de uma arquitetura P2P em relação à arquitetura cliente/servidor para distribuição de um conteúdo, considere um modelo simples para representar o cenário em que um provedor de conteúdo dissemina para N clientes (ou *peers*) um arquivo de tamanho igual a F bytes. Sejam u_s e u_c as capacidades de *upload* (em bytes por segundo) atribuídas, respectivamente, ao servidor original do conteúdo e aos clientes interessados no arquivo. Inicialmente, assuma que $u_s \geq u_c$. Por fim, suponha que a capacidade de *download* dos clientes (d_c) seja grande o suficiente para que os clientes estejam sempre fazendo *download* de dados, desde que haja capacidade de *upload* disponível no sistema (por exemplo, $d_c = \infty$ ou que, pelo menos, $d_c \gg u_s$). Assim, o tempo de *download* do conteúdo pelo cliente, nesta análise, estará limitado apenas pela capacidade de *upload* dos dados na rede. Outra análise, relaxando essa suposição, será discutida mais adiante. (Note que todos os clientes têm as mesmas capacidades de *upload* e *download*: $u_i = u_c$ e $d_i = d_c$, para $i = 1, \ldots, N$.)

Na arquitetura cliente/servidor, uma cópia do arquivo com F bytes deve ser transmitida para um dos N clientes do sistema. Tarefa essa que deve ser realizada, exclusivamente, pelo servidor. Já na arquitetura P2P, os clientes (peers) auxiliam ao servidor na disseminação do conteúdo. Esses peers, ao receberem uma parte do arquivo enviada pelo servidor (ou por um outro peer), passam a auxiliar na disseminação do conteúdo, operando como servidor daquele pedaço do arquivo, para outros peers da rede. A partir desse modelo simplificado, é possível estimar o tempo necessário para que o arquivo seja distribuído, por completo, a todos os clientes do sistema, na arquitetura cliente/servidor (equação 2.2) e na arquitetura P2P (equação 2.3).

$$D_{cs} = \frac{NF}{u_s} \tag{2.2}$$

$$D_{p2p} = \frac{NF}{u_s + \sum_{i=1}^{N} u_i} \tag{2.3}$$

Pelas equações 2.2 e 2.3, nota-se que o tempo para distribuição do conteúdo na arquitetura P2P será sempre menor ou igual ao tempo de distribuição na arquitetura

cliente/servidor. Quando existir apenas um cliente no sistema, o tempo para disseminação do conteúdo será o mesmo nas duas arquiteturas. No entanto, à medida que o número de clientes cresce ($N \to \infty$), a diferença entre D_{cs} e D_{p2p} tende a aumentar. Isso porque, na arquitetura cliente/servidor, cada cliente adicional traz ao sistema apenas um acréscimo de serviço ao único distribuidor existente no sistema; enquanto que, na arquitetura P2P, novos clientes agregam também capacidade ao sistema.

Uma generalização desse modelo foi apresentada por Kumar e Ross, em [96]. No trabalho, os autores relaxam algumas das suposições apresentadas acima (primeiro parágrafo desta subseção) e chegam a um modelo mais geral, que permite computar o limite inferior do tempo de distribuição do arquivo nas duas arquiteturas. Diferente do modelo anterior, o proposto por Kumar e Ross prevê a possibilidade de capacidades de *download* distintas entre os clientes. O modelo também não assume que as capacidades de *download* sejam, necessariamente, muito grandes ou muito maiores que u_s, além de não restringir que a capacidade de *upload* do servidor (u_s) seja maior ou igual às capacidades de *upload* dos clientes u_c.

Os limites do tempo de distribuição do arquivo nas duas arquiteturas são dados pelas equações 2.4 e 2.5, conforme comentado em [96]. Na arquitetura cliente/servidor (equação 2.4), o tempo de distribuição será maior ou igual ao máximo, dentre os seguintes fatores: (i) NF/u_s, que representa o tempo máximo para que o servidor faça *upload* das N cópias do arquivo para os clientes, desde que sempre existam clientes com capacidade de *download* disponível; (ii) F/d_{min}, que é o tempo necessário para o cliente, com a menor capacidade de *download* (representado por d_{min}), recuperar um arquivo de tamanho F, desde que haja capacidade de *upload* disponível. Na arquitetura P2P, o tempo para disseminar todo o conteúdo é maior ou igual ao máximo entre esses três fatores: (i) $NF/(u_s + \sum_{i=1}^{N} u_i)$, que é o tempo necessário para disseminar as N cópias do arquivo para os clientes, se sempre houver clientes com capacidade de *download* disponível; (ii) F/d_{min}, que representa o tempo para que o cliente com a menor capacidade faça o *download* do arquivo; (iii) F/u_s, tempo requerido para que um conteúdo de tamanho F seja transmitido pelo

servidor.

$$D_{cs} \geq MAX \left[\frac{NF}{u_s}, \frac{F}{d_{min}} \right] \tag{2.4}$$

$$D_{p2p} \geq MAX \left[\frac{NF}{u_s + \sum_{i=1}^{N} u_i}, \frac{F}{d_{min}}, \frac{F}{u_s} \right] \tag{2.5}$$

No Capítulo 5 será introduzido o conceito de redes de sistemas P2P (*swarms*) auto-sustentáveis. Na ocasião será mostrado que, para alguns casos particulares de *swarms* auto-sustentáveis, esse limite definido pela equação 2.5, para o tempo de disseminação do conteúdo em arquitetura P2P, não é válido.

Dentre os inúmeros trabalhos dedicados a analisar o desempenho de sistemas P2P e compará-la em relação à arquitetura cliente/servidor, um dos primeiros foi apresentado em [97]. Naquele trabalho, Qiu e Srikant apresentam um modelo de fluido para capturar a interação de peers em um *swarm*. O modelo captura a essência do sistema, para o caso em que um número muito grande de usuários participam do swarm, e calcula o tempo médio de *download* do arquivo. Através do modelo, é possível compreender melhor características fundamentais do sistema P2P analisado em questão (no caso, o BitTorrent), como os mecanismos de incentivo *tit-for-tat* e de distribuição *rarest-first* desse sistema. (Detalhes sobre o funcionamento do protocolo BitTorrent e de seus mecanismos serão apresentados no Capítulo 5 desta tese)

2.2.2 Análise de disponibilidade de conteúdo em aplicações P2P

Nas aplicações P2P, um arquivo é considerado disponível quando 100% do conteúdo encontra-se disponível para *download* por outras máquinas da rede. Esse conteúdo pode estar disponível, por completo, em uma única máquina ou, em partes complementares, localizadas em diferentes *peers* da rede. Caso qualquer parte do arquivo não esteja acessível pelos clientes de uma rede P2P, esse conteúdo passa a ser considerado indisponível.

O problema da disponibilidade de conteúdo é inerente a todos os sistemas P2P. Conteúdos muito populares, em geral, são amplamente difundidos nas redes P2P. Já os arquivos que não são de interesse dos usuários, ou que perderam popularidade com

o passar do tempo, tendem a possuir uma baixa disponibilidade no sistema. Quando comparado às demais aplicações P2P, no BitTorrent a questão da disponibilidade torna-se ainda mais crítica, uma vez que nesse sistema falta incentivo aos usuários para manterem o conteúdo disponível, após concluírem o download.

Os mecanismos de incentivo, existentes nos atuais sistemas P2P, podem ser: (i) baseados em cooperação a longo prazo (a exemplo da rede eDonkey2000). Neste caso, um usuário que coopera com o sistema em um determinado *swarm* acumula "fichas" que podem ser utilizadas em benefício próprio em outro *swarm* da mesma rede; ou, (ii) baseado em reciprocidade instantânea (esquema adotado pelo BitTorrent), em que o crédito acumulado pela cooperação em um *swarm* só pode ser utilizado naquele mesmo *swarm*. Pode-se dizer que as duas soluções apresentam vantagens e desvantagens. A solução (i) tem como principal desvantagem a dificuldade de se implementar sistemas econômicos distribuídos, sem a existência de uma entidade central (e.g., um "banco") para regular a quantidade de "dinheiro". Sem a existência de uma entidade reguladora, torna-se possível que usuários burlem o sistema, acumulem créditos falsos e usem as fichas para levar vantagem sobre os demais usuários. Já a solução (ii), os sistemas baseados em reciprocidade direta estão intrinsecamente limitados pela ausência de crédito global. Não existe acúmulo de crédito para ser usado no futuro, o que implica que todas as trocas são feitas usando barganha. Assim, não há incentivos para que os usuários, após concluírem o *download*, permaneçam por mais tempo, para cooperar com o sistema compartilhando os arquivos.

No grupo de estudo de aplicações P2P da CNRG/UMass-Amherst, foi desenvolvida uma arquitetura para monitoramento em larga escala da rede BitTorrent. Essa infra-estrutura encontra-se em atividade desde agosto de 2008, coletando informações sobre todos os usuários conectados aos *swarms* anunciados pelo Mininova[5]. Os monitores, definidos na arquitetura, conectam-se à rede e coletam diversas informações dos demais clientes conectados ao *swarm*, dentre elas o percentual de *download* concluído do arquivo. O resultado mostrado na Figura 2.12 foi obtido de coletas feitas, pelo grupo da UMass, entre os meses de agosto de 2008 e março

[5]Mininova.org é um site de busca e divulgação dos *swarms* da rede BitTorrent. Recentemente este site foi parcialmente desativado e atualmente limita-se a divulgar apenas *swarms* de conteúdo legal.

de 2009, onde, na ocasião, mais de 66 mil *swarms* estavam sendo monitorados.

A Figura 2.12 ilustra a função distribuição cumulativa (CDF) da fração de tempo em que o conteúdo esteve disponível, nos *swamrs* monitorados. A linha sólida mostra a disponibilidade considerando apenas os 30 primeiros dias de existência do *swarm*, período em que se espera que o conteúdo seja mais popular. Essa curva mostra que menos de 35% dos *swarms* tiveram o conteúdo disponível, ao longo de todo o seu primeiro mês de vida. Quando é considerado todo o período de medição, a indisponibilidade nos *swarms* é ainda maior. A linha tracejada mostra que, aproximadamente, 75% dos *swarms* se mantiveram disponíveis por no máximo 20% do tempo, durante os meses de monitoramento.

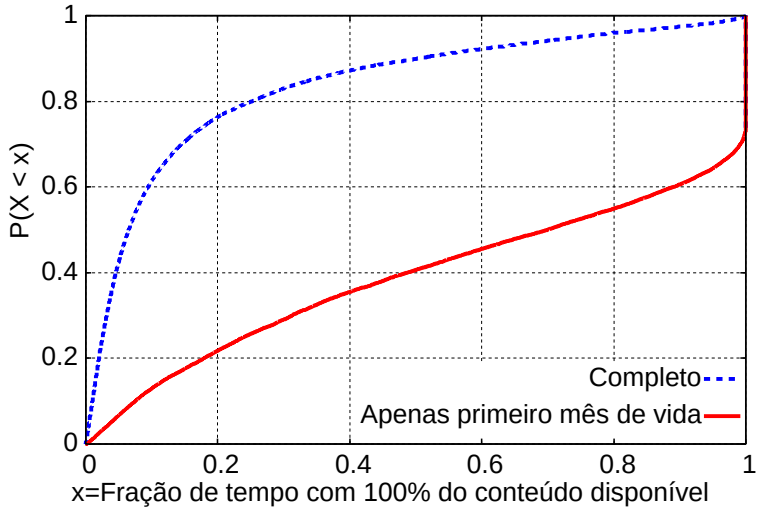


Figura 2.12: CDF dos arquivos disponíveis.

Quando o conteúdo (ou parte dele) não está disponível em um *swarm*, os usuários que desejam recuperar esse arquivo ficam bloqueados, a espera de que algum peer que possua esses dados retorne à rede. Ramash et al. foram os primeiros a alertar para a questão que eles chamaram, em [98], de Problema de Leechers Bloqueados (do inglês *BLP-Blocked Leecher Problem*). Clientes que desejam um arquivo devem esperar indefinidamente para obter certas partes do arquivo que não se encontram mais disponíveis. A solução para esse problema, inerente ao BitTorrent, apresentada em [98], foi o Bitstore: uma arquitetura que reduz o problema de indisponibilidade de conteúdo no BitTorrent, utilizando um sistema de incentivo baseado em fichas.

Resultados de uma grande sessão de monitoramento, apresentados por Guo et al. em [99], demonstram que a popularidade de um *swarm* (definido como a taxa

de chegada de novos peers) decai exponencialmente ao longo do tempo. Assim, usuários que cheguem tarde ao sistema "perdem o melhor da festa" e podem não mais encontrar o conteúdo desejado disponível. Pouwelse et al. foram pioneiros em estudos de medições em larga escala para o BitTorrent. Dentre outras conclusões, os resultados apresentados em [100] comprovam, por exemplo, que existe uma grande correlação entre a popularidade e a disponibilidade dos arquivos no BitTorrent.

A questão da disponibilidade de conteúdo também foi analisada para outros sistemas P2P [101, 102, 103]. O curto tempo de monitoramento adotado em alguns desses trabalhos (poucas semanas em [101] e alguns dias em [102]) limitam as conclusões dos estudos. No entanto, as conclusões dos dois trabalhos apontam problemas de indisponibilidade também nas redes Napster, Gnutella e Overnet [101, 102]. Resultados de experimentos de maior duração foram apresentados em [103], onde foram analisados dados de mais de 200 dias do tráfego coletado na rede da Universidade de Washington, referentes à aplicação Kazaa. Uma das conclusões do trabalho sugere que usuários peer-to-peer são mesquinhos. Isto é, a maioria dos usuários consomem dados, mas provêem pouco em contrapartida.

O trabalho apresentado por Neglia et al. [104] também analisa a disponibilidade de conteúdo em sistemas P2P. O estudo, desenvolvido através de um largo experimento utilizando o protocolo BitTorrent, analisa o impacto na disponibilidade do conteúdo, quando há falhas na disseminação de informações de controle sobre o *swarm*. A maior parte do controle do *swarm* é feita por entidades denominadas *trackers* e os resultados apresentados em [104] demonstram que eventuais falhas dessas entidades ocasionam impactos significativos no desempenho experimentado pelos usuários do *swarm*.

O desenvolvimento de novos mecanismos para sistemas P2P, cujo objetivo seja aumentar a disponibilidade do conteúdo, tem sido tema de pesquisa na literatura. Gkantsidis e Rodriguez, em [105], propõem o uso de *network coding* no protocolo utilizado pelo BitTorrent para distribuição de conteúdo em larga escala utilizando BitTorrent. A idéia é explorar a aleatoriedade introduzida pelo processo de codificação para auxiliar na programação da transmissão de bloco e, como isso, tornar a distribuição mais eficiente. Através de simulações, os autores demonstram que a adoção de *network coding*, no mecanismo de disseminação do BitTorrent, pode

representar melhorias significativas na disponibilidade e desempenho da aplicação. O trabalho apresentado em [105] prevê a alteração do protocolo BitTorrent. Outros trabalhos propõem soluções para o problema de disponibilidade no BitTorrent, sem alterações na estrutura do protocolo. Um desses trabalhos é parte das contribuições desta tese e será detalhado no Capítulo 5.

2.2.3 Redução de custo para distribuição de conteúdo em P2P

Em 2007, numa entrevista concedida ao TorrentFreak[106], Bram Cohen, criador do BitTorrent e co-fundador do BitTorrent Inc., destacou como um dos futuros grandes desafios da comunidade o uso, como solução comercial, de protocolos P2P para a otimização da distribuição de conteúdo na Internet. Desde então, a busca por soluções que otimizem o custo (em termos de redução de consumo de banda passante ou mesmo de energia) para a disseminação de conteúdo comercial tem se estabelecido como um tema de pesquisa que desperta o interesse, tanto da comunidade acadêmica quanto das empresas. Os fundadores da Kontiki Inc., desenvolvedora de uma solução comercial para distribuição de conteúdo através de P2P, relatam em [107] os principais desafios deparados no desenvolvimento desse sistema.

A McAfee e a Akamai são exemplos de empresas que também vêm adotando soluções P2P, como relatam os artigos apresentados em [108, 109]. O serviço desenvolvido pela McAfee, VirusScan ASaP, usa técnicas P2P para compartilhamento de atualizações de antivirus. Antes de buscar nos repositórios oficiais da McAfee, estações VirusScan ASaP checam se já existe alguma outra máquina na mesma rede local que contenha esses dados de atualização. Se houver, os dados para atualização do software são recuperados localmente, economizando tráfego no canal de acesso à Internet. Mais recentemente, a *Akamai Technologies* adquiriu uma empresa especializada em soluções para transferência de dados via P2P, com o objetivo de desenvolver e, então, oferecer a seus clientes, serviços de disseminação de conteúdo utilizando esse modelo de arquitetura.

Os benefícios do uso de soluções P2P para distribuição de atualização de software são discutidos em [110]. Naquele trabalho, os autores investigam o sistema de atualização automática do Windows, um dos maiores serviços de atualização de

software existentes na Internet. Resultados, apresentados por Gkantsidis et al.[110], comprovam que a arquitetura P2P trata-se de uma solução de grande potencial para um serviço mais eficiente aos clientes e, ao mesmo tempo, de menor custo de distribuição para os provedores.

Uma solução otimizada para disseminação de conteúdo é o modo de operação Super-seeding[111], implementada por John Hoffman no BitTornado[112], uma aplicação cliente do protocolo BitTorrent. O objetivo desta solução é minimizar o montante total de dados servidos por um cliente BitTorrent, que, eventualmente, seja o único a possuir 100% do conteúdo no *swarm*. O cliente BitTornado, operando no modo Super-seed, alega não possuir qualquer parte do arquivo. À medida que os *peers* se conectam ao *swarm*, o Super-seed informa a um novo *peer* possuir um pedaço do arquivo, que não foi enviado a nenhum outro *peer* da rede, e envia para esse novo *peer* o pedaço do arquivo. O novo *peer*, que acabou de receber um pedaço do arquivo que só ele tem no *swarm*, só volta a receber um outro pedaço de arquivo do Super-seed, quando outros *peers* da rede anunciarem o recebimento daquele pedaço enviado anteriormente. Alterações simples à estratégia de serviço utilizada pelo protocolo BitTorrent também foram propostas e avaliadas em outros trabalhos [113, 114, 115].

Em [116], sistemas que utilizam uma arquitetura P2P para disseminação de conteúdo comercial são chamados de sistemas híbridos P2P, pois o tráfego de um servidor central é reduzido pelo uso da capacidade de seus clientes. Naquele trabalho, Ioannidis e Marbach analisam formalmente esse modelo de sistemas. Através de experimentos de simulação, os autores observam a eficiência das arquiteturas de sistemas híbridos P2P, em que uma grande população pode ser servida, mesmo com um uso limitado de recursos da máquina provedora de conteúdo.

Pesquisa recente considera a seguinte questão: como otimizar a alocação de banda de um servidor entre um conjunto de *swarms* e seus respectivos *peers*, de tal forma a minimizar o tempo de *download* experimentado por esses clientes? Para lidar com essa questão, em [117], os autores propõem o uso do Antfarm: um sistema P2P de distribuição de conteúdo coordenado para múltiplos e concorrentes *swarms*. Para um dado conjunto de *swarms* concorrentes, a entidade central de controle do Antfarm determina a melhor distribuição da banda do servidor entre os *swarms*, de

forma a minimizar o tempo médio de *download* experimentado pelos usuários.

A questão tratada em [117] possui semelhanças com um dos problemas tratados no Capítulo 5 desta tese. No entanto, diferente do objetivo definido em [117], que é minimizar o tempo de *download*, neste trabalho o objetivo é minimizar o custo para a distribuição do conteúdo. Uma outra diferença entre os trabalhos está no fato do sistema Antfarm tratar-se de um protocolo específico P2P, enquanto que a solução apresentada no Capítulo 5 pode ser diretamente adotada ao BitTorrent, sem qualquer alteração ao protocolo do sistema.

Capítulo 3

Soluções não cooperativas para estimar a média e a variância do atraso em um sentido na Internet

ESTE capítulo disserta sobre as contribuições desenvolvidas nesta tese, para a estimativa da média e variância da distribuição do atraso de pacotes em um único sentido, de uma máquina origem A para uma máquina destino D, sem a necessidade de acesso a essa máquina remota D. A descrição da técnica proposta é apresentada na Seção 3.1. Para facilitar a explicação do algoritmo, será considerado, inicialmente, que os relógios das máquinas envolvidas na medição estão perfeitamente sincronizados. Na seção seguinte (3.2) é apresentada a extensão da técnica, quando essa suposição é relaxada. Validações, através de simulações e experimentos reais, são apresentados na Seção 3.3. Por fim, a Seção 3.4 analisa o impacto nos resultados da suposição mais forte definida para a técnica proposta: a de que os tempos de propagação, nos caminhos de ida e volta da rede, são aproximadamente iguais.

3.1 Descrição da técnica proposta

Suponha que sondas são geradas a partir de duas (ou mais) máquinas fonte (i.e. A e B) para uma mesma máquina alvo D. O objetivo é estimar d_{AD} e d_{BD}, isto é, o atraso unidirecional sofrido por cada uma das sondas enviadas pelas máquinas A e

B para a máquina D. Isso sem privilégio de acesso à máquina alvo para execução de processos para coletar as sondas enviadas.

Para lidar com a falta de acesso à máquina remota, foram desenvolvidas duas versões para a técnica proposta. As versões se distinguem quanto ao pré-requisito para a geração das sondas; no entanto, após coletadas as sondas, os algoritmos aplicados são semelhantes. Uma primeira versão requer que o sistema operacional da máquina alvo implemente um contador global para os valores do campo IPID dos pacotes enviados. Como já foi mencionado no Capítulo 2 de trabalhos relacionados, apenas alguns sistemas operacionais implementam um contador global, dentre eles o Microsoft Windows. Quando a máquina alvo não possui um sistema operacional com IPID global, uma segunda versão da técnica pode ser utilizada. Neste caso, é necessário que ao menos uma das máquinas fonte envolvidas na medição seja capaz de transmitir pacotes com *spoofing* do endereço IP.

Para facilitar a compreensão da técnica básica e suas versões, primeiro será descrita a solução desenvolvida para o caso em que a máquina alvo dispõe de um sistema operacional com IPID global. Em seguida, será apresentada a versão da técnica que utiliza *IP spoofing* nos pacotes gerados pelas máquinas fonte.

3.1.1 A técnica utilizando *IPID*

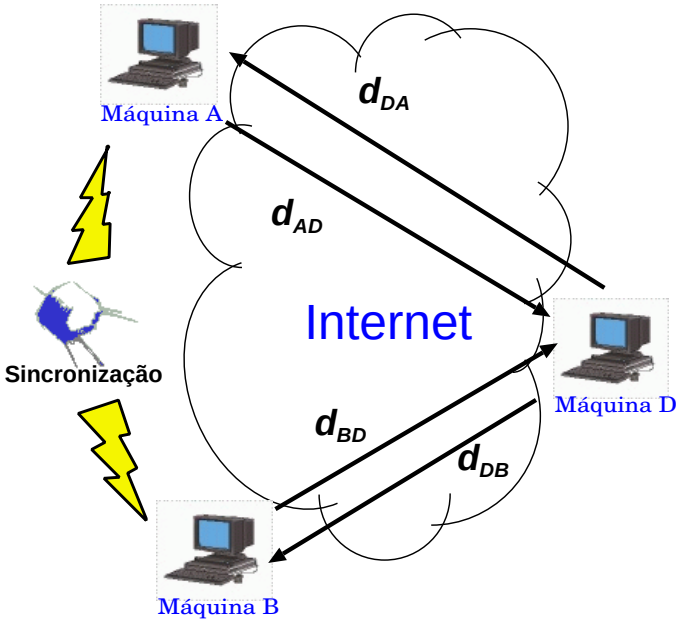


Figura 3.1: Sondas geradas das máquinas A e B para a máquina D.

Considere, por exemplo, o cenário ilustrado pela Figura 3.1, em que as máquinas A e B, com relógios sincronizados, geram sondas para a máquina alvo D. As sondas não são coletadas pela máquina remota e são replicadas de volta às máquinas de origem. Assim como na técnica definida por Chen et al. em [55] (descrita na Seção 2.1.2 desta tese), vamos supor que as sondas enviadas de A e B, que chegam muito próximas umas das outras à máquina alvo, apresentam valores próximos para o IPID, ao serem replicadas por D. Para cada amostra coletada em A e em B, que chegaram juntas em D, é possível montar o seguinte sistema de equações:

$$\begin{cases} d_{AD} + d_{DA} = RTT_{ADA} \quad (i) \\ d_{BD} + d_{DB} = RTT_{BDB} \quad (ii) \\ d_{AD} - d_{BD} = \Psi_{AD-BD} \quad (iii) \\ d_{DA} - d_{DB} = \Psi_{DA-DB} \quad (iv) \end{cases} \tag{3.1}$$

onde, Ψ_{AD-BD} e Ψ_{DA-DB}, obtidos pelo método de Chen et al. [55], representam, respectivamente, a diferença entre os atrasos de A e B para D e de D para A e B; e, RTT_{ADA} e RTT_{BDB} são os atrasos de ida e volta computados para as amostras enviadas de A e de B, respectivamente.

O atraso sofrido por um pacote na rede é formado basicamente pela soma dos tempos de transmissão (T^{tx}), propagação (T^{prop}), processamento (T^{proc}) e filas nos roteadores (T^{fila}). Considerando que o tempo de processamento é desprezível em relação aos demais, então o atraso sofrido por um pacote no caminho entre as máquinas A e D, por exemplo, é igual à soma desses três termos:

$$d_{AD} = T^{tx}_{AD} + T^{prop}_{AD} + T^{fila}_{AD}.$$

Logo, o sistema de equações definido (3.1) pode ser reescrito da seguinte forma:

$$\begin{cases} T^{tx}_{AD} + T^{prop}_{AD} + T^{fila}_{AD} + T^{tx}_{DA} + T^{prop}_{DA} + T^{fila}_{DA} = RTT_{ADA} \quad (i) \\ T^{tx}_{BD} + T^{prop}_{BD} + T^{fila}_{BD} + T^{tx}_{DB} + T^{prop}_{DB} + T^{fila}_{DB} = RTT_{BDB} \quad (ii) \\ T^{tx}_{AD} + T^{prop}_{AD} + T^{fila}_{AD} - (T^{tx}_{BD} + T^{prop}_{BD} + T^{fila}_{BD}) = \Psi_{AD-BD} \quad (iii) \\ T^{tx}_{DA} + T^{prop}_{DA} + T^{fila}_{DA} - (T^{tx}_{DB} + T^{prop}_{DB} + T^{fila}_{DB}) = \Psi_{DA-DB} \quad (iv) \end{cases} \tag{3.2}$$

No sistema de Equações 3.2, apenas os valores dos termos RTT_{ADA}, RTT_{BDB}, Ψ_{AD-BD} Ψ_{DA-DB} são conhecidos. Das quatro equações definidas para o sistema,

apenas três delas são independentes. A dependência linear das equações pode ser facilmente verificada, somando as equações (ii), (iii) e (iv) para obter a equação (i). Além disso, o número de incógnitas existentes nesse sistema (um total de 12 variáveis) é maior do que o número de equações independentes (apenas 3 equações independentes). Logo, o sistema formado pela Equações 3.2 é linearmente dependente, possível e indeterminado, e, portanto, apresenta infinitas soluções.

A técnica definida consiste em restringir o espaço de soluções do sistema de Equações 3.2, inferindo os tempos de transmissão e propagação dos atrasos em cada um dos sentidos. Dessa forma, quando as sondas enviadas por A ou as sondas enviadas por B não encontrarem fila nos caminhos de ida e volta, é possível resolver o sistema e estimar o atraso sofrido pelas sondas em cada um dos sentidos (d_{AD}, d_{DA}, d_{BD} e d_{DB}).

Estimando os tempos de transmissão e propagação

Para estimar os tempos de transmissão e de propagação, é realizado um procedimento que consiste de três fases, cada uma com gerações de sondas de tamanhos distintos. Assim como em outros trabalhos relacionados [66, 67], aqui assume-se que os tempos de propagação nos caminhos de ida e volta (AD e DA, por exemplo) são idênticos, porém, as capacidades e os tempos em fila nos enlaces percorridos nos dois sentidos podem ser diferentes. (Note que a técnica não assume caminhos simétricos, isto é, embora estejamos supondo que $T_{AD}^{prop} = T_{DA}^{prop}$, os tempos T_{AD}^{tx} e T_{AD}^{queue} podem ser diferentes de T_{DA}^{tx} e T_{DA}^{queue}.)

Na primeira fase do método, n sondas com l *bytes* são geradas de uma das máquinas fonte (vamos supor, da máquina A), para a máquina alvo D. Essas sondas são, então, replicadas pela máquina alvo D para a máquina A com o mesmo tamanho l. Em seguida, outras n sondas, desta vez com o tamanho igual a $10l$ *bytes*, são geradas de A para D e replicadas de volta para A, também com os mesmos $10l$ *bytes* de tamanho. Por fim, numa terceira fase, outras n sondas com $10l$ *bytes* são enviadas da máquina A para a máquina D. Porém, desta vez, as sondas replicadas por D não terão o mesmo tamanho daquelas enviadas por A. Nesta fase, o tamanho das sondas de D para A será igual a l *bytes*. A explicação de como ocorre o envio de sondas de diferentes tamanhos é dado a seguir.

Utilizando o protocolo *ICMP*, é trivial enviar e receber sondas de mesmo tamanho, uma vez que a especificação deste protocolo, apresentada em [51], define que o recebimento de mensagens do tipo *ICMP echo request* devem ser respondidas com uma mensagem do tipo *ICMP echo reply* de mesmo tamanho. De acordo com as especificações, para formar uma mensagem de *echo reply*, a máquina deve apenas alterar no cabeçalho da mensagem o código do tipo da mensagem *ICMP* de 8 (*echo request*) para 0 (*echo reply*), inverter os endereços de origem e destino e recalcular novo *checksum*. Os dados originais da mensagem são mantidos, preservando assim o tamanho da mensagem de resposta. Dessa forma, sondas de mesmo tamanho podem ser enviadas e recebidas. No entanto, as especificações do protocolo *ICMP* não permitem que o emissor da mensagem de *echo request* defina o tamanho das mensagens de *echo reply* a serem enviadas pelo receptor. Para contornar essa limitação, pares de pacotes são utilizados para emular o efeito do envio de um pacote de $10l$ *bytes* e o recebimento de uma resposta de tamanho l *bytes*.

Os pares de sondas são formados por um primeiro pacote *ICMP echo reply* de tamanho $10l$ *bytes*, seguido de um segundo pacote *ICMP echo request* de tamanho l *bytes*. Note que a primeira sonda do par é uma mensagem *ICMP echo reply*, gerada espontaneamente pela máquina fonte, sem que esta tenha recebido uma mensagem de *ICMP echo request*. Os pacotes do par atravessam o mesmo caminho de rede até chegarem ao destino. Neste cenário, o segundo pacote será atrasado a cada salto pelo tempo de transmissão do primeiro, uma vez que este é dez vezes maior que o segundo pacote. Ao chegarem à máquina destino, a primeira sonda será descartada pela máquina (por ser uma mensagem de *ICMP echo reply*) e uma mensagem de *ICMP echo reply* de tamanho l será imediatamente enviada de volta para a máquina de origem. Dessa forma, podemos assumir que, no sentido de ida, a segunda sonda do par sofrerá um atraso de transmissão equivalente ao de um pacote de tamanho $10l$, enquanto que, no sentido de volta, o tempo de transmissão será igual ao de um pacote de tamanho l.

Sejam $RTT_{m,ADA}^{l-l}$, $RTT_{m,ADA}^{10l-10l}$ e $RTT_{m,ADA}^{10l-l}$ os menores valores estimados para o atraso de ida e volta, dentre as n amostras geradas em cada uma das três fases, com os tamanhos especificados pelo procedimento descrito acima. Considerando um número suficiente de amostras, é comum assumir que os valores referentes aos tempos

em fila para $RTT^{l-l}_{m,ADA}$, $RTT^{10l-10l}_{m,ADA}$ e $RTT^{10l-l}_{m,ADA}$ são nulos ([64, 65, 66, 67, 68]). Assim, considerando a suposição de que os tempos de propagação são iguais nos dois sentidos ($T^{prop}_{AD} = T^{prop}_{DA}$), chega-se ao seguinte sistema de equações:

$$\begin{cases} T^{tx}_{AD} + T^{tx}_{DA} + 2T^{prop}_{AD} = RTT^{l-l}_{m,ADA} \\ 10T^{tx}_{AD} + 10T^{tx}_{DA} + 2T^{prop}_{AD} = RTT^{10l-10l}_{m,ADA} \\ 10T^{tx}_{AD} + T^{tx}_{DA} + 2T^{prop}_{AD} = RTT^{10l-l}_{m,ADA} \end{cases} \tag{3.3}$$

onde, o valor "10" é devido ao tamanho do maior pacote, 10 vezes maior que o outro; e, os valores de $RTT^{l-l}_{m,ADA}$, $RTT^{10l-10l}_{m,ADA}$ e $RTT^{10l-l}_{m,ADA}$ são conhecidos.

Este sistema é linearmente independente e fornece uma estimativa para os tempos de transmissão e propagação, em cada um dos sentidos, entre as máquinas A e D. De forma semelhante, o mesmo procedimento pode ser executado entre a máquina B e D. Desta forma, as equações lineares são obtidas e a sua solução fornece as estimativas dos tempos de transmissão e propagação para os caminhos BD e DB.

Calculando a média e variância do atraso em um sentido

As equações formadas pelo procedimento descrito acima permitem estimar os tempos de transmissão e propagação em cada um dos sentidos entre as máquinas A e D e entre B e D. O sistema previamente definido pelas de Equações 3.2 pode, então, ser reformulado da seguinte forma:

$$\begin{cases} T^{fila}_{AD} + T^{fila}_{DA} = RTT^{10l-10l}_{ADA} - [10T^{tx}_{AD} + 2T^{prop}_{AD} + 10T^{tx}_{DA}] \\ T^{fila}_{BD} + T^{fila}_{DB} = RTT^{10l-10l}_{BDB} - [10T^{tx}_{BD} + 2T^{prop}_{BD} + 10T^{tx}_{DB}] \\ T^{fila}_{AD} - T^{fila}_{BD} = \Psi_{AD-BD} - [10T^{tx}_{AD} + T^{prop}_{AD} - 10T^{tx}_{BD} - T^{prop}_{BD}] \\ T^{fila}_{DA} - T^{fila}_{DB} = \Psi_{DA-DB} - [10T^{tx}_{DA} + T^{prop}_{AD} - 10T^{tx}_{DB} - T^{prop}_{BD}] \end{cases} \tag{3.4}$$

onde, o valor "10" é devido ao tamanho considerado aqui para as sondas enviadas por A e B.

O sistema reformulado tem agora um espaço de soluções bem mais reduzido. Todos os termos conhecidos das equações foram agrupados no segundo membro das expressões. O número de incógnitas do sistema de Equações 3.4 agora é quatro. No entanto, o número de equações independentes continua sendo inferior. (Lembre-se que das quatro equações, apenas três são independentes). Logo, ainda não é possível obter uma única solução para o sistema, apenas com essas equações.

Para que o sistema de Equações 3.4 possa ser resolvido e, finalmente, sejam determinados os atrasos sofridos pelas sondas em cada um dos sentidos (d_{AD}, d_{DA}, d_{BD} e d_{DB}), informações extras são necessárias. Por exemplo, se soubermos o valor de uma das quatro incógnitas restantes no sistema de Equações 3.4, é possível resolver o sistema. Logo, se a sonda enviada por A (ou a enviada por B) tiver o tempo em fila nos caminhos de ida e volta aproximadamente iguais a zero, é possível estimar os atrasos sofridos pelas sondas em cada um dos sentidos (d_{AD}, d_{DA}, d_{BD} e d_{DB}). Isto porque, adicionando a equação $T_{AD}^{fila} = 0$ ou $T_{DA}^{fila} = 0$ ao sistema de Equações 3.4, então ele pode ser resolvido, determinando os valores das incógnitas T_{BD}^{fila} e T_{DB}^{fila}. (O mesmo vale para o caso em que T_{BD}^{fila} e T_{DB}^{fila} são nulos e, neste caso, são obtidos valores de T_{AD}^{fila} e T_{DA}^{fila}.)

Para inferir a média e a variância da distribuição do atraso em um sentido, diversas amostras deste atraso devem ser estimadas. Supondo que, de todas as sondas geradas entre as máquinas A e D e entre B e D, i amostras originadas de A e B retornaram de D com valores de IPID muito próximos; e que, dessas i amostras, o atraso em cada sentido foi estimado para j sondas. Sejam $d_{AD}(n)$, $d_{DA}(n)$, $d_{BD}(n)$ e $d_{DB}(n)$ os atrasos em um sentido estimados para a n-ésima dessas j amostras, a média e a variância amostral da distribuição do atraso em cada sentido são calculadas por:

$$\overline{d}_{sentido} = \frac{1}{j}\sum_{n=1}^{j} d_{sentido}(n) \quad e \quad Var(d_{sentido}) = \frac{1}{j-1}\sum_{n=1}^{j}\left(d_{sentido}(n) - \overline{d}_{sentido}\right)^2$$

onde, "sentido" representa o caminho desejado da métrica: AD, DA, BD ou DB

Algoritmo para estimar o atraso em um sentido usando o IPID

A solução da técnica proposta, que explora o IPID para estimar a média e variância do atraso unidirecional, pode ser resumida em três idéias básicas. Idéias essas que permitem se obter um conjunto de equações lineares e independentes, relacionando os tempos de transmissão, propagação e fila, nos dois sentidos, entre as máquinas $A - D$ e $B - D$.

- Idéia I: Transmissão de sondas de dois tamanhos distintos;
- Idéia II: Emular o efeito de transmissão de sondas de um tamanho e recebimento de outro tamanho;
- Idéia III: Dentre os conjuntos de sondas enviadas, identificar pares de sondas tal que uma tenha partido de A e outra de B e as duas tenham alcançado D no mesmo instante (semelhante à idéia de Chen et al. [55]). Além disso, formar dois subconjuntos a partir desses pares, tal que: um é formado pelos pares cujo os tempos em fila nos sentidos AD e DA sejam nulos e o outro formado pelos pares cujo os tempos em fila iguais a zero tenham ocorrido nos sentidos BD e DB.

Ademais, a única suposição do método é de que o tempo de propagação em cada um dos sentidos AD e BD sejam idênticos. Isto é, $T_{AD}^{prop} = T_{DA}^{prop}$ e $T_{BD}^{prop} = T_{DB}^{prop}$.

O Algoritmo 3.1 sintetiza um passo-a-passo do método.

3.1.2 A técnica com *IP Spoofing*

A técnica descrita na subseção anterior pressupõe que o sistema operacional da máquina alvo implementa um contador global para o IPID. Embora seja indiscutível que inúmeras máquinas na Internet atual possuem um IPID global, uma vez que essa característica é inerente ao sistema operacional Windows, relaxar tal suposição permite expandir a aplicabilidade da proposta. Assim, uma variação da técnica foi desenvolvida, permitindo que sejam computados os atrasos unidirecionais de sondas enviadas de duas (ou mais) máquinas fonte para uma máquina alvo, independente do contador de IPID implementado pelas máquinas envolvidas na medição.

Essa variação requer que ao menos uma das máquinas fontes seja capaz de enviar sondas com *IP spoofing*. Ao contrário das estimativas obtidas pelo método com IPID, o algoritmo utilizando *IP spoofing* permite computar o atraso em apenas um dos sentidos (de ida ou de volta) por vez, dependendo da máquina fonte e endereço IP de origem utilizados nas sondas enviadas para a máquina alvo. Isto é, para computar o atraso sofrido pelos pacotes no caminho de rede entre as máquinas fontes A e B para a máquina alvo D (d_{AD} e d_{BD}), sondas devem ser geradas de A e B para D, sendo que os pacotes enviados por uma das máquinas fonte (A ou B) devem conter

Algoritmo 3.1 Algoritmo da técnica utilizando IPID.

Passo 1: Gerar três sequências de n sondas das máquinas A e B para D, conforme procedimento descrito na Subseção 3.1.1. Identificar, dentre todas as amostras de atraso de ida e volta, o menor valor de RTT para cada sequência de cada uma das máquinas fonte: $RTT^{X-Y}_{m,ADA}$ e $RTT^{X-Y}_{m,BDB}$, onde $(X-Y) = (l-l), (10l-10l), (10l-l)$;

Passo 2: Utilizando o sistema de Equações 3.3, estimar os tempos de transmissão e propagação em cada um dos sentidos (AD, DA, BD e DB);

Passo 3: Gerar k_A e k_B sondas adicionais, respectivamente, de A e de B para D. (Consideramos o tamanho $10l$ para essas sondas enviadas por A e B.) Formar o conjunto $\mathcal{I}$ com i pares de amostras (s_A, s_B), onde s_A e s_B são sondas enviadas de A e B, respectivamente. O par de sondas (s_A, s_B) é selecionado se os pacotes replicados por D para A e B apresentam valores de IPID muito próximos, indicando que s_A e s_B chegaram a D aproximadamente no mesmo instante;

Passo 4: Selecionar, do conjunto $\mathcal{I}$, todos os pares de amostra (s_A, s_B) cujo o atraso em fila de uma das duas amostras seja negligível. O par i é selecionado se satisfizer uma das seguintes condições: (a) $RTT^{10l-10l}_{ADA}(i) \leq 1.01 RTT^{10l-10l}_{m,ADA}$; (b) $RTT^{10l-10l}_{BDB}(i) \leq 1.01 RTT^{10l-10l}_{m,BDB}$. Considere $\mathcal{J}_\mathcal{A}$ como sendo um subconjunto de $\mathcal{I}$, formado pelos j_A pares de amostras que satisfazem a condição (a), e $\mathcal{J}_\mathcal{B}$ o subconjunto de $\mathcal{I}$, formado pelos j_B pares de amostras que satisfazem a condição (b);

Passo 5: Para cada par existente no subconjunto $\mathcal{J}_\mathcal{A}$, obter os tempos em fila nos sentidos BD e DB e estimar uma amostra de d_{BD} e d_{DB}. Para cada par do subconjunto $\mathcal{J}_\mathcal{B}$, obter os tempos em fila nos sentidos BD e DB e estimar uma amostra de d_{AD} e d_{DA}. Isso utilizando o sistema de Equações 3.4;

Passo 6: A média e a variância do atraso em um sentido podem ser computados por:

$$\overline{d}_{sentido} = \frac{1}{j_s} \sum_{n=1}^{j_P} d_{sentido}(n)$$

$$Var(d_{sentido}) = \frac{1}{j_s - 1} \sum_{n=1}^{j_s} \left(d_{sentido}(n) - \overline{d}_{sentido} \right)^2$$

sendo que, "sentido" é substituído por AD, DA, BD ou DB.

o endereço IP da outra máquina. Para estimar os atrasos no sentido oposto, nos caminhos de D para A e B (d_{DA} e d_{DB}), todas as sondas são enviadas de uma mesma máquina (A, por exemplo), sendo que parte dessas sondas são enviadas fazendo *IP spoofing* com o endereço da outra máquina (neste caso, B).

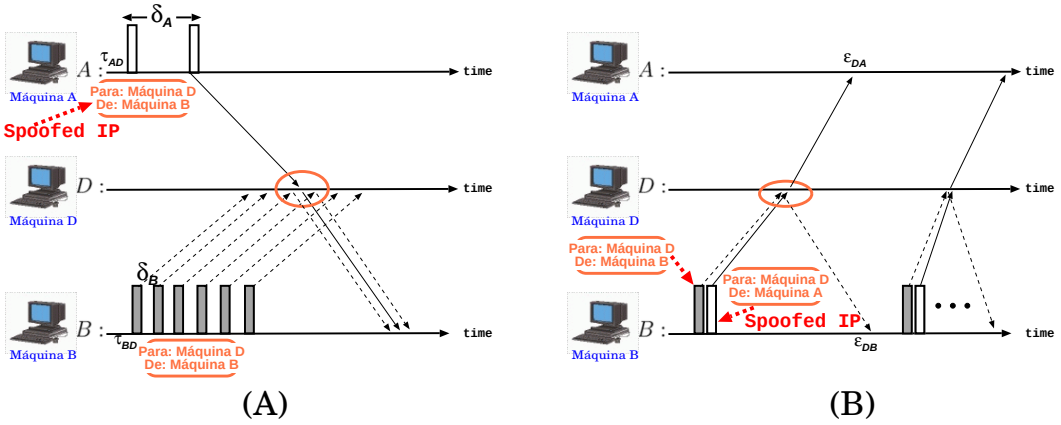


Figura 3.2: Sondas geradas das máquinas A e B para a máquina D, utilizando a técnica com *IP spoofing*.

Considere, primeiro, o caso cujo objetivo seja estimar os atrasos no sentido de ida (ou seja, d_{AD} e d_{BD}), ilustrado no cenário representado na Figura 3.2(A). Neste caso, as máquinas A e B, com relógios sincronizados, geram sondas para a máquina alvo D. No entanto, as sondas enviadas pela máquina A contêm o endereço IP de origem da máquina B. Já a geração das sondas a partir de B é feita em intervalos de tempo pequenos e constantes, e sem *IP spoofing* dos pacotes. Essas sondas são replicadas pela máquina alvo de volta às máquinas de origem, sendo que as respostas às sondas enviadas por A serão encaminhadas à máquina B, devido ao endereço IP forjado por A. Se um dos pacotes enviados originalmente por A, chegar a D entre duas sondas consecutivas enviadas por B, todas as respostas correspondentes, inclusive as enviadas por A, serão replicadas à máquina B em sequência e uma logo após a outra. Para cada par de amostras, sendo uma originalmente enviada por A e outra por B, que chegaram juntas a D e as respectivas respostas foram recebidas por B,

é possível formular o seguinte sistema de equações:

$$\begin{cases} 10T_{AD}^{tx} + T_{AD}^{prop} + T_{AD}^{fila} + 10T_{DB}^{tx} + T_{DB}^{prop} + T_{DB}^{fila} = RTT_{ADB}^{10l-10l} \\ 10T_{BD}^{tx} + T_{BD}^{prop} + T_{BD}^{fila} + 10T_{DB}^{tx} + T_{DB}^{prop} + T_{DB}^{fila} = RTT_{BDB}^{10l-10l} \\ 10T_{AD}^{tx} + T_{AD}^{prop} + T_{AD}^{fila} - (10T_{BD}^{tx} + T_{BD}^{prop} + T_{BD}^{fila}) = \Psi_{AD-BD} \end{cases} \tag{3.5}$$

onde, Ψ_{AD-BD} é a diferença entre os atrasos de A para D e de B para D ($d_{AD} - d_{BD}$). (Note que Ψ_{AD-BD} é a mesma métrica computada por Chen et al. em [55], mas agora estimada sem utilizar o IPID da máquina remota, como era feito originalmente naquele trabalho.) RTT_{BDB} é o atraso de ida e volta estimado no caminho BDB, e RTT_{ADB} é a diferença do instante de chegada do *echo reply* à máquina B e o instante de envio do *echo request* pela máquina A.

Assim como o sistema de Equações 3.2, obtido com a técnica utilizando IPID, o sistema acima apresenta um número maior de incógnitas do que de equações. No entanto, utilizando o procedimento descrito na Seção 3.1.1, é possível obter as Equações 3.3 para estimar os tempos de transmissão e propagação em cada um dos sentidos entre AD e entre BD. Com isso, é possível reformular esse sistema e obter o sistema de Equações que tem o espaço de soluções bem mais reduzido.

$$\begin{cases} T_{AD}^{fila} + T_{DB}^{fila} = RTT_{ADB}^{10l-10l} - [10T_{AD}^{tx} + T_{AD}^{prop} + T_{BD}^{prop} + 10T_{DB}^{tx}] \\ T_{BD}^{fila} + T_{DB}^{fila} = RTT_{BDB}^{10l-10l} - [10T_{BD}^{tx} + 2T_{BD}^{prop} + 10T_{DB}^{tx}] \\ T_{AD}^{fila} - T_{BD}^{fila} = \Psi_{AD-BD} - [10T_{AD}^{tx} + T_{AD}^{prop} - 10T_{BD}^{tx} - T_{BD}^{prop}] \end{cases} \tag{3.6}$$

Quando as sondas enviadas por A (forjadas com o endereço IP de B) ou as sondas enviadas por B não encontrarem fila nos caminhos de ida e volta, é possível estimar o atraso sofrido pelas sondas no sentido de ida (d_{AD} e d_{DA}) . Por exemplo, quando as sondas enviadas por B não encontram fila no caminho de ida e volta, ou seja, T_{BD}^{fila} e T_{DB}^{fila} forem iguais a zero, a solução do sistema de Equações 3.6 possibilita estimar o atraso sofrido pelas sondas no sentido de ida entre A e D (d_{AD}). De forma semelhante, quando os tempos em fila para as sondas enviadas por A forem nulos, será possível computar o atraso de ida do pacote no caminho entre BD (d_{BD}).

Quando o objetivo for calcular os atrasos no sentido de volta (d_{DA} e d_{DB}), as sondas devem ser enviadas de uma mesma máquina fonte (por exemplo, B), para a máquina alvo D. Conforme ilustra a Figura 3.2(B), nesse cenário, a cada duas

sondas enviadas consecutivamente da máquina fonte (B) para a máquina alvo (D), uma delas é enviada fazendo *spoofing* do endereço IP da outra máquina fonte (nesse caso, a máquina A). Assumindo que esses dois pacotes, enviados consecutivamente, seguem juntos ao longo de todo o caminho entre B e D, eles sofrerão o mesmo atraso nesse sentido do caminho. Essas sondas chegarão a D juntas e serão replicadas, uma para a máquina A e outra para B, em instantes muito próximos de tempo. Quando essas mensagens replicadas por D chegarem a A e a B, o seguinte sistema de equações será obtido:

$$\begin{cases} 10T_{BD}^{tx} + T_{BD}^{prop} + T_{BD}^{fila} + 10T_{DA}^{tx} + T_{DA}^{prop} + T_{DA}^{fila} = RTT_{BDA}^{10l-10l} \\ 10T_{BD}^{tx} + T_{BD}^{prop} + T_{BD}^{fila} + 10T_{DB}^{tx} + T_{DB}^{prop} + T_{DB}^{fila} = RTT_{BDB}^{10l-10l} \\ 10T_{DA}^{tx} + T_{DA}^{prop} + T_{DA}^{fila} - (10T_{DB}^{tx} + T_{DB}^{prop} + T_{DB}^{fila}) = \Psi_{DA-DB} \end{cases} \tag{3.7}$$

Reformulando o sistema, após obtidos os tempos de transmissão e propagação, o sistema de Equações 3.8 é formado. Pelas equações, os valores dos atrasos (d_{DA} ou d_{DB}) podem ser estimados, quando o tempo em fila for zero nos caminhos BDB ou BDA.

$$\begin{cases} T_{BD}^{fila} + T_{DA}^{fila} = RTT_{BDA}^{10l-10l} - [10T_{BD}^{tx} + T_{BD}^{prop} + T_{AD}^{prop} + 10T_{DA}^{tx}] \\ T_{BD}^{fila} + T_{DB}^{fila} = RTT_{BDB}^{10l-10l} - [10T_{BD}^{tx} + 2T_{BD}^{prop} + 10T_{DB}^{tx}] \\ T_{DA}^{fila} - T_{DB}^{fila} = \Psi_{DA-DB} - [10T_{DA}^{tx} + T_{AD}^{prop} - 10T_{DB}^{tx} - T_{BD}^{prop}] \end{cases} \tag{3.8}$$

Algoritmo para estimar o atraso em um sentido usando *IP spoofing*

Os algoritmos listados resumem a variação da técnica definida para estimar o atraso em um sentido utilizando *IP spoofing*. Para o caso em que o objetivo é estimar os atrasos no sentido de ida (d_{AD} e d_{BD}), utiliza-se o Algoritmo 3.2. Quando o foco for computar os atrasos no sentido de volta, é usado o Algoritmo 3.3.

Algoritmo 3.2 Algoritmo da técnica utilizando *IP spoofing* para estimar os atrasos no sentido de ida.

Passo 1: Gerar três sequências de n sondas das máquinas A e B para D, conforme procedimento descrito na Subseção 3.1.1. Identificar, dentre todas as amostras de atraso de ida e volta, o menor valor de RTT para cada sequência de cada uma das máquinas fonte: $RTT_{m,ADA}^{X-Y}$ e $RTT_{m,BDB}^{X-Y}$, onde $(X-Y) = (l-l), (10l-10l), (10l-l)$;

Passo 2: Gerar k_A e k_B sondas adicionais para D, respectivamente de A e de B, sendo que o endereço de origem dos pacotes enviados por A são forjados com o IP de B. (Consideramos o tamanho $10l$ para essas sondas enviadas por A e B.) Formar o conjunto $\mathcal{I}$, com os i pares de sondas (s_A, s_B), dentre todas as k_A e k_B amostras, cujas respostas chegaram juntas de D a B;

Passo 3: Selecionar do conjunto $\mathcal{I}$ todos os pares de amostra (s_A, s_B) cujo o atraso em fila de uma das duas amostras seja negligível. O par i é selecionado se satisfizer uma das seguintes condições: (i) se $RTT_{ADB}^{10l-10l}(i) \leq 1.01 RTT_{m,ADB}^{10l-10l}$ ou (ii) $RTT_{BDB}^{10l-10l}(i) \leq 1.01 RTT_{m,BDB}^{10l-10l}$. Sejam $\mathcal{J}_\mathcal{A}$ um subconjunto de $\mathcal{I}$, formado pelos j_A pares de amostras que satisfazem a condição (i), e $\mathcal{J}_\mathcal{B}$ um subconjunto de $\mathcal{I}$, formado pelos j_B pares de amostras que satisfazem a condição (ii);

Passo 4: Para cada par existente no subconjunto $\mathcal{J}_\mathcal{A}$, estimar uma amostra de d_{BD}, e para cada par do subconjunto $\mathcal{J}_\mathcal{B}$, estimar uma amostra de d_{AD}, utilizando as Equações 3.6;

Passo 5: A média e a variância do atraso em um sentido podem ser computados por:

$$\overline{d}_{sentido} = \frac{1}{j_s} \sum_{n=1}^{j_P} d_{sentido}(n)$$

$$Var(d_{sentido}) = \frac{1}{j_s - 1} \sum_{n=1}^{j_s} \left(d_{sentido}(n) - \overline{d}_{sentido} \right)^2$$

sendo que, "sentido" é substituído por AD ou BD.

Algoritmo 3.3 Algoritmo da técnica utilizando *IP spoofing* para estimar os atrasos no sentido de volta.

Passo 1: Gerar três sequências de n sondas das máquinas A e B para D, conforme procedimento descrito na Subseção 3.1.1. Identificar, dentre todas as amostras de atraso de ida e volta, o menor valor de RTT para cada sequência de cada uma das máquinas fonte: $RTT^{X-Y}_{m,ADA}$ e $RTT^{X-Y}_{m,BDB}$, onde $(X-Y) = (l-l), (10l-10l), (10l-l)$;

Passo 2: Gerar k_A e k_B sondas para D , todas de B, sendo que a transmissão de cada uma das k_A sondas deve ser feita imediatamente após o envio de uma das k_B sondas. (Consideramos o tamanho $10l$ para essas sondas.) Apesar de serem enviados por B, nos k_A pacotes é feito *IP spoofing* e utilizado o IP de A como endereço de origem dos pacotes. Formar o conjunto $\mathcal{I}$ com todos os i pares de sondas (s_A, s_B) enviadas consecutivamente por B e que suas respectivas respostas chegaram, respectivamente, às máquinas A e B.

Passo 3: Selecionar do conjunto $\mathcal{I}$ todos os pares de amostra (s_A, s_B) cujo o atraso em fila de uma das duas amostras seja negligível. O par i é selecionado se satisfizer uma das seguintes condições: se (i) $RTT^{10l-10l}_{BDA}(i) \leq 1.01 RTT^{10l-10l}_{m,BDA}$ ou (ii) $RTT^{10l-10l}_{BDB}(i) \leq 1.01 RTT^{10l-10l}_{m,BDB}$. Sejam $\mathcal{J}_\mathcal{A}$ um subconjunto de $\mathcal{I}$, formado pelos j_A pares de amostras que satisfazem a condição (i), e $\mathcal{J}_\mathcal{B}$ um subconjunto de $\mathcal{I}$, formado pelos j_B pares de amostras que satisfazem a condição (ii).

Passo 4: Para cada par existente no subconjunto $\mathcal{J}_\mathcal{A}$, estimar uma amostra de d_{DB}, e para cada par do subconjunto $\mathcal{J}_\mathcal{B}$, estimar uma amostra de d_{DA}, utilizando as Equações 3.8;

Passo 5: A média e a variância do atraso em um sentido podem ser computados por:

$$\overline{d}_{sentido} = \frac{1}{j_s} \sum_{n=1}^{j_P} d_{sentido}(n)$$

$$Var(d_{sentido}) = \frac{1}{j_s - 1} \sum_{n=1}^{j_s} \left(d_{sentido}(n) - \overline{d}_{sentido} \right)^2$$

sendo que, "sentido" é substituído por DA ou DB.

3.2 Extensão da técnica para fontes não sincronizadas

A técnica descrita na seção anterior pressupõe o uso de sondas geradas por máquinas com relógios sincronizados. Nesta seção será demonstrada como essa suposição pode ser relaxada, estendendo a técnica para o caso em que os relógios das máquinas fonte não estejam sincronizados. Os problemas para estimar o atraso unidirecional dos pacotes, entre duas máquinas que não possuem os seus relógios sincronizados, já foi amplamente discutido na Seção 2.1.3 desta tese, assim como as soluções existentes na literatura [64, 65, 66, 67, 68, 71].

Para estimar o *Skew* e o *Offset* existente entre duas máquinas (A e B, por exemplo), as técnicas existentes requerem que sondas sejam geradas diretamente entre elas. Entretanto, na técnica descrita na Seção 3.1, as sondas não são geradas diretamente de A para B ou vice-versa, mas sim de A e B para uma máquina alvo D. Ao passo que, para utilizar os algoritmos existentes na literatura, sondas extras deveriam também ser geradas entre A e B, causando uma sobrecarga ainda maior na rede. Assim, uma abordagem nova foi definida para tratar esses dois problemas, sem a necessidade de que sondas sejam geradas diretamente entre A e B.

Para tratar o problema de *Skew*, o algoritmo apresentado por Zhang et al. em [68] foi adaptado. O método proposto em [68] é baseado na estimativa do limite inferior do fecho convexo da sequência $\Omega = [(\tau_{AB}(r), d_{AB}(r)) : r = 1, \ldots, i]$, onde $\tau_{AB}(r)$ é o instante de envio da r-ésima sonda da sequência e $d_{AB}(r)$ o atraso computado no destino, incluindo os valores de *Skew* e *Offset*. Como a técnica definida nesta tese não gera sondas entre A e B para computar os valores de d_{AB}, o método adaptado prevê uma definição diferente para a sequência Ω.

Seja, então, $\Omega := [(\tau_{AD}(r), d_{AD-DB}(r)) : r = 1, \ldots, i]$ uma sequência obtida das coletas dos i pares de sondas que chegaram à máquina D aproximadamente no mesmo instante. Essa sequência Ω pode ser formada tanto com a variação da técnica utilizando IPID, quanto para a variação que faz uso de pacotes com *IP spoofing*. Nos dois casos, $\tau_{AD}(r)$ equivale ao instante de envio por A da sonda pertencente ao r-ésimo par da sequência Ω. Já os valores de d_{AD-DB} na sequência dependem da variação da técnica adotada. Para a variação utilizando IPID, $d_{AD-DB}(r)$ equivale

à diferença entre o instante de recebimento na máquina B e o instante de envio na máquina A das respectivas sondas pertencentes ao r-ésimo par da sequência Ω. No caso da variação da técnica utilizando pacotes com *IP spoofing*, $d_{AD-DB}(r)$ é igual a $RTT_{ADB}(r)$, diferença entre os instantes de chegada do *echo reply* à máquina B e de envio do *echo request* pela máquina A, fazendo *IP spoofing* do pacote com o endereço de B.

A Figura 3.3(A) ilustra uma sequência Ω formada a partir das coletas de um experimento que será descrito na próxima seção.

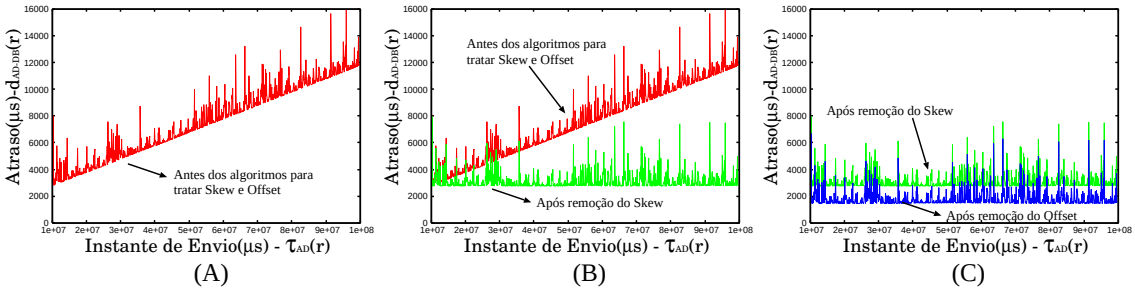


Figura 3.3: Tratamento dos problemas de Skew e Offset nas coletas.

Na Figura 3.3(A) é possível verificar a tendência de crescimento nos valores computados dos atrasos das amostras. Essa tendência é causada pela diferença nas taxas dos relógios. A sequência Ω permite identificar um limite inferior para os valores de $d_{AD-DB}(r)$. Esse limite é definido pela soma dos tempos de transmissão e propagação nos caminhos de A para D e de D para B, acrescido dos valores causados pelo *Skew* e *Offset*. Assim como em [68], o objetivo é estimar uma função linear que esteja abaixo e mais próxima possível de todos os pontos em Ω. Esta função representa a tendência de crescimento ou decrescimento entre os relógios das máquinas e pode ser removida da coleta.

Tratado o problema da diferença entre as taxas de crescimento dos relógios, uma nova sequência γ é então gerada após o cálculo do atraso sem *Skew* (d^s_{AD-DB}) para todas as r sondas. Esta sequência está ilustrada na Figura 3.3(B). É importante perceber que, como os relógios não se encontram sincronizados no início da medição, os valores estimados de d^s_{AD-DB} na sequência γ contém o *Offset* inicial da coleta. Portanto, podemos assumir que

$$d^s_{AD-DB}(r) = T^{tx}_{AD}(r) + T^{prop}_{AD}(r) + T^{fila}_{AD}(r) + T^{tx}_{DB}(r) + T^{prop}_{DB}(r) + T^{fila}_{DB}(r) + O_{AB}$$

O algoritmo apresentado em [67] poderia ser utilizado para estimar e remover o *Offset* da coleta. No entanto, sondas deveriam ser geradas da máquina A para a máquina B e vice-versa. Evitando que sondas extras sejam geradas, a estimativa do *Offset* pode ser feita a partir da diferença entre os menores valores computados para RTT_{BDB} e d_{AD-DB} dentre todas as r amostras. Se considerarmos que os menores valores destas amostras representam o caso em que estas sondas não experimentaram fila ao longo dos seus caminhos de rede, podemos definir $d^s_{m,AD-DB}$ como sendo o menor valor de d^s_{AD-DB} existente entre as r sondas da sequência γ e $RTT_{m,BDB}$ como sendo o menor valor do atraso de ida e volta computado para as sondas enviadas de B para D. Assim,

$$RTT_{m,BDB} = T^{tx}_{BD} + T^{prop}_{BD} + T^{tx}_{DB} + T^{prop}_{DB}$$

e

$$d^s_{m,AD-DB} = T^{tx}_{AD} + T^{prop}_{AD} + T^{tx}_{DB} + T^{prop}_{DB} + O_{AB}$$

A diferença entre $RTT_{m,BDB}$ e $d_{m,AD-DB}$ é então:

$$RTT_{m,BDB} - d_{m,AD-DB} = (T^{tx}_{BD} + T^{prop}_{BD}) - (T^{tx}_{AD} + T^{prop}_{AD}) + O_{AB}$$

Como os valores dos tempos de transmissão e propagação em cada sentido são conhecidos, independente da existência ou não de problemas como *Skew* e *Offset*. Então, é possível estimar o O_{AB} da seguinte forma:

$$O_{AB} = RTT_{m,BDB} - d_{m,AD-DB} - (T^{tx}_{BD} + T^{prop}_{BD}) + (T^{tx}_{AD} + T^{prop}_{AD})$$

A Figura 3.3(C) ilustra os valores dos atrasos em um sentido estimados para a sequência Ω, após removidos os valores de *Skew* e *Offset*.

3.3 Experimentos e validações

A fim de validar a técnica proposta e avaliar a sua eficácia, foram realizados tanto experimentos na Internet como utilizados modelos de simulação. Os resultados obtidos para as duas variações da técnica serão apresentados nesta seção.

3.3.1 Experimentos reais na Internet

Uma série de experimentos foram executados utilizando diferentes cenários. Os experimentos foram realizados na Internet e parte deles envolveram máquinas do ambiente PlanetLAB [118].

Inicialmente, foram utilizadas máquinas fonte sincronizadas por GPS. Por isso, neste primeiro conjunto de resultados, não houve a necessidade de tratar os problemas de *Skew* e *Offset*. Serão apresentados cinco resultados em que foram utilizadas máquinas fonte com relógios sincronizados: três para o algoritmo utilizando o IPID e dois para o algoritmo usando *IP Spoofing*.

Exceto quando mencionado explicitamente, as taxas de geração das sondas utilizadas por cada uma das duas fontes foram 100 e 1000 pacotes por segundo. Considerando que a maioria dos pacotes são de tamanho $l = 50$ bytes, a sobrecarga introduzida na rede por cada uma das máquinas fonte é, respectivamente, 40 kbps e 400 kbps. Para as altas taxas de transmissão alcançadas atualmente pelas redes, esse tráfego não pode ser considerado intrusivo para a rede.

Experimentos com a técnica utilizando IPID

O primeiro resultado avalia a precisão da técnica utilizando o algoritmo com IPID, executando medições com três máquinas, sendo duas delas localizadas no Brasil (uma na UFRJ e outra na UNIFACS) e a terceira nos Estados Unidos (UMass-Amherst). Em cada rodada do experimento, com 30 minutos de duração, foram utilizadas duas das três máquinas como fontes (A e B) e a terceira máquina como alvo (D). Sendo que, as máquinas utilizadas como fontes e alvo alternaram entre uma rodada e outra de experimento. Para cada rodada foram estimadas a média e a variância do atraso unidirecional, em cada um dos sentidos (AD, DA, BD e DB), através da técnica proposta utilizando o algoritmo com IPID. Apenas para validação, foram também medidos os atrasos reais sofridos pelas sondas. A Tabela 3.1 sintetiza os resultados dos três experimentos executados, através dos erros relativos computados para a média e variância dos atrasos estimados em relação aos valores reais dos atrasos unidirecionais em cada um dos caminhos medidos. Os baixos valores dos erros relativos indicam a boa precisão da técnica neste cenário.

O segundo conjunto de experimentos com o algoritmo IPID foi realizado uti-

Tabela 3.1: Erro relativo - experimentos UFRJ, Unifacs e UMass.

Caminho	Erro relativo Média/Variância	Caminho	Erro relativo Média/Variância	Caminho	Erro relativo Média/Variância
UFRJ-UMass	0.004 / 0.626	UFRJ-Unifacs	0.009 / 0.152	UMass-UFRJ	0.042 / 0.005
UMass-UFRJ	0.005 / 0.022	Unifacs-UFRJ	0.009 / 0.038	UFRJ-UMass	0.041 / 0.049
Unifacs-UMass	0.016 / 0.710	Umass-Unifacs	0.001 / 0.015	Unifacs-UFRJ	0.007 / 0.475
UMass-Unifacs	0.015 / 0.087	Unifacs-Umass	0.001 / 0.099	UFRJ-Unifacs	0.052 / 0.076

lizando três máquinas do PlanetLab e em horários distintos do dia. Nesses experimentos, máquinas fonte, localizadas em Seattle e no Texas, geraram sondas para uma máquina de destino na Coréia durante o primeiro minuto de cada hora, por 10 horas (entre 5am-3pm GMT). Cada sessão de um minuto foi dividida em 6 subsessões de 10 segundos de duração. Para cada subsessão, foram estimadas a média e variância do atraso durante aquela sucessão. Com os 6 valores de média e variância, para cada uma das 10 sessões, foi computada a média amostral e o intervalo de confiança para um nível de significância de 95%. O objetivo foi investigar se a técnica seria capaz de capturar, com precisão, o comportamento das métricas em diferentes períodos do dia. As Figuras 3.4 (A) e (B) mostram os intervalos de confiança dos valores estimados pelo nosso algoritmo e dos valores reais, para o caminho Coréia-Seattle. Os resultados demonstram a precisão da técnica em suas estimativas.

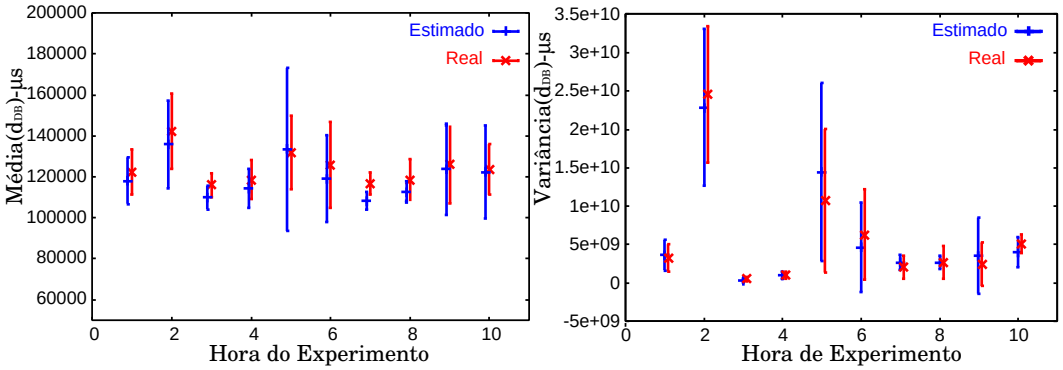


Figura 3.4: Intervalo de confiança da média (A) e variância (B) do atraso computado no caminho Coréia-Seattle.

No terceiro conjunto de experimentos, máquinas fonte no Texas, Stanford, Berkeley, Unifacs, Kaist, França, Israel, Reino Unido e Hong Kong geraram sondas simultaneamente para uma máquina de destino na UMass, que enviou as sondas de volta com um valor global de IPID. O objetivo principal deste experimento foi investigar o atraso unidirecional estimado em vários caminhos, a partir de diferentes máquinas

fontes para um mesmo alvo. A Figura 3.5 ilustra o valor médio do atraso estimado pela técnica (e os valores reais para comparação) dos caminhos cada uma das máquinas fonte e a máquina da UMass. Este experimento demonstra que a técnica poderia ser utilizada, por exemplo, como uma solução para aplicações que desejem escolher o melhor "caminho" (isto é, com o valor mínimo para a média e/ou variância do atraso) para atender um pedido de uma máquina cliente (neste exemplo, a máquina da UMass). Nota-se que, para este experimento, o caminho "Texas-UMass" foi o que obteve a menor média do atraso, dentre todos os caminho mensurados.

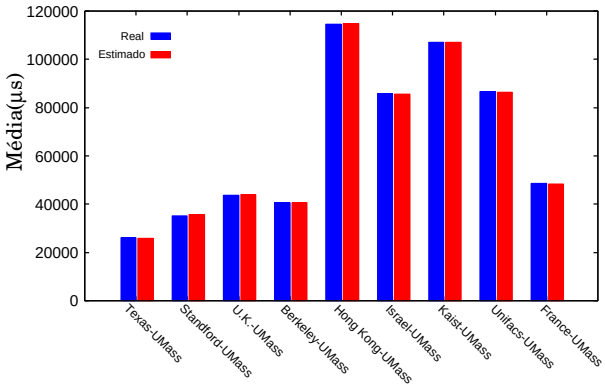


Figura 3.5: Experimento simultâneo, envolvendo diversas máquinas fonte para uma máquina alvo, usando o algoritmo de IPID.

Experimentos com a técnica usando *IP Spoofing*

No próximo conjunto de experimentos, foi utilizado o algoritmo com *IP Spoofing* para estimar o atraso entre duas máquinas fonte (uma na UFRJ e outra na UMass) e uma máquina alvo localizada no Japão. O objetivo deste experimento foi avaliar o algoritmo com *IP Spoofing* para estimar a média e variância do atraso. Nesses experimentos, sondas foram geradas pelas máquinas da UFRJ e UMass para a máquina no Japão durante 3 minutos. Os pacotes originados da UFRJ foram enviados com o endereço IP de origem falsificado, contendo o endereço da máquina da UMass. A Tabela 3.2 apresenta os resultados do experimento para as estimativas do atraso entre as máquinas fonte e a máquina alvo. Experimentos também foram realizados para calcular o atraso na direção inversa (do Japão para a UFRJ e UMass) e as estimativas obtidas são apresentadas na Tabela 3.3. As Tabelas 3.2 e 3.3 também

mostram os baixos valores de erro relativo obtidos pelos experimentos, o que reforça a precisão da técnica quando é utilizado o algoritmo com *IP Spoofing*.

Tabela 3.2: Atraso da UFRJ e da UMass para máquina alvo no Japão.

Caminho	Média	Variância
	Estimado(μs) / Real(μs) / Erro Relativo	Estimado / Real / Erro Relativo
UMass-JP	92829 / 95469 / 0.0284	11499500 / 20513285 / 0.4394
UFRJ-JP	189084 / 190643 / 0.0081	3132221465 / 3227306843 / 0.0294

Tabela 3.3: Atraso da máquina alvo no Japão para a UFRJ e UMass.

Caminho	Média	Variância
	Estimado(μs) / Real(μs) / Erro Relativo	Estimado / Real / Erro Relativo
JP-UMass	92068 / 93543 / 0.0157	179157696 / 249516602 / 0.2819
JP-UFRJ	172809 / 174006 / 0.0068	11325595 / 16741362 / 0.3234

Outro conjunto de medições foi realizado simultaneamente para estimar média e variância do atraso nos caminhos entre diversas máquinas fonte e uma mesma máquina alvo. O objetivo foi investigar novamente o atraso de diferentes caminhos até uma máquina alvo, desta vez usando o algoritmo com *IP Spoofing*. No experimento, máquinas localizadas na UFRJ, UMass e outras pertencentes ao ambiente do PlanetLAB (UCLA, Reino Unido, Berkeley e Japão) geraram sondas para a máquina de destino localizada na Universidade de Columbia (também pertencente ao PlanetLAB). Os pacotes com endereço IP forjados foram enviados pela máquina da UFRJ. Cada sessão de medição teve duração de 10 minutos. A Figura 3.6 apresenta a média estimada para o atraso unidirecional de todas as 6 máquinas fonte para a máquina da Columbia. Pelo gráfico é possível notar que os valores estimados e valores reais estão muito próximos, e que o caminho "UMass-Columbia" foi o que apresentou a menor média de atraso dentre todos os caminhos medidos. Os resultados para as estimativas da média e da variância computadas nesta sessão de experimentos estão sucintamente apresentados na forma de erros relativos na Tabela 3.4 e confirmam novamente a precisão da técnica com o algoritmo de *IP Spoofing*.

Experimentos com relógios não sincronizados

Todos os conjuntos de experimentos descritos até aqui têm sido executados utilizando máquinas fonte com relógios sincronizados. Com o objetivo de validar a

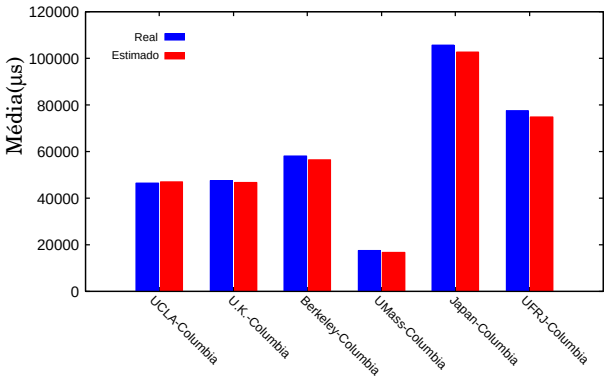


Figura 3.6: Experimento simultâneo, envolvendo diversas máquinas fonte para uma máquina alvo, usando o algoritmo de *IP Spoofing*.

Tabela 3.4: Erro relativo do experimento simultâneo utilizando o algoritmo de *IP Spoofing*.

Caminho	Média	Variância
	Erro Relativo	Erro Relativo
UCLA-Columbia	0.010	0.109
U.K.-Columbia	0.017	0.072
Berkeley-Columbia	0.028	0.331
UMass-Columbia	0.047	0.566
Japão-Columbia	0.026	0.039
UFRJ-Columbia	0.034	0.060

extensão da técnica definida para lidar com problemas como *Skew* e *Offset* (descrita na Seção 3.2), experimentos foram executados envolvendo máquinas fonte cujos relógios não estavam sincronizados. O cenário utilizado nestes experimentos consiste de três máquinas fonte (digamos, A, $B1$ e $B2$), em que uma delas ($B1$, neste caso) não possui qualquer artifício para sincronizar seu relógio, tais como NTP ou GPS. Durante os experimentos, foram utilizados dois pares de geradores de sonda, um par sem os relógios sincronizados (formado pelas máquinas A e $B1$) e outro par formado pelas máquinas A e $B2$ que têm seus relógios perfeitamente sincronizados através de GPS. Como ilustrado na Figura 3.7 (A) e (B), em ambos os casos, as sondas foram geradas para a mesma máquina de destino D. Note que as máquinas $B1$ e $B2$ estão localizadas na mesma rede local e, portanto, podemos supor que os valores da média e variância do atraso de $B1$ a D são iguais aos valores de $B2$ a D, se medidos simultaneamente.

Para estimar a média e variância do atraso das sondas geradas a partir do par A e $B1$, é necessário compensar a falta de sincronismo entre os relógios com a extensão da

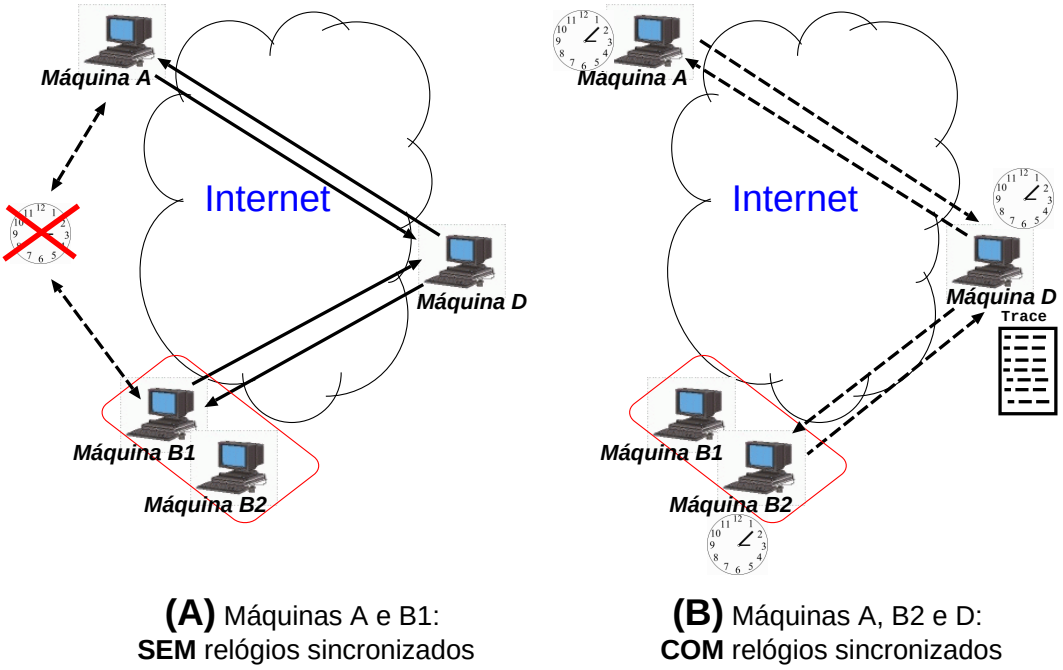


Figura 3.7: Cenário utilizado para validação da extensão da técnica.

técnica (descrita na Seção 3.2). Para que fosse possível validar essas estimativas, os instantes de chegada das sondas enviadas pelo par A e $B2$ são coletados pela máquina D. Como essa máquina é também equipada com um GPS, é possível estimar os valores da média e variância reais do atraso nos caminhos entre as máquinas A e $B2$ para a máquina D e compará-los com os valores estimados com a extensão da técnica para os caminhos de A e $B1$ para D.

Três resultados, obtidos com o algoritmo usando o IPID e considerando a extensão da técnica para o caso em que as fontes não encontram-se sincronizadas, são mostrados nas Tabelas 3.5, 3.6 e 3.7. Em todos os três experimentos, $B1$ e $B2$ foram sempre máquinas localizadas numa mesma rede local do laboratório LAND/UFRJ, enquanto as máquinas de A e D variaram de acordo com cada um dos experimentos. Por exemplo, nos resultados apresentados na Tabela 3.5, as máquinas do PlanetLab, localizada no Reino Unido e na Coréia foram utilizadas como A e D, respectivamente. Já os experimentos cujos resultados são mostrados na Tabela 3.6, foram executados usando uma máquina A em Berkeley (pertencente ao PlanetLab) e a máquina D na UMass. Por fim, os resultados apresentados na Tabela 3.6 foram obtidos a partir de experimentos cuja máquina A foi uma máquina do PlanetLab localizada no Reino Unido (U.K.) e a máquina D na UMass.

Tabela 3.5: Resultados dos experimentos usando máquinas não sincronizadas (da UFRJ e U.K. para Coréia) - Usando algoritmo IPID.

Caminho	Média	Variância
	Estimado(μs) / Real(μs) / Erro Relativo	Estimado / Real / Erro Relativo
UFRJ-Coréia	179878 / 181312 / 0.0079	17599124 / 25076445 / 0.2981
Coréia-UFRJ	173610 / 170890 / 0.0159	26191355 / 20269163 / 0.2921
UK-Coréia	157369 / 163038 / 0.0347	12977318 / 16092578 / 0.1935
Coréia-UK	143778 / 137527 / 0.0454	1187083357 / 1184729945 / 0.0019

Tabela 3.6: Resultados dos experimentos usando máquinas não sincronizadas (da UFRJ e Berkeley para UMass) - Usando algoritmo IPID.

Caminho	Média	Variância
	Estimado(μs) / Real(μs) / Erro Relativo	Estimado / Real / Erro Relativo
UFRJ-UMass	94929 / 91551 / 0.0368	6665930 / 7440538 / 0.1041
UMass-UFRJ	96262 / 99675 / 0.0342	20739281 / 19045402 / 0.0889
Berkeley-UMass	35152 / 30098 / 0.1679	3542833 / 2828705 / 0.2524
UMass-Berkeley	40580 / 45172 / 0.1016	336687495 / 395117089 / 0.1478

Tabela 3.7: Resultados dos experimentos usando máquinas não sincronizadas (da UFRJ e U.K. para UMass) - Usando algoritmo IPID.

Caminho	Média	Variância
	Estimado(μs) / Real(μs) / Erro Relativo	Estimado / Real / Erro Relativo
UFRJ-UMass	93137 / 92107 / 0.011	5793954 / 6392301 / 0.094
UMass-UFRJ	97553 / 98697 / 0.012	6241695 / 7765052 / 0.196
UK-UMass	48231 / 47189 / 0.022	2225373 / 1110086 / 1.005
UMass-UK	54227 / 55383 / 0.020	27928963 / 64791866 / 0.569

Um último experimento para validar a extensão da técnica com a variação *IP Spoofing* do algoritmo foi realizado. Neste experimento, a máquina A (localizada em Hong Kong) e as máquinas $B1$ e $B2$ (localizadas na rede local do LAND/UFRJ) enviaram sondas para uma máquina alvo D, no Texas. O *IP spoofing* é feito nos pacotes enviados das máquinas $B1$ e $B2$ enviados da UFRJ, forjando o endereço da máquina de Hong Kong. Após a remoção dos valores relativos aos problemas de *Skew* e *Offset* das coletas, foram estimadas a média e variância do atraso nos caminhos Texas-UFRJ e Texas-Hong Kong. Os resultados obtidos são listados na Tabela 3.8.

Exceto para o caso da variância computada no caminho *UK-UMass*, todos os erros relativos, apresentados nas Tabelas 3.5, 3.6, 3.7 e 3.8, apontam estimativas muito precisas com o uso da extensão da técnica. A explicação para a imprecisão no cálculo da variância é a pequena quantidade de amostras de sondas obtidas no

Tabela 3.8: Resultados dos experimentos usando máquinas não sincronizadas (da UFRJ e Hong Kong para Texas) - Usando algoritmo *IP Spoofing*.

Caminho	Média	Variância
	Estimado(μs) / Real(μs) / Erro Relativo	Estimado / Real / Erro Relativo
Texas-UFRJ	62006 / 66068 / 0.061	220503272 / 206525851 / 0.067
Texas-Hong Kong	153832 / 150263 / 0.023	60731332 / 90811542 / 0.331

caminho *UK-UMass* (menos de 5% dos pares de amostras que chegaram juntas à máquina alvo). Embora os resultados, obtidos ao longo do período de experimentos desta tese, tenham demonstrado que a precisão das estimativas da variância sejam mais sensíveis ao número pequeno de amostras, poucas amostras podem ocasionar também imprecisões na estimativa da média. Por isso, as estimativas só são consideradas confiáveis quando ao menos 10% dos pares coletados servirem para computar a variância amostral e 5% para computar a média.

3.3.2 Simulação

Os resultados experimentais apresentados na seção anterior demonstraram a precisão da técnica em ambientes reais. No entanto, os resultados apresentados, obtidos através de experimentos executados na Internet, não permitiram responder a uma outra questão importante: qual a influência que a sobrecarga na utilização da largura de banda dos roteadores, localizados ao longo dos caminhos entre as máquinas fonte e a máquina alvo, pode causar à precisão das estimativas fornecidas pelo algoritmo? Para analisar esta questão, simulações foram realizadas no ambiente de modelagem *TANGRAM-II* [74, 75, 76].

Descrição do modelo

A Figura 3.8 ilustra o cenário modelado no simulador. No modelo, os objetos *Host_A* e *Host_B* representam as máquinas fonte, geradoras de sondas. As sondas geradas seguem pelos caminhos de rede (formado pelos objetos *Router*) até a máquina alvo (objeto *Host_Target*). Quando recebida pela máquina alvo, as sondas são replicadas e enviadas de volta pela rede às máquinas fonte.

Inicialmente, o modelo foi desenvolvido com um contador global para o IPID da máquina alvo. Neste caso, as sondas replicadas pela máquina alvo possuem o valor

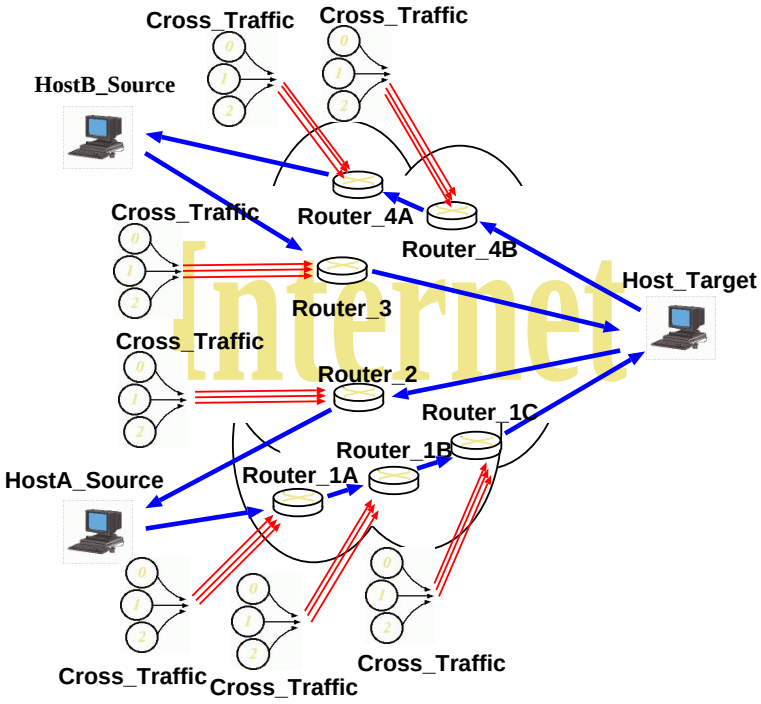


Figura 3.8: Cenário do modelo utilizado nas simulações.

atual do IPID implementado no objeto *Host_Target*. Em seguida, esse contador global foi removido do modelo e implementado, nos objetos *Host_A* e *Host_B*, o mecanismo para envio de sondas forjando a origem e emular o *IP spoofing* nos pacotes.

Distintas capacidades de transmissão foram atribuídas aos canais ligados aos roteadores e às máquinas fonte e alvo. Além das sondas, tráfego concorrente também passa pelos canais que interligam os roteadores. As sondas geradas pelas máquinas fonte e replicadas pela máquina alvo são encaminhadas aos seus destinos ou ao próximo roteador. Já os pacotes de tráfego concorrente são roteados para outros caminhos da rede.

O tráfego concorrente, injetado em cada roteador da rede, é gerado por diversas fontes *On-Off*. O tempo de permanência nos estados *On* e *Off* dessas fontes é modelado por uma distribuição Pareto com parâmetro $\alpha < 2$. Em [119] foi mostrado que a agregação destas fontes produz um tráfego com características de dependência de longa duração e que este modelo é adequado para caracterizar o tráfego real de uma rede.

Diversas simulações foram executadas variando os parâmetros das fontes de tráfego concorrente e, consequentemente, a utilização dos canais dos roteadores.

Ao término de cada rodada de simulação, foram estimadas a média e a variância, em cada um dos sentidos, utilizando os algoritmos propostos. A título de comparação, o modelo também computa os valores reais dessas métricas. A análise da eficiência da técnica se deu através da comparação entre os valores estimados e os valores reais. Alguns dos resultados obtidos são apresentados a seguir: três para a variação do algoritmo com o IPID e um para a versão com o *IP Spoofing*.

Em uma das simulações, conforme indicado no texto da descrição dos resultados, foram definidos relógios não sincronizados nas máquinas fonte, o que obrigou a utilização da extensão da técnica para tratar os problemas de *Skew* e *Offset*. Os demais resultados são das simulações considerando os relógios das máquinas fonte sincronizados.

Análise dos resultados

Os três primeiros resultados apresentados referem-se a simulações utilizando o algoritmo com IPID. Em uma dessas simulações, os parâmetros das fontes de tráfego concorrente foram ajustados para que a utilização dos canais ao longo do tempo de simulação variasse de 30% a 50% (intervalo típico de operação de uma rede). Os gráficos (A) e (B) da Figura 3.9 mostram, respectivamente, as estimativas da média e variância para o caminho DB em função do tempo de simulação. Nota-se para o resultado inicial de 20 segundos de simulação, os valores estimados são imprecisos. Isso ocorre porque o número de amostras é ainda pequeno para se obter uma estimativa precisa da média e variância do atraso. Após passados 40 segundos de simulação, a precisão já é muito boa.

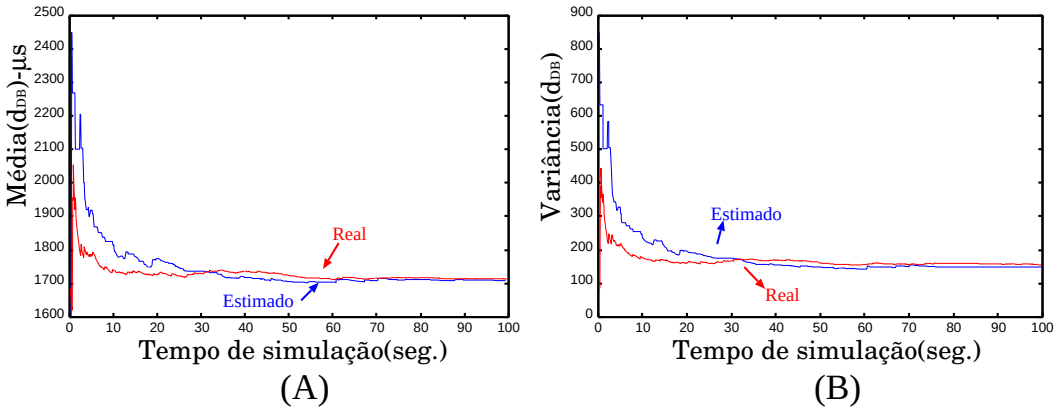


Figura 3.9: Média e variância do atraso no caminho DB (utilização entre 30 e 50%).

A Figura 3.10 apresenta os resultados para o caminho AD, quando a utilização varia entre 65% e 80% e os relógios das máquinas fonte não estão sincronizados. Quando comparado aos resultados para uma utilização mais moderada, percebe-se pelos gráficos mostrados na Figura 3.10 que o tempo de simulação necessário para que os valores estimados se aproximem dos reais é bem maior. Isto é esperado uma vez que, quanto maior for a utilização dos roteadores ao longo do caminho medido, menor será o número de amostras para a estimativa das medidas. No entanto, é possível notar que a média e variância estimadas rapidamente convergem para os valores reais, mesmo para a alta utilização definida nesta simulação.

A Tabela 3.9 sumariza os erros relativos da média e variância computados nas duas primeiras rodadas de simulação. Os erros relativos são menores que 2% (média) e 13% (variância), quando as utilizações variam entre baixas a moderadas. Com utilizações mais altas, os erros relativos para a média e variância são menos de 8% e 29%, respectivamente.

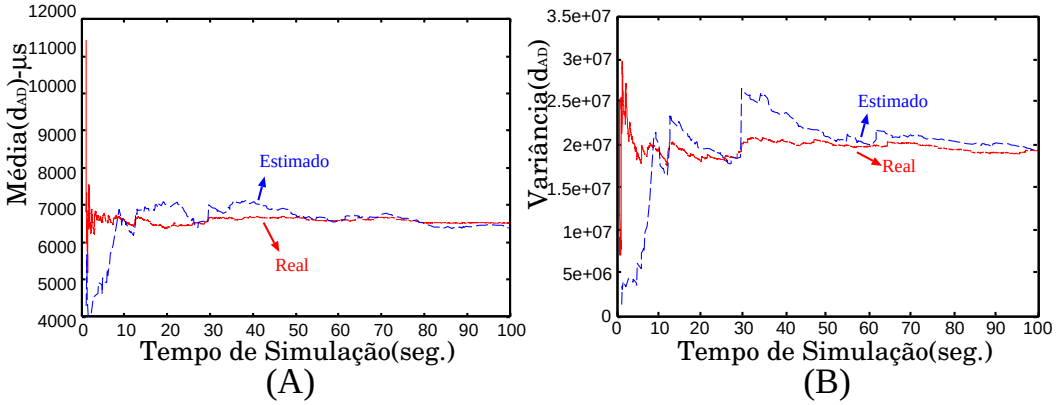


Figura 3.10: Média e variância do atraso no caminho AD (utilização entre 65 e 80%).

Tabela 3.9: Erro relativo computado nas duas primeiras rodadas de simulação com o algoritmo IPID.

Caminho	Utilizações baixas a moderadas	Utilizações altas
	Média / variância	Média / variância
AD	0.020/ 0.058	0.025 / 0.001
DA	0.013 / 0.011	0.082 / 0.290
BD	0.013 / 0.132	0.057 / 0.220
DB	0.002 / 0.033	0.062 / 0.078

No terceiro cenário, várias rodadas de simulação foram executadas variando a

utilização de apenas dois canais do modelo. O objetivo foi analisar o intervalo de confiança das estimativas para diferentes cargas nos roteadores da rede. A utilização, em todos os canais, foi fixada em aproximadamente 50%, exceto o canal entre os roteadores $1B$ e $1C$ (no caminho AD) e o canal entre o roteador $4B$ a $4A$ (do caminho DB). Diversas rodadas foram executadas variando a utilização destes canais para sete diferentes valores (de 20% a 80% de utilização). Para cada valor de utilização, foram executadas 12 rodadas de simulação, estimadas as médias e variância de todos os caminhos e calculado o intervalo de confiança dessas estimativas, considerando 95% no nível de significância. As Figuras 3.11 (A) e (B) mostram os resultados da média e variância do caminho AD e 3.12 (A) e (B) mostram os resultados para o caminho DB. Podemos observar nos gráficos que os intervalos de confiança das médias e variâncias estimadas pela técnica através do algoritmo com IPID e os valores reais dessas medidas são muito próximos. Esse resultados evidenciam a eficiência da abordagem proposta para diferentes utilizações dos canais ao longo da rede.

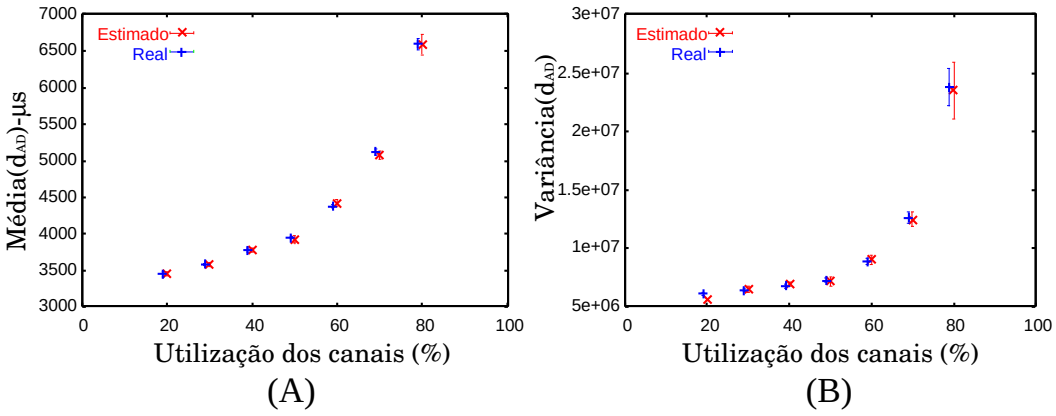


Figura 3.11: Intervalo de confiança computado para a média e variância estimada pelo algoritmo com IPID no caminho AD.

Simulações também foram executadas para analisar o modelo quando utilizado o algoritmo com *IP Spoofing*. Note que nesta versão do modelo não é mais implementado o contador global para o IPID no objeto *Host_Target*, Além disso, os objetos *Host_A* e *Host_B* podem enviar pacotes forjando o endereço IP de origem. Inicialmente, foram avaliados os resultados de simulações para cenários em que a utilização de todos os canais dos roteadores eram aproximadamente iguais, primeiro igual a 50% e depois igual a 70%. Os resultados, na forma de erro relativo, são ap-

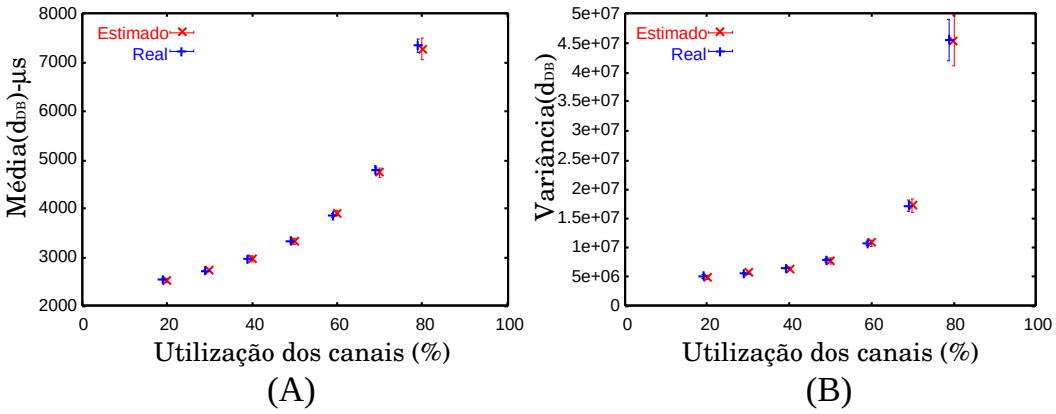


Figura 3.12: Intervalo de confiança computado para a média e variância estimada pelo algoritmo com IPID no caminho DB.

resentados nas tabelas abaixo. A Tabela 3.10 refere-se aos resultados obtidos para o sentido AD e BD, caso em que as sondas enviadas pelo *Host_A* possuem o endereço de origem forjados. Os resultados para os sentidos opostos (DA e DB), caso em que todas as sondas são enviadas do mesmo objeto *Host_B* e parte delas têm o endereço de origem do *Host_A*, estão na Tabela 3.11.

Tabela 3.10: Erro relativo computado para os caminhos AD e BD com o algoritmo *IP Spoofing*.

Caminho	Utilização dos canais 50%	Utilização dos canais 70%
	Média / variância	Média / variância
AD	0.012 / 0.002	0.029 / 0.110
BD	0.013 / 0.132	0.060 / 0.049

Tabela 3.11: Erro relativo computado para os caminhos DA e DB com o algoritmo *IP Spoofing*.

Caminho	Utilização dos canais 50%	Utilização dos canais 70%
	Média / variância	Média / variância
DA	0.022 / 0.091	0.027 / 0.168
DB	0.012 / 0.093	0.035 / 0.112

Diversas rodadas de simulação, variando a utilização de apenas dois canais do modelo, foram também executadas para analisar o intervalo de confiança das estimativas. Assim como no cenário definido para esta análise feita com o algoritmo usando IPID, nestas simulações a utilização de todos os canais foi novamente fixada em aproximadamente 50%, exceto um dos canais no caminho AD (do roteador $1B$

para o roteador $1C$) e outro canal no caminho DB (entre o roteador $4B$ a $4A$). Foram 12 rodadas para cada uma das taxas de utilização definidas, que variaram entre 20% e 80%.

As Figuras 3.13 (A) e (B) mostram os resultados obtidos para a média e variância computados para o caminho AD. É possível observar que os valores estimados são quase os mesmos que os valores reais. Nota-se também que estes resultados são muito semelhantes aos apresentados anteriormente na Figuras 3.11, obtido com o modelo considerando o algoritmo com IPID.

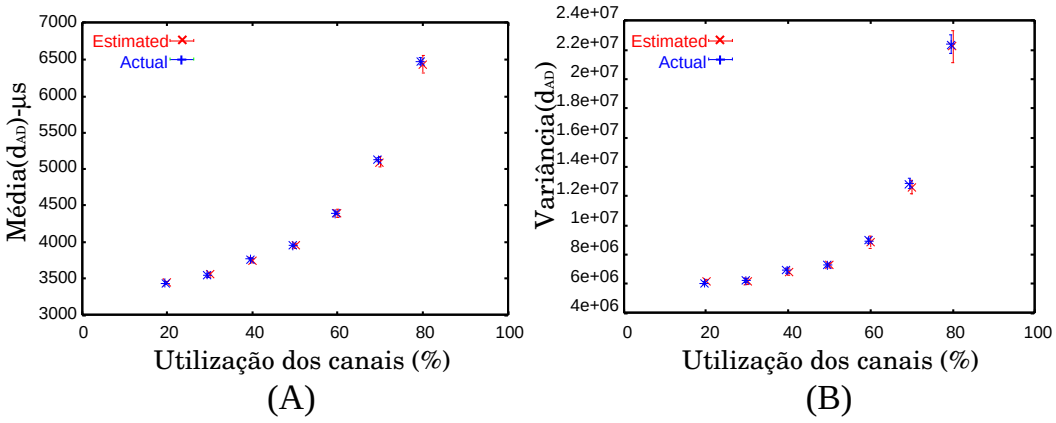


Figura 3.13: Intervalo de confiança computado para a média e variância estimada pelo algoritmo com *IP Spoofing* no caminho AD.

3.4 Análise de incerteza para a suposição da igualdade nos tempos de propagação

A técnica proposta nesta tese, para estimar a média e variância dos atrasos unidirecionais, depende fortemente da suposição de que os tempos de propagação nos caminhos de ida e de volta, entre duas máquinas quaisquer da Internet, são aproximadamente iguais. Eventuais diferenças entre os tempos de propagação em cada um dos sentidos, acarretará em erros nas estimativas finais das métricas de interesse. A incerteza sobre a veracidade desta suposição motivaram dois estudos sobre as seguintes questões fundamentais para a eficiência da técnica: (i) considerando diversas máquinas, localizadas em diferentes pontos da Internet, qual é a distribuição do erro ao compararmos os atrasos de propagação estimados pelo método descrito

na Seção 3.1.1 e os valores reais? (ii) qual o impacto que um eventual erro nesta suposição poderá ocasionar ao resultado final das estimativas obtidas pelos algoritmos desenvolvidos nesta tese? É importante ressaltar que esses estudos não servirão para fornecer uma resposta geral para essas questões, muito menos provar a validade da suposição. O objetivo é intuir sobre a validade e os possíveis impactos da incerteza desta suposição na técnica apresentada nesta seção.

Para analisar a primeira questão, o primeiro estudo trata-se de uma análise experimental em larga escala, realizada utilizando máquinas do PlanetLAB, e teve como finalidade estimar a distribuição do erro relativo existente entre os tempos de propagação estimados pelo método utilizado na técnica proposta e valores que podemos considerar muito próximos dos reais. Já na segunda questão, uma análise tratou de avaliar quantitativamente o erro causado na estimativa final, decorrente de possíveis diferenças existentes entre os valores reais do atraso de propagação e aqueles estimados pela técnica.

3.4.1 Análise experimental dos tempos de propagação

Um experimento em larga escala foi realizado na Internet, utilizando 20 (vinte) máquinas estrategicamente selecionadas do ambiente PlanetLAB. A seleção destas máquinas se deu de acordo com os seguintes critérios: (i) localização geográfica, pois foi de interesse utilizar máquinas localizadas em todos os continentes no qual o PlanetLAB se faz presente, sendo que algumas regiões contaram com mais de uma máquina neste conjunto; (ii) as máquinas deveriam estar equipadas com dispositivos específicos (como por exemplo, GPS ou CDMA) ou terem seus relógios sincronizados por intermédio de uma máquina, equipada com um desses dispositivos, que estivesse localizada na rede da mesma instituição. A seleção final das máquinas consiste de duas no Brasil, nove na América do Norte (uma no Canadá e oito nos Estados Unidos, distribuídas entre o leste, oeste e centro), duas na Ásia (uma na China e outra no Japão) e sete na Europa (uma na Espanha e duas na França, duas na Inglaterra e duas na Alemanha).

O experimento, que teve aproximadamente sete dias de duração, transcorreu da seguinte forma. A cada sete minutos, um par de máquinas (digamos, máquinas A e D) da lista acima era selecionado aleatoriamente. Durante cinco minutos, sondas

foram geradas da máquina A para a máquina D e replicadas de volta para A, de acordo com o método definido na Seção 3.1.1, com o objetivo de estimar os tempos de propagação em cada um dos sentidos (AD e DA). Ao final dos sete dias, ocorreram 1330 sessões de geração de sondas.

Para cada sessão de geração de sonda, além dos tempos de propagação estimados através do método definido neste trabalho, foram computados também os valores "reais" dos tempos de propagação em cada um dos sentidos. Uma vez que as máquinas A e D possuem seus relógios perfeitamente sincronizados e a máquina D colete os instantes de chegada das sondas, estimar o tempo de propagação é trivial. Sejam $d^{50}_{m,AD}$ e $d^{500}_{m,AD}$ os menores valores de atraso unidirecional computados durante uma sessão de medição para sondas de tamanho 50 e 500 *bytes*, respectivamente. O valor "real" do tempo de propagação entre as máquinas A e D ($\widehat{T}^{prop}_{AD}$) pode ser calculado utilizando as equações 3.9.

$$\begin{cases} d^{50}_{m,AD} = T^{tx}_{AD} + \widehat{T}^{prop}_{AD} + T^{fila}_{AD} \\ d^{500}_{m,AD} = T^{tx}_{AD} + \widehat{T}^{prop}_{AD} + T^{fila}_{AD} \end{cases} \tag{3.9}$$

Note que, para estimar os tempos de propagação (T^{prop}_{AD}) através da técnica proposta, não é necessário que as máquinas A e D estejam sincronizadas, uma vez que informações sobre o instante de tempo do relógio em D não são utilizadas no cálculo. A sincronização dos relógios só se faz necessária para que seja estimado o valor de $\widehat{T}^{prop}_{AD}$.

Para todas as sessões do experimento, são computados os erros relativos das estimativas obtidas pelo método proposto (T^{prop}_{AD}) em relação aos valores "reais" do tempo de propagação ($\widehat{T}^{prop}_{AD}$). A distribuição de probabilidade do erro relativo está ilustrada no gráfico da Figura 3.14. Os resultados demonstram que aproximadamente 75% das estimativas obtidas nos experimentos para T^{prop}_{AD} tiveram um erro relativo menor que 5% em relação aos valores "reais" de $\widehat{T}^{prop}_{AD}$. Se considerarmos um erro relativo de 10%, o número de ocorrências com resultados menores ou iguais a esse valor é então superior a 93%. Esses resultados reforçam a validade da suposição usada pela técnica de que os tempos de propagação nos dois sentidos podem ser considerados aproximadamente iguais.

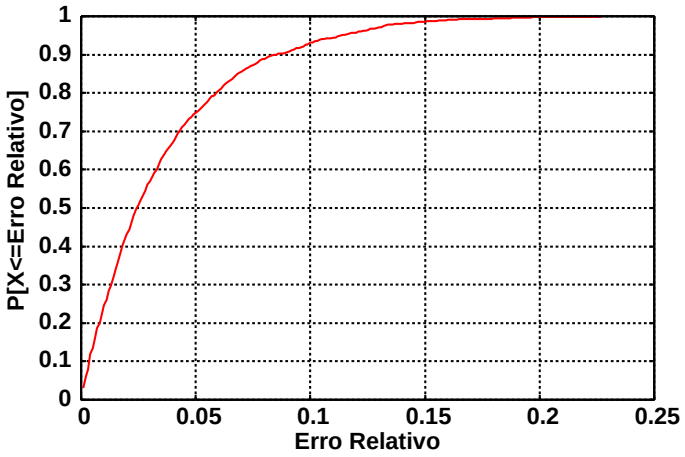


Figura 3.14: Distribuição do erro relativo computado entre os valores estimados pela técnica e os valores "reais".

3.4.2 Análise quantitativa do erro nas estimativas do atraso

Os resultados apresentados na subseção anterior reforçam a suposição de igualdade nos tempos de propagação, ao menos para uma grande parcela dos caminhos experimentados na Internet. Ainda assim, uma análise quantitativa ainda foi feita com o objetivo de avaliar o erro causado às estimativas finais no caso de violação da hipótese que os tempos de propagação são iguais ($T_{prop}^{ida} = T_{prop}^{volta}$).

Valores obtidos de experimentos reais, apresentados na Seção 3.3.1, foram utilizados na análise. O estudo se deu utilizando amostras de experimentos, no qual foi possível estimar os valores de atraso (vamos supor, por exemplo, d_{AD} e d_{DA}) através da suposição de igualdade nos tempos de propagação (neste caso, $T_{prop}^{AD} = T_{prop}^{DA}$ e $T_{prop}^{BD} = T_{prop}^{DB}$), e consistiu da comparação desses valores estimados (d_{AD} e d_{DA}) com os valores obtidos de atraso (e.g., $\widehat{d}_{AD}$ e $\widehat{d}_{DA}$) assumindo outros valores para os tempos de propagação (onde, $\widehat{T}_{AD}^{prop} \neq \widehat{T}_{DA}^{prop}$ e $\widehat{T}_{BD}^{prop} \neq \widehat{T}_{DB}^{prop}$).

Sejam e_{AD} e e_{BD} as diferenças obtidas, respectivamente, por $T_{AD}^{prop} - \widehat{T}_{AD}^{prop}$ e $T_{BD}^{prop} - \widehat{T}_{BD}^{prop}$. Portanto, $\widehat{T}_{AD}^{prop} = T_{AD}^{prop} + e_{AD}$, $\widehat{T}_{DA}^{prop} = T_{DA}^{prop} - e_{AD}$, $\widehat{T}_{BD}^{prop} = T_{BD}^{prop} + e_{BD}$ e $\widehat{T}_{DB}^{prop} = T_{DB}^{prop} - e_{DB}$. Assim, para cada valor considerado de e_{AD} e e_{BD}, novas estimativas foram obtidas de $\widehat{d}_{AD}$ e $\widehat{d}_{DA}$).

A análise desenvolvida comparou os atrasos estimados utilizando a suposição de igualdade de propagação (d_{AD} e d_{DA}) e os valores de atraso estimados para o caso de desigualdade entre os tempos de propagação ($\widehat{d}_{AD}$ e $\widehat{d}_{DA}$). A comparação foi feita

através do cálculo da diferença relativa, que é dada por:

$$\frac{|d_{sentido} - \widehat{d}_{sentido}|}{d_{sentido}}$$

onde, "sentido" representa o caminho AD ou DA.

Inicialmente, foi analisado o caso em que apenas $e_{AD} \neq 0$. Isto é, quando os tempos de propagação são iguais nos sentidos BD e DB ($T_{BD}^{prop} = T_{DB}^{prop}$), mas diferentes nos sentidos AD e DA ($T_{AD}^{prop} \neq T_{AD}^{prop}$). Diversos valores foram considerados para e_{AD}. A diferença de d_{AD} e de d_{DA} em relação aos diversos $\widehat{d}_{AD}$ e $\widehat{d}_{DA}$ computados para cada um dos valores considerados de e_{AD}, nesta primeira análise, estão descritas na Tabela 3.12. Pelos resultados, é possível notar que o crescimento do erro introduzido às estimativas finais é inferior ao crescimento dos valores de e_{AD}.

Tabela 3.12: Resultados das estimativas do atraso (em μs) para os sentidos AD e DA com diferentes valores de e_{AD}.

Sentido	$[e_{AD} = 0]$ / $d_{sentido}$	$[e_{AD} = 0.01]$ / $\widehat{d}_{sentido}$/Dif. Rel.	$[e_{AD} = 0.02]$ / $\widehat{d}_{sentido}$/Dif. Rel.	$[e_{AD} = 0.05]$ / $\widehat{d}_{sentido}$/Dif. Rel.	$[e_{AD} = 0.1]$ / $\widehat{d}_{sentido}$/Dif. Rel.
AD	99000	98472 / 0.005	97944 / 0.011	96361 / 0.027	93722 / 0.053
DA	75.000	75528 / 0.007	76056 / 0.014	77638 / 0.035	80278 / 0.070

Por fim, a análise foi feita para o caso da ocorrência de erros nos cálculos do tempo de propagação não só no sentido AD, mas também no sentido BD. A diferença relativa foi feita para comparar os valores estimados quando e_{DA} e e_{DB} são iguais a zero e os valores e_{AD} e e_{BD} variam entre -0.20 e 0.20. Os gráficos apresentados pelas Figuras 3.15 e 3.16 ilustram os resultados obtidos. Nos gráficos é possível notar que a diferença relativa computada é sempre inferior aos valores assumidos para e_{AD} e e_{BD}.

3.5 Conclusão

Neste capítulo foi apresentada uma nova técnica para estimar a média e a variância do atraso em um único sentido. A proposta trata-se de um método não cooperativo de medições ativas, pois descarta a necessidade de permissões de acesso à máquina remota para executar qualquer processo de coleta de sondas. Para contornar a falta de acesso à máquina alvo, foram desenvolvidas duas variações para

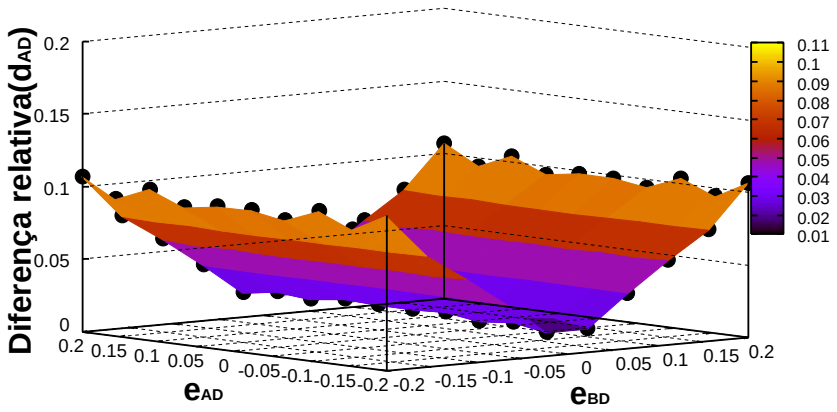


Figura 3.15: Resultados das estimativas do atraso para o sentido AD com diferentes valores de e_{AD} e e_{BD}.

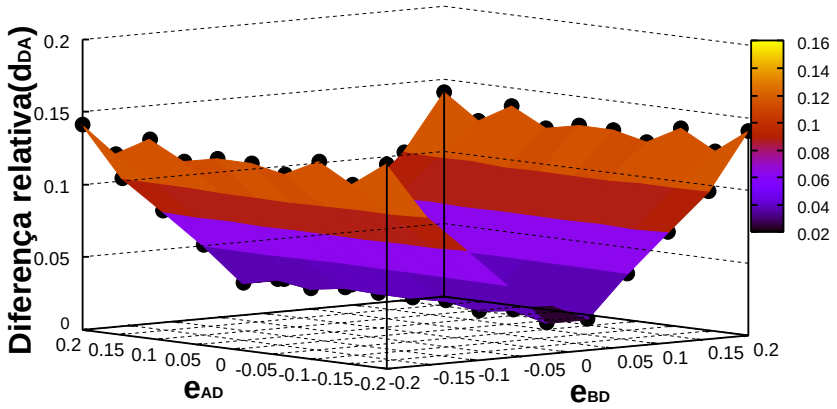


Figura 3.16: Resultados das estimativas do atraso para o sentido DA com diferentes valores de e_{AD} e e_{BD}.

a técnica: (i) a primeira faz uso do campo IPID dos pacotes replicados por esta máquina alvo e supõe que ela possui um sistema operacional que implemente um contador global para os pacotes enviados; (ii) envia pacotes com endereços de origem forjados fazendo *IP spoofing* nos pacotes enviados à máquina remota. Além disso, a técnica pode ser utilizada mesmo se os relógios das máquinas fonte das sondas não estejam sincronizados. Portanto, é uma ferramenta valiosa para medições de atraso unidirecional na Internet independentemente de se ter acesso e permissão de coleta na máquina remota alvo.

Diversos experimentos usando máquinas conectadas à Internet (algumas delas do ambiente PlanetLab) foram realizados. Os resultados obtidos nesses experimentos demostram a precisão da técnica, tanto utilizando o algoritmo com IPID quando o algoritmo com *IP Spoofing*. Experimentos foram também utilizados para validar

a extensão da técnica que incluía o tratamento de problemas como *Skew* e *Offset*, quando os relógios das máquinas fonte não encontram-se sincronizados.

Também foi avaliada a eficácia da técnica através de simulação, para vários cenários utilizando modelos desenvolvidos no ambiente de simulação da ferramenta TANGRAM-II. O objetivo principal foi analisar os algoritmos quando as medições eram aplicadas sobre diferentes valores de utilização da largura de banda. Os resultados confirmam a eficácia do desempenho da técnica, para diferentes cargas nos canais da rede.

Ao final do capítulo, foram apresentados ainda resultados de experimentos reais com o objetivo de avaliar o impacto da incerteza sobre a suposição de igualdade nos tempos de propagação dos caminhos de ida e volta na rede. Os resultados experimentais demonstraram que, ao menos entre as máquinas selecionadas do PlanetLAB para os experimentos, os caminhos medidos na Internet possuem tempos de propagação aproximadamente iguais nos dois sentidos. Já a análise quantitativa demonstrou que o erro na estimativa final da métrica possui um crescimento inferior quando comparado ao erro decorrente de uma eventual diferença nos tempos de propagação dos caminhos em cada um dos sentidos.

Capítulo 4

Uma técnica de medição fim-a-fim para estimar a taxa de transmissão em uma rede local sem fio

NESSE capítulo, é apresentada a descrição de uma técnica de medição fim-a-fim desenvolvida para estimar a taxa de transmissão (capacidade em *bits* por segundo) de um dispositivo conectado à Internet por meio de uma rede de acesso sem fio IEEE 802.11a/b/g. Para contextualizar a técnica proposta neste capítulo, que é descrita na Seção 4.3, inicialmente é apresentado na Seção 4.1 uma introdução sobre redes de acesso e feita uma revisão do padrão IEEE 802.11 na Seção 4.2. Validações da acurácia do método proposto, obtidas através de simulações e de experimentos em ambientes reais, são também apresentadas ao final deste capítulo (Seção 4.4).

4.1 Redes de acesso

Uma rede de acesso consiste na conexão entre os sistemas finais e os roteadores de borda da Internet. Os tipos de conexão, utilizados pelas redes de acesso na Internet, podem ser classificados como de baixa velocidade (*por exemplo, Dial-up*) e alta velocidade (conexões em banda larga). As conexões *dial-up* são feitas através das linhas telefônicas e estão limitadas a taxas inferiores a 56 *kbps*. Já as conexões de banda larga alcançam taxas superiores a 64 *kbps*. Dentre os serviços de acesso de banda larga mais utilizados atualmente estão as conexões do tipo *ADSL*, *Cable*

Modem, *Ethernet* e *WLAN*. Recentemente, algumas outras tecnologias para rede de acesso, como os padrões 3G (e.g., EVDO-UMTS e HSDPA/HSUPA-CDMA2000) e o WiMAX (também chamados de IEEE 802.16), vêm ganhando destaque, mas ainda são bem menos utilizados. Esses vários tipos de acesso diferem radicalmente em algumas de suas características, tais como capacidade de transmissão, vazão máxima e meio físico de propagação.

Uma técnica de medição fim-a-fim, que permita identificar o tipo de conexão existente no último salto de um caminho na Internet, pode ser útil para alguns serviços da Internet. Trabalhos existentes na literatura propõem, por exemplo, novas versões para o TCP que têm como objetivo aumentar o desempenho do protocolo quando o último salto é uma *WLAN* [120, 121] ou um *Cable Modem* [122]. Essas propostas pressupõem o conhecimento prévio do tipo de conexão, mas em nenhuma delas é definida como deve ser feita a identificação da rede de acesso no último salto.

4.1.1 Inferências sobre as redes de acesso

Em [123, 124, 125] são apresentadas propostas para identificar o tipo de acesso utilizado por uma máquina remota em sua conexão com a Internet. O trabalho desenvolvido por Cheng e Marsic, em [123], foi a primeira técnica desenvolvida para identificar se um fluxo de dados é oriundo de uma conexão sem fio ou de uma conexão cabeada. A inferência é feita a partir dos valores computados para os atrasos de ida-e-volta dos pacotes de conexões TCP observadas. Nos trabalhos apresentados em [124, 125], os autores definem técnicas para classificar a conexão de acesso do último salto entre Ethernet, WLAN, ADSL, Cable Modem ou Dial-up. A diferença entre os trabalhos [124] e [125] é que o primeiro trata-se de uma técnica ativa de medição, enquanto que o segundo consiste de uma versão passiva para a técnica. As duas técnicas são baseadas nos cálculos da mediana e da entropia do intervalo entre chegadas de pares de pacotes, para inferir o tipo de rede de acesso.

Dentre os diferentes tipos de acesso à Internet, as redes locais sem fio têm se tornado, sem dúvida, uma das formas mais populares. As altas taxas de transmissão alcançadas pelos padrões IEEE 802.11a/b/g e a significativa redução nos custos dos equipamentos são alguns dos fatores que justificam o crescimento da utilização desta tecnologia. Locais públicos como aeroportos, bibliotecas, campi universitários,

cafés, shoppings, além de residências e escritórios particulares, são apenas alguns dos ambientes nos quais as redes sem fio têm sido largamente oferecidas como serviço de acesso à Internet. Uma característica inerente às conexões 802.11a/b/g é que a taxa de transmissão adotada pelo dispositivo pode variar, a depender das condições do meio como o nível sinal/ruído ou ocorrências de colisão. As taxas de transmissão podem variar de valores relativamente altos, que chegam a 54Mbps, até valores significativamente baixos, como 1 ou 2Mbps.

Quando o último salto for classificado como uma conexão 802.11 por qualquer um dos métodos de [123, 124, 125], estimar também a taxa de transmissão do dispositivo sem fio torna-se importante para diversas aplicações. Em serviços de mídia contínua, por exemplo, a estimativa desta taxa pode ser utilizada para auxiliar no melhor ajuste da taxa de transmissão do servidor para o cliente multimídia. Servidores multimídia como o *Windows Streaming Media* utilizam o método de pares de pacotes para estimar a capacidade de contenção ao longo do caminho do servidor para o cliente [126]. No entanto, resultados apresentados em [127] demonstram que as estimativas obtidas por estas aplicações são imprecisas quando os clientes encontram-se conectados por uma rede local sem fio. Para aplicações P2P, o critério para a escolha dos vizinhos pode levar em consideração também as capacidades de transmissão dos clientes, ao invés de apenas o tipo de acesso como sugerido em [124]. Para os serviços de inferência da topologia física, o conhecimento da capacidade de transmissão pode ser de grande utilidade para o gerenciamento de recursos [128]. Por fim, os trabalhos com propostas de novas versões do TCP podem explorar o conhecimento da capacidade de transmissão da máquina na rede sem fio, para aumentar o desempenho do TCP [121, 120].

A técnica apresentada como contribuição desta seção da tese é complementar aos trabalhos apresentados em [123, 124, 125]. O método proposto, para inferir a taxa de transmissão (em bits por segundo) do último salto, assume que um dos mecanismos existentes já identificou a rede de acesso como sendo uma conexão IEEE 802.11, ou que simplesmente essa informação é conhecida. Antes de descrever a técnica se faz necessária uma revisão sobre o padrão 802.11.

Padrão	Limites de Frequência	Taxa de Transmissão de Dados
802.11b	2.4GHz - 2.485 GHz	até 11Mbps
802.11a	5.1GHz - 5.8 GHz	até 54Mbps
802.11g	2.4GHz - 2.485 GHz	até 54Mbps

Tabela 4.1: Faixas de frequência e taxas de transmissão dos padrões IEEE 802.11.

4.2 Revisão do padrão 802.11

O padrão IEEE 802.11, descrito em [39], assim como as versões mais recentes que contemplam taxas de transmissão maiores, descritas em [40], definem a camada física e o controle de acesso ao meio *(Medium Access Network - MAC)* para as redes locais sem fio. Diferentes modelos para a camada física das redes 802.11 foram definidos, incluindo os padrões 802.11a, 802.11b e 802.11g. Cada um destes padrões opera sobre uma faixa de frequência e com velocidades específicas, como mostra a Tabela 4.1. O padrão 802.11a opera na banda de frequência de 5 GHz, o que o torna incompatível com os padrões 802.11b e 802.11g. Atualmente, a maioria dos projetos e equipamentos para redes locais sem fio utiliza a tecnologia 802.11b ou 802.11g. A motivação para a utilização dos padrões "b" e "g" é a compatibilidade entre os equipamentos destes dois padrões, além da falta de regulamentação que ainda existe em muitos países para a utilização do espaço de frequência de 5 GHz.

As áreas de cobertura de uma rede local 802.11 são denominadas áreas básicas de serviço *(Basic Service Area - BSA)*. Um grupo de terminais sem fio 802.11, operando em uma mesma *BSA*, define um conjunto básico de serviço *(Basic Service Service - BSS)*. A rede formada pelos terminais sem fio em uma BSS pode estar operando no modo *Ad Hoc* ou com infraestrutura.

No modo *Ad Hoc*, teoricamente, qualquer terminal está apto a estabelecer uma comunicação direta com qualquer outra estação da mesma *BSS*. Para a operação da rede neste modo, não há necessidade de um ponto centralizado de controle. No entanto, degradações no meio de transmissão devido ao enfraquecimento do sinal ou à interferência podem fazer com que o sinal transmitido por algum terminal não seja detectado por algumas estações da mesma *BSS*, causando o problema do terminal oculto.

Na operação em modo infra-estruturado, a *BSS* é formada por terminais sem fio e por um ponto centralizado de controle, chamado de ponto de acesso (*Access Point - AP*). Todos os pacotes endereçados a um dos terminais da *WLAN* deverão ser encaminhados ao *AP* que se encarregará de transmití-los ao terminal de destino. De forma análoga, todo pacote enviado por um terminal sem fio será enviado primeiro ao *AP*, e este o encaminhará à estação de destino dentro da *BSS*, ou em algum ponto na Internet.

Independente dos padrões (802.11 a, b ou g) que definem faixa de frequência e taxa de transmissão distintas, e independente do modo de operação (*Ad Hoc* ou infra-estruturado), a mesma estrutura de acesso ao meio é utilizada. Na subcamada *MAC*, o padrão 802.11 prevê dois métodos de acesso ao meio: (i) o método com uma função de coordenação centralizada *(Point Coordination Function - PCF)*, em que um esquema de controle centralizado de acesso ao meio é implementado e esta unidade central coordena a disputa pelo direito de transmissão no meio; (ii) O método com uma função de coordenação distribuída *(Distributed Coordination Function - DCF)*, que é baseado no *CSMA/CA (Carrier Sense Multiple Access / Collision Avoidance)*, onde não existe a figura de um coordenador central do canal e todos os terminais disputam entre si o acesso ao meio para obter o direito de transmissão dos pacotes. Embora o modo *PCF* seja apropriado para a transmissão de tráfego de tempo real e possa coexistir com o método *DCF*, o método *PCF* raramente é implementado pelos fabricantes dos produtos 802.11 e, em geral, este método não é utilizado atualmente nas *WLAN's*. Por isso, nesta tese é considerado apenas o método *DCF*.

No método de acesso *DCF* são ainda definidos o mecanismo básico de acesso e o mecanismo opcional com reserva de acesso ao canal com quadros de controle *Request-To-Send/Clear-To-Send (RTS/CTS)*. No segundo mecanismo, antes de transmitir efetivamente os dados pelo canal, o terminal sem fio deve enviar um quadro de *RTS* ao receptor e aguardar que o receptor envie de volta o quadro de *CTS*. O uso de quadros *RTS/CTS* tem como objetivo ajudar a reduzir o problema causado pelas colisões em redes com altas cargas e o problema do terminal oculto. Porém, este mecanismo adiciona um *overhead* significativo na rede e raramente é adotado nas *WLAN's* com sobrecarga moderada e em transmissões de pacotes pequenos. Na tese

é assumida sempre a utilização do método de acesso básico.

A Figura 4.1 ilustra uma transmissão utilizando o método *DCF* básico. Antes de iniciar a transmissão, o terminal monitora o meio para verificar se outra estação está transmitindo. Se o meio ficar ocioso por um período igual a *DIFS (Distributed Interframe Space)*, o terminal efetuará a transmissão. Porém, se alguma transmissão for detectada no período de *DIFS*, o terminal deverá adiar a sua transmissão. O terminal continua a monitorar o meio e assim que perceber que o canal está ocioso por um período igual a *DIFS* será realizado o procedimento de *backoff* exponencial binário. Um intervalo aleatório, chamado de intervalo de *backoff*, é selecionado. Esse intervalo equivale a um valor uniformemente distribuído entre (0, CW-1) [1] vezes um *slot* de transmissão [2]. Um temporizador é iniciado com o valor do intervalo de *backoff*. O temporizador é decrementado sempre que o meio estiver ocioso e não muda de valor, enquanto uma transmissão for detectada pela estação. voltando a ser decrementado quando o meio voltar a ficar ocioso por um período igual a *DIFS*. Assim que o temporizador expirar, o pacote será transmitido pelo terminal. Se o pacote for recebido corretamente, a estação receptora se encarregará de enviar um *ACK* após um período igual a *SIFS (Short Interframe Space)*. Caso o *ACK* não seja recebido pela estação transmissora, o pacote original será escalonado para retransmissão.

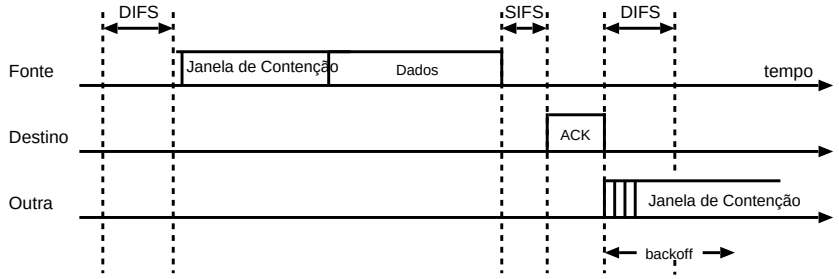


Figura 4.1: Transmissão em uma rede local 802.11 utilizando o método DCF básico.

O padrão IEEE 802.11 prevê ainda um ajuste automático da taxa de transmissão a depender das condições encontradas no meio. Em condições significativas de interferência ou colisão no canal, a estação pode ajustar automaticamente sua taxa de

[1] CW é o tamanho da janela de contenção que, inicialmente, possui tamanho 32, mas aumenta exponencialmente a cada tentativa de transmissão ocorrida sem sucesso.

[2] Um *slot* de transmissão corresponde ao tempo de ida e volta do sinal dentro de uma *BSS*.

Padrão	Taxas de Transmissão
802.11a	54, 48, 36, 24, 18, 12, 9 e 6Mbps
802.11b	11, 5.5, 2 e 1 Mbps
802.11g	54, 48, 36, 24, 18, 12, 9 e 6Mbps
802.11g + legado	11, 5.5, 2 e 1 Mbps

Tabela 4.2: Taxas de transmissão suportadas por cada um dos padrões.

transmissão para obter um melhor desempenho na rede. No entanto, o 802.11 não define um algoritmo padrão para o ajuste da taxa e, então, fica a cargo do fabricante implementar o algoritmo que mais lhe interesse. Os algoritmos de seleção de taxa atualmente implementados são classificados de acordo com a informação utilizada para a tomada de decisão. Normalmente, os algoritmos tomam as decisões baseados em estatísticas obtidas do histórico de envios de pacotes ou na relação sinal ruído. As taxas de transmissão suportadas por cada um dos padrões estão descritas na Tabela 4.2. Note que, para manter o legado do padrão 802.11b, o 802.11g suporta ainda as taxas do padrão anterior (11, 5.5, 2, e 1 Mbps).

4.3 Estimando a taxa de transmissão de um enlace de acesso sem fio

Dois aspectos fundamentais devem ser considerados para estimar a taxa de transmissão de um enlace sem fio localizado no último salto em um caminho de rede: (i) o *overhead* do protocolo 802.11; e, (ii) a possibilidade da conexão sem fio não ser o enlace de contenção (de menor capacidade) ao longo do caminho de rede.

Para exemplificar a questão do *overhead*, a Figura 4.2 ilustra a transmissão de dois pacotes consecutivos (ou seja, um par de pacotes) em um enlace 802.11. No exemplo, é pressuposto um cenário ideal, no qual não há tráfego concorrente durante a transmissão dos pacotes. Conforme mostra a figura, o intervalo entre as chegadas dos pacotes do par ao receptor será igual à soma dos seguintes tempos: *SIFS*, transmissão do *ACK, DIFS, backoff* e transmissão do segundo pacote do par. Claramente, nesse caso, a capacidade de transmissão não pode ser obtida através

da equação $C = B/T$ (onde, B é o tamanho dos pacotes e T o intervalo entre a chegada do par), como é usado pela técnica original de pares de pacotes, mencionada na Seção 2.1.4 desta tese.

Os tempos de *SIFS, DIFS* e transmissão do *ACK* são constantes. A equação, mencionada acima para estimar a capacidade, poderia ser facilmente adaptada para considerar esses valores. No entanto, o tempo de *backoff* é uma variável aleatória dependente de alguns fatores, como por exemplo a carga da rede, e não é trivial de adaptá-la à equação. Mesmo na ausência de tráfego concorrente, o tempo de *backoff* entre a transmissão de dois pacotes consecutivos de uma mesma máquina tem alta variabilidade. Por isso, não é possível garantir que o segundo pacote do par será transmitido em um intervalo curto de tempo, imediatamente após a transmissão do primeiro. De acordo com a revisão do padrão IEEE 802.11, apresentada na Seção 4.2, o menor valor da janela de contenção ($CW_{\min}$) é 32 e o tempo de *backoff* é determinado por uma variável aleatória uniforme entre $[0, CW_{\min} - 1]$. Esse tempo de *backoff* é decrementado sempre que o canal estiver livre.

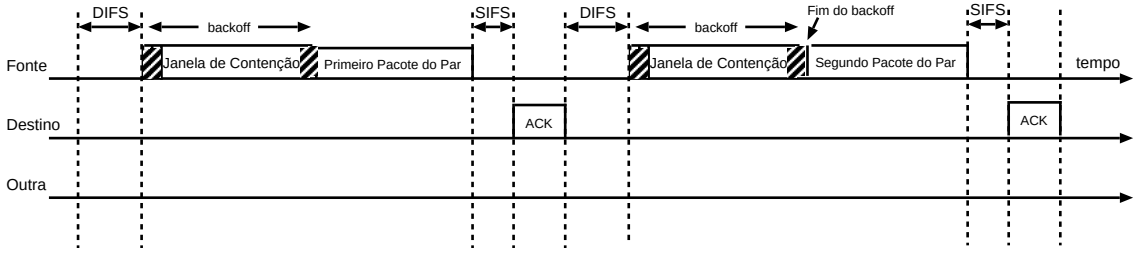


Figura 4.2: Transmissão de um par de pacotes em uma rede local 802.11 utilizando o método DCF básico.

O outro aspecto a ser considerado é que nem sempre o salto sem fio é o de menor capacidade de transmissão no caminho da rede. Com as altas taxas de transmissão alcançadas atualmente pelos dispositivos 802.11, não é incomum que o salto sem fio possua uma capacidade de transmissão superior à capacidade de alguns enlaces cabeados ao longo de um caminho. Esse aspecto não é de grande relevância, quando o objetivo é a estimativa da métrica capacidade de contenção, mas não pode ser descartado quando se deseja medir a taxa de transmissão do enlace sem fio localizado na rede de acesso.

4.3.1 Descrição da técnica proposta

A técnica proposta é uma variação do método tradicional de pares de pacotes para geração de sondas, com um filtro de seleção dos pares. Uma equação também é definida para auxiliar na estimativa da capacidade de uma conexão através de uma rede de acesso 802.11. O método desenvolvido considera aspectos fundamentais como a possibilidade de existirem enlaces de menor capacidade ao longo do caminho, a ocorrência de tráfego concorrente, e o *overhead* do protocolo 802.11.

Para inferir a taxa de transmissão do enlace sem fio que conecta o computador B à Internet, por exemplo, uma sequência de m grupos de sondas são enviadas de uma máquina fonte (A) para a máquina alvo (B). Cada um dos m grupos de sondas é formado por quatro pares de pacotes, como ilustra a Figura 4.3. As sondas de uma sequência podem ser representadas por $\psi_{i,j}^{k}$, onde o índice k ($k = 1, ..., m$) identifica um dos m grupos, j ($j = 1, 2, 3, 4$) indica o índice de um par, em particular, do grupo e i ($i = 1, 2$) indica o primeiro ($i = 1$) ou o segundo ($i = 2$) pacote de determinado par.

Ao contrário do método tradicional de pares de pacotes, onde os dois pacotes de um par de sondas possuem o mesmo tamanho, no método utilizado neste trabalho são atribuídos tamanhos distintos entre o primeiro ($P1$) e o segundo ($P2$) pacote de cada par. Seja $L_{i,j}^{k}$ o tamanho (em *bytes*) do pacote $\psi_{i,j}^{k}$. O tamanho da primeira sonda de todos os pares, denotada por $L_{1,j}^{k}$ (para qualquer j e k), é igual à unidade máxima de transmissão (MTU), definida para as redes Ethernet (1500 bytes). Esse valor é definido para evitar fragmentação do datagrama IP. Em todos os m grupos, os seguintes tamanhos são atribuídos aos pacotes $P2$ de cada um dos quatro pares: $L_{2,1}^{k} = 600$, $L_{2,2}^{k} = 800$, $L_{2,3}^{k} = 1000$ e $L_{2,4}^{k} = 1200$ bytes. O tamanho do segundo pacote de cada um dos quatro pares de um grupo assumem um dos quatro valores acima especificados (600, 800, 1000, 1200 *bytes*). Note que todos são menores do que o primeiro pacote do par.

A racionalidade por trás do uso de pares de pacotes com tamanhos distintos ($L_{1,j}^{k} > L_{2,j}^{k}$) é aumentar as chances da segunda sonda chegar ao ponto de acesso da rede sem fio, antes ou imediatamente após o envio da primeira sonda. Assumindo que os dois pacotes percorrem um mesmo caminho de rede, formado por n saltos, cujas capacidades dos enlaces são dadas por C_l (para $l = 1, ..., n$), o tempo de

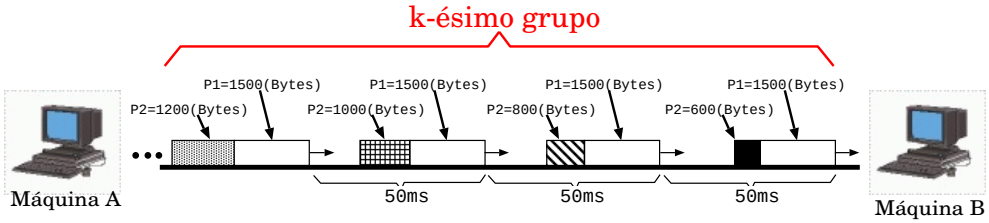


Figura 4.3: Conjunto de pares de pacotes utilizado na técnica proposta.

transmissão do segundo pacote, em todos os n enlaces, será inferior ao tempo de transmissão do primeiro: $L^k_{1,j}/C_l > L^k_{2,j}/C_l$, para todo $l = 1, ..., n$. A diferença do tamanho do primeiro pacote, em relação ao segundo, é uma forma de reduzir a dispersão do par, eventualmente imposta pelos enlaces ao longo do percurso, até sua chegada ao ponto de acesso da rede sem fio. Considerando a inexistência de tráfego concorrente, em todos os saltos do caminho entre A e B, a transmissão de $P2$ terá início imediatamente após a transmissão de $P1$.

Utilizar sondas de diferentes tamanhos é, sem dúvida, uma solução eficiente para reduzir a dispersão entre as chegadas de $P1$ e $P2$ ao ponto de acesso. No entanto, não se trata de uma solução suficiente para garantir a chegada consecutiva das sondas, uma vez que a distribuição das capacidades dos enlaces ao longo do caminho e o tráfego concorrente podem atrapalhar a chegada consecutiva das sondas de um par.

A distribuição das capacidades de transmissão dos enlaces do caminho pode ser determinante para a chegada consecutiva dos pares de pacotes à rede de acesso. Isso porque, se o tempo de transmissão de $P2$, em um determinado enlace do caminho (por exemplo, enlace l), for superior ao tempo de transmissão do primeiro pacote no enlace seguinte (e.g., enlace $l+1$), $P2$ chegará ao enlace $l+1$ após $P1$ já ter sido transmitido ($L^k_{1,j}/C_{l+1} < L^k_{2,j}/C_l$). Isso, obviamente, desconsiderando a existência de filas nos enlaces. Porém, considerando que o caminho percorrido pelos pares de pacotes, entre as máquinas A e B, passa por roteadores de núcleo e de borda da Internet, não é incorreto acreditar que à medida em que o par se aproxima do núcleo da Internet, $P1$ e $P2$ encontram enlaces de maior capacidade de transmissão e, ocasionalmente, podem se dispersar um do outro. Em contrapartida, à medida que eles voltam a se aproximar da borda, passando por roteadores de menor capacidade, a tendência é que, se eventualmente houver alguma dispersão, essa volte a reduzir.

O tráfego concorrente também pode atrapalhar a chegada consecutiva do par de

sondas ao ponto de acesso. A presença de pacotes entre $P1$ e $P2$ pode ocasionar um aumento na dispersão do par. No entanto, observe que, se o objetivo é que as sondas $P1$ e $P2$ cheguem juntas para transmissão do último salto, da mesma forma que o tráfego concorrente pode interferir na chegada consecutiva, ele poderá também resultar em uma redução da dispersão existente entre $P1$ e $P2$, antes de chegar ao ponto de acesso. Isso porque, se o tráfego concorrente for inserido à frente do primeiro pacote de um par, eventuais filas nos roteadores podem retardar a progressão da primeira sonda, ocasionando uma redução da dispersão entre $P1$ e $P2$.

Considere um cenário em que o caminho entre as máquinas A e B possui quatro saltos, sendo o último desses saltos uma conexão de rede local IEEE 802.11g. Esse cenário é idêntico a um dos experimentos apresentados na seção de validação (4.4) e é utilizado aqui para enfatizar alguns dos principais aspectos do algoritmo. Em um dos experimentos executados neste cenário, dez grupos de pares de pacotes foram gerados (num total de 40 pares de sondas). Uma sequência, denotada por δ_j^k, é formada a partir das dispersões computadas pela máquina B, ao receber cada um dos quatro pares dos k grupos.

A Figura 4.4(A) ilustra os valores das dispersões computadas para cada um dos pares de sondas na sequência δ_j^k. No gráfico (B), da Figura 4.4, os mesmos valores são mostrados, mas agora as amostras estão organizadas em função do tamanho do segundo pacote de cada par (definido pelo índice j, na sequência δ_j^k). Pelos gráficos é possível notar a alta variabilidade dos valores de dispersão computados para os pares de sondas. Mesmo entre os pares de mesmo tamanho de $P2$ (como mostra a Figura 4.4(B)), existe uma variação considerável entre as dispersões computadas.

A alta variabilidade, vista nos gráficos da Figura 4.4, é basicamente causada pela ocorrência de um (ou alguns) dos seguintes fatores: (i) tráfego concorrente; (ii) capacidade de transmissão dos enlaces do caminho; e (iii) tempo de *backoff* do padrão 802.11. Para reduzir os efeitos desses fatores, o método proposto prevê uma seleção dos pares que serão usados para o cálculo da taxa de transmissão. Para cada índice j, será selecionada a amostra cujo o par possui o menor valor de dispersão:

$$\delta_j^{min} = \min_{\forall k}\{\delta_j^k\}, \text{ para } j = 1, 2, 3, 4 \tag{4.1}$$

A nova sequência δ_j^{min} é formada pelas quatro amostras selecionadas, uma para

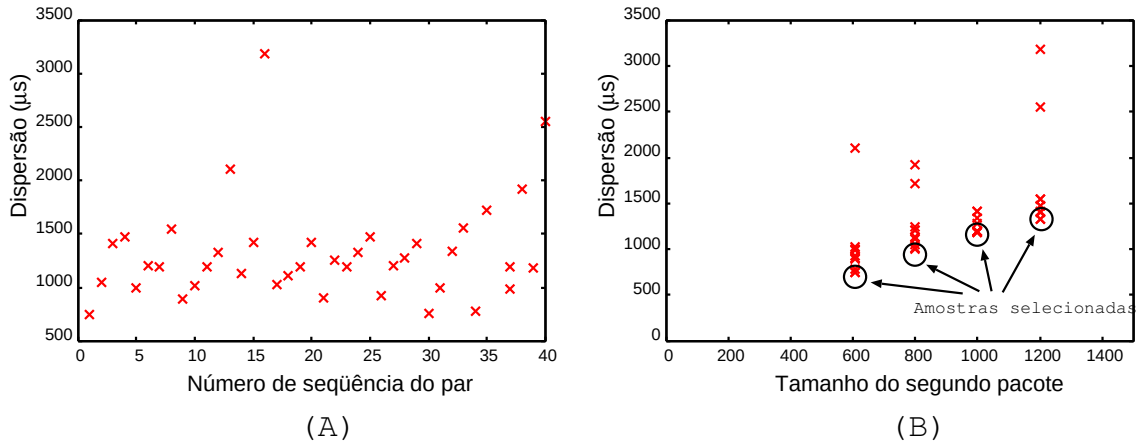


Figura 4.4: Dispersões computadas para a geração de pares de pacotes com o método proposto.

cada tamanho de segundo pacote. No experimento mostrado pela Figura 4.4(B), as amostras que formam a sequência δ_r^{min} estão identificadas no gráfico.

Em um cenário ideal, esses pares selecionados, supostamente, foram enviados um logo após o outro no enlace sem fio; não sofreram nenhuma (ou muito pouca) influência do tráfego concorrente, durante a transmissão no último salto; e, não tiveram valores significativos de *backoff*, entre os envios de $P1$ e $P2$. Nesse cenário idealizado, a dispersão entre um par de pacotes é dada pela soma dos tempos de *SIFS*, transmissão do *ACK*, *DIFS* e transmissão do segundo pacote, onde os tempos de *SIFS* e *DIFS* são constantes e os tempos de transmissão do *ACK* e do segundo pacote possuem uma relação linear entre os seus tamanhos, em *bytes*, e a taxa de transmissão do enlace sem fio. Assim, considerando a possibilidade de valores nulos de *backoff* e a inexistência de tráfego concorrente entre os pares, o limite inferior para a dispersão computada na recepção dos pacotes para uma taxa de transmissão do enlace sem fio igual a C_w é dado por:

$$D_{j,C_w} = t_{SIFS} + t_{DIFS} + L_{ACK}/C_w + L_{2,j}^{min}/C_w. \tag{4.2}$$

onde, t_{SIFS} e t_{DIFS} são os intervalos de tempo de *SIFS e DIFS*, respectivamente, L_{ACK}/C_w é o tempo de transmissão do *ACK* e $L_{2,j}^{min}/C_w$ é o tempo de transmissão de $P2$ da j-ésima amostra da sequência δ_j^{min}.

Para cada uma das doze taxas de transmissão definidas para os padrões IEEE 802.11a/b/g, uma função diferente é definida para D_{j,C_w}. A Figura 4.5 ilustra um gráfico com algumas das doze funções definidas para os limites inferiores da dis-

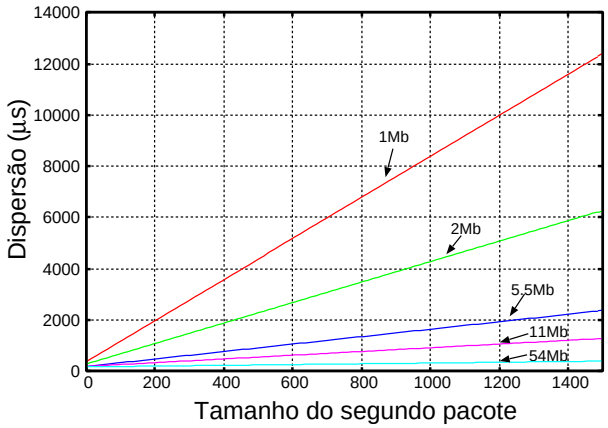


Figura 4.5: Funções dos limites inferiores para a dispersão dos pares de pacotes.

persão dos pares de pacotes. A Tabela 3 apresenta os valores dos termos utilizados pela Equação 4.2. Note que, na tabela, os valores referentes aos tempos de transmissão dos dados e do *ACK* estão somados aos tempos de transmissão do cabeçalho da camada física. Como as funções são definidas em relação à taxa de transmissão, algumas suposições são feitas para obtermos uma única função por taxa, independente do padrão considerado. Por exemplo, para as taxas de transmissão dos padrões 802.11 a/g assume-se os valores de *DIFS* e *SIFS* como o menor entre os dois padrões; o mesmo foi feito para o tempo de transmissão do cabeçalho da camada física dos padrões 802.11b/g+legado.

Para determinar a taxa de transmissão do salto sem fio, o passo final do algoritmo consiste em calcular o *MSE (Mean Square Error)* entre os valores de dispersões dos pares selecionados nos experimentos (que formam a sequência δ_j^{min}) e os limites inferiores obtidos com as funções D_{j,C_w}, para todo valor de C_w. A estimativa para a taxa de transmissão do último salto sem fio é determinada por:

$$C_{tx} = \min_{\forall C_w}\{MSE(\delta_j^{min}, D_{j,C_w})\} \tag{4.3}$$

A técnica proposta pode ser resumida pelo algoritmo descrito abaixo:

C_w	t_{DIFS}	t_{SIFS}	L_ACK/C_w	$L_{2,j}^{min}/C_w$
1	50	10	$(14*8/1)+192$	$(L_{2,j}^{min}*8/1)+192$
2	50	10	$(14*8/2)+192$	$(L_{2,j}^{min}*8/2)+192$
5.5	50	10	$(14*8/5.5)+192$	$(L_{2,j}^{min}*8/5.5)+192$
11	50	10	$(14*8/11)+192$	$(L_{2,j}^{min}*8/11)+192$
6	28	10	$(14*8/6)+192$	$(L_{2,j}^{min}*8/6)+192$
9	28	10	$(14*8/9)+192$	$(L_{2,j}^{min}*8/9)+192$
12	28	10	$(14*8/12)+192$	$(L_{2,j}^{min}*8/12)+192$
18	28	10	$(14*8/18)+192$	$(L_{2,j}^{min}*8/18)+192$
24	28	10	$(14*8/24)+192$	$(L_{2,j}^{min}*8/24)+192$
36	28	10	$(14*8/36)+192$	$(L_{2,j}^{min}*8/36)+192$
48	28	10	$(14*8/48)+192$	$(L_{2,j}^{min}*8/48)+192$
54	28	10	$(14*8/54)+192$	$(L_{2,j}^{min}*8/54)+192$

Tabela 4.3: Valores dos termos da Equação 4.2, para cada uma das taxas de transmissão dos padrões IEEE 802.11a/b/g.

Algoritmo 4.1 Estimando a taxa de transmissão da rede de acesso sem fio.

Passo 1: Utilizando alguma das técnicas existentes (e.g., [123, 124, 125]), identificar o tipo de conexão do último salto. Se é uma rede sem fio, então prosseguir com os Passos 2-5;

Passo 2: Gerar uma sequência de m grupos de pares de pacotes e coletá-los no receptor;

Passo 3: No receptor, computar a dispersão δ_j^k de todos os $(4*m)$ pares, onde $k = 1, ..., m$ é o índice do grupo e $j = 1, 2, 3, 4$ é o índice de um par em particular do grupo;

Passo 4: Usando a Equação 4.1, selecionar a menor dispersão para todos os valores de $j = 1, 2, 3, 4$ e obter δ_j^{min};

Passo 5: Estimar C_{tx} utilizando a Equação 4.3, que é determinada pelo menor *MSE* computado entre as amostras selecionadas do experimento (δ_j^{min}) e as funções D_{j,C_w} para todos os valores de C_w.

4.3.2 Ajuste automático da taxa de transmissão

Como foi mencionado na Seção 4.2, o padrão IEEE 802.11 prevê um ajuste automático da taxa de transmissão, dependendo das condições existentes no meio de propagação do sinal. Embora o ajuste automático da taxa de transmissão não seja habilitado em todas as redes locais, e nem mesmo implementado por alguns fabricantes, é desejável que o método proposto seja capaz, inclusive, de detectar essas eventuais alterações nos enlaces medidos.

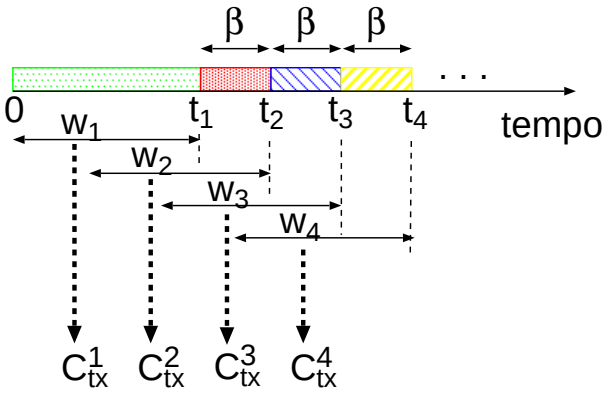


Figura 4.6: Dinâmica do algoritmo para computar a taxa de transmissão.

Para que a medida seja feita de forma dinâmica, os grupos de pares de pacotes são gerados continuamente, durante todo o período de interesse. Para a estimativa da taxa de transmissão (C_{tx}), uma janela de W pares de pacotes é utilizada pelo Algoritmo 4.1 (nesse caso, $m = W/4$). Para uma nova estimativa, a janela desliza por β pares de pacotes. As novas β dispersões substituem as amostras mais antigas e o algoritmo estima novamente C_{tx}. A dinâmica do algoritmo é ilustrada pela Figura 4.6. A cada instante t_i (para $i = 1, 2, ...$), uma nova estimativa de C^i_{tx} é obtida, utilizando os W pares de pacotes contidos na janela w_i.

Claramente, existe um *trade-off* entre os valores definidos para W e β e a precisão da estimativa. O tamanho da janela W tem que ser grande o suficiente para a obtenção de resultados precisos. Quanto maior for o valor de W, mais amostras são utilizadas pelo algoritmo e, com isso, maior é a probabilidade das amostras selecionadas (que formam a sequência δ^{min}_j) terem chegado juntas ao último salto, terem sofrido pouca influência de tráfego concorrente e terem valores pequenos de *backoff* para $P2$. Em compensação, valores muito grandes de W retardam a estimativa ou exigem uma redução no intervalo entre o envio de pares, aumentando a sobrecarga

na rede. Já o parâmetro β determina a frequência com que as taxas de transmissão devem ser recomputadas. Se esse valor for pequeno, por exemplo $\beta = 1$, uma nova taxa é estimada a cada novo par de pacote recebido. Quanto menor o valor de β, mais rápida será a identificação de alterações na taxa de transmissão. Na seção de validação da técnica (Subseção 4.4), essa questão voltará a ser abordada.

4.4 Validação

Para validar a técnica proposta e avaliar a sua eficiência, foram realizados experimentos reais, em ambientes controlados e na Internet, e foi utilizado um modelo de simulação desenvolvido no NS-2[129]. Os experimentos tinham como objetivo analisar a técnica em ambientes reais de características distintas (por exemplo, quando o canal de contenção é o enlace sem fio e quando é algum outro canal do caminho de rede). Já as simulações tiveram como objetivo analisar a eficiência da técnica quando o enlace sem fio medido está configurado para operar com a opção de ajuste automático da taxa de transmissão. Nesta seção serão apresentados os resultados obtidos.

4.4.1 Resultados de experimentos

Diversos experimentos foram executados, utilizando dois cenários distintos. Em todos eles sondas foram geradas, conforme os requisitos da técnica proposta, a uma taxa de 40 pares de pacotes por segundo (equivalente a 96KBps), durante 10 segundos. (Em cada sessão de experimento, foram gerados $m = 100$ grupos de 4 pares de pacotes.) Os resultados das estimativas foram comparadas às diferentes taxas de transmissão do ponto de acesso da rede sem fio, que foi configurado para operar sem o controle automático de taxa. Nesses experimentos, o objetivo foi avaliar a precisão das estimativas obtidas com a técnica proposta.

O primeiro conjunto de experimentos foi realizado na rede local do departamento PESC/UFRJ. A Figura 4.7 ilustra a topologia utilizada. O cenário consiste de duas máquinas fonte ($A1$ e $A2$), conectadas à rede por um mesmo *switch*, e duas máquinas destino ($B1$ e $B2$), conectadas por um ponto de acesso a uma rede sem fio IEEE 802.11g. Os pacotes enviados pelas máquinas fonte atravessam dois roteadores,

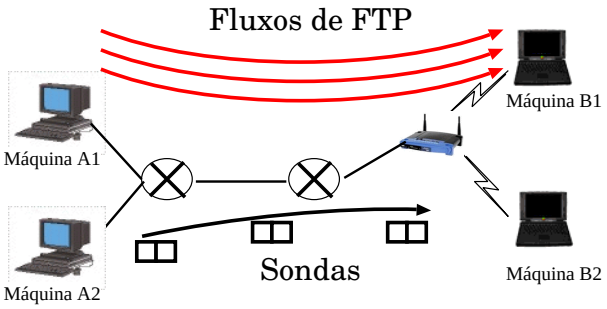


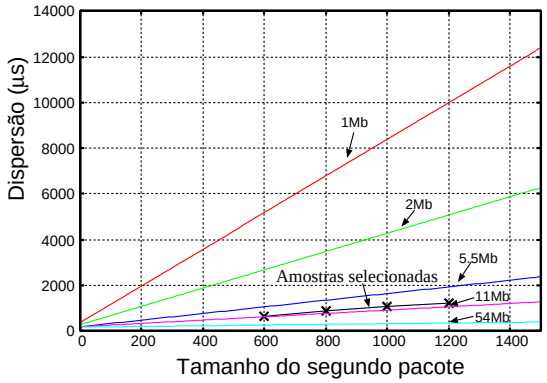
Figura 4.7: Cenário utilizado no primeiro experimento.

antes de chegar às máquinas destino: *COS1* (roteador do departamento) e *Araruama* (roteador do laboratório LAND). Exceto o salto sem fio, que foi configurado para operar em diferentes taxas, a capacidade de todos os saltos do caminho de rede era de 100 *Mbps*. O objetivo foi avaliar o comportamento da técnica para diversas taxas de transmissão.

As sondas, utilizadas para inferir a taxa de transmissão do último salto no caminho, foram geradas da máquina $A2$ para a máquina $B2$. Simultaneamente, três fluxos de *FTP* foram estabelecidos entre as máquinas $A1$ e $B1$, com o propósito de produzir tráfego concorrente no caminho de rede percorrido pelas sondas. Os fluxos de *FTP* permaneceram em atividade ao longo de todo o experimento. A rede utilizada não esteve dedicada, exclusivamente, para esses experimentos. Durante todo o período de medição, o tráfego gerado por outras aplicações, utilizadas por usuários deste ambiente, também concorreram com as sondas dos experimentos.

Inicialmente, o ponto de acesso da rede sem fio foi configurado para operar a uma taxa de $11 Mbps$. Os valores de dispersão, das amostras selecionadas para a sequência δ_j^{min}, são mostradas na Figura 4.8. Algumas das funções de dispersão (D_{j,C_w}) definidas para as taxas de transmissão são também ilustradas no gráfico. Visualmente, é possível verificar a proximidade dos valores obtidos pelo experimento com a função D_{j,C_w} definida para $C_w = 11 Mbps$. A Figura 4.8 também mostra o resultado do *MSE* das funções D_{j,C_w} para todas as capacidades definidas. Note que a taxa de transmissão estimada para o experimento pela técnica proposta neste trabalho foi a taxa real de $11 Mbps$.

Neste mesmo cenário, a técnica foi testada exaustivamente, com o ponto de acesso sendo configurado diversas vezes para operar a diferentes taxas de transmissão. Para



Taxa	MSE
1	48194925
2	10231681
5.5	824976
6	639383
9	286762
11	=>244734<=
12	494648
18	283931
24	337879
36	413688
48	459787
54	476372

Figura 4.8: Resultado do experimento pelo método proposto com a rede sem fio operando a 11*Mbps*.

todos os valores, o algoritmo estimou corretamente a taxa de transmissão do enlace sem fio. Os resultados obtidos para as taxas de 5.5 Mbps e 54 Mbps estão ilustrados nas Figuras 4.9 (A) e (B), respectivamente. Os resultados do *MSE* para as funções mais próximas da taxa configurada são também mostrados na figura.

Taxa	MSE
1	33682846
2	4161902
5.5	=>127568<=
6	242075
9	827492
11	1082290

Taxa	MSE
12	360024
18	154570
24	84633
36	36558
48	20719
54	=>16654<=

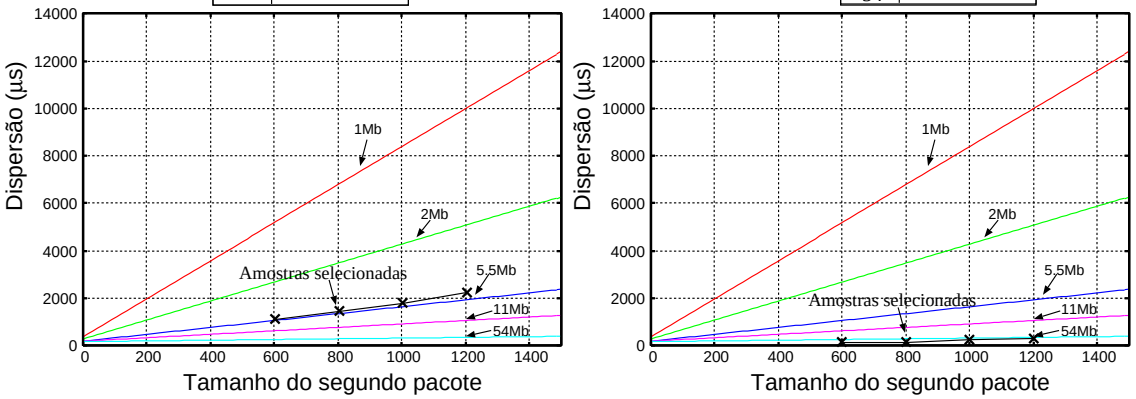
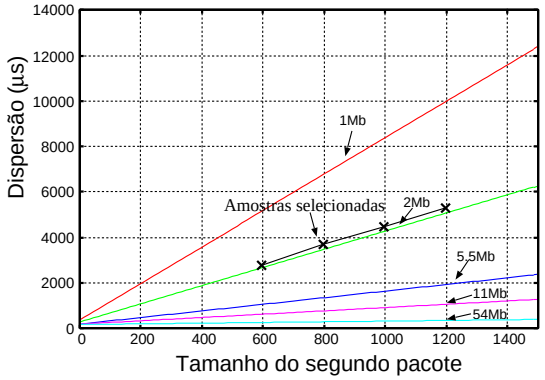


Figura 4.9: Resultado do experimento com o método proposto com a rede sem fio operando com as taxas: (A) 5.5*Mbps*; e, (B) 54*Mbps*.

No segundo conjunto de experimentos foi considerado um cenário onde a rede sem fio não era o canal de contenção do caminho entre a origem e o destino das sondas. Os pares de pacotes foram gerados de uma máquina do LAND/UFRJ para a máquina de destino, localizada em uma residência (do Rio de Janeiro), dotada de

uma rede sem fio. Onze roteadores existem entre as máquinas fonte e destino. A rede sem fio, ao qual a máquina destino encontrava-se conectada, operava a uma taxa de 2 Mbps. Um ponto relevante para este experimento é que a capacidade de transmissão do penúltimo salto era de 512 *Kbps*, portanto, inferior à taxa de transmissão configurada no ponto de acesso. A Figura 4.10 apresenta os resultados obtidos para este experimento, com o *MSE* para as diferentes taxas, demonstrando, novamente, a precisão da técnica para estimar a taxa de transmissão do enlace sem fio, em experimentos reais.



Taxa	MSE
1	10980400
2	=>208884<=
5.5	8106042
6	8895886
9	11564614
11	12469772
12	13030199
18	14582867
24	15392299
36	16223556
48	16647337
54	16789594

Figura 4.10: Resultados de experimentos quando a rede sem fio não é o canal de contenção e opera a $2Mbps$.

4.4.2 Resultados de simulações

A Figura 4.11 ilustra a topologia utilizada para o modelo de simulação desenvolvido no NS-2. Os nós $S1$ e $S2$ representam as máquinas fontes do tráfego gerado para as máquinas receptoras, representadas na figura pelos nós $W1$ e $W2$. O caminho de rede percorrido pelo tráfego das fontes ($S1$ e $S2$) até os destinos ($W1$ e $W2$) consiste de três saltos cabeados e um último salto sem fio. As capacidades atribuídas aos enlaces $L1$, $L2$ e $L3$ são iguais a $100Mbps$, já a capacidade definida para o enlace $L4$, entre o roteador $R2$ e o ponto de acesso, é igual a $10Mbps$. O valor de $10Mbps$, definido para $L4$, foi escolhido para possibilitar a análise de cenários em que a rede sem fio não seja o canal de contenção do caminho. O atraso de propagação configurado em todos os canais foi de 10ms.

Os pares de sondas são geradas pela máquina $S2$ para a máquina $W2$ e utilizados para inferir a taxa de transmissão do enlace $L4$. Em paralelo, três conexões *TCP*

são estabelecidas entre $S1$ e $W1$, para simular o tráfego concorrente.

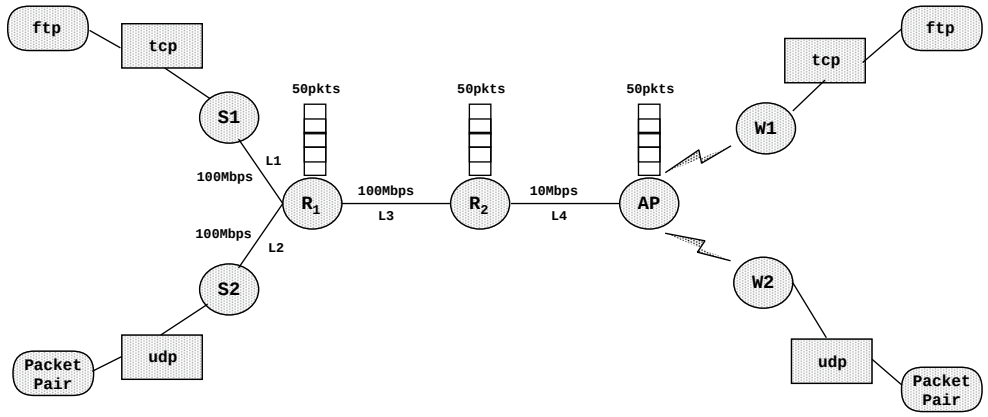


Figura 4.11: Modelo de simulação utilizado no NS-2.

Para simular a função de ajuste automático de taxa existente no padrão IEEE 802.11, foram utilizados *traces* de amostras coletadas de um experimento realizado no ambiente do laboratório LAND, utilizando dispositivos configurados para operar com o ajuste automático de taxa. No experimento, voluntários com *laptops*, conectados ao ponto de acesso da rede local sem fio, andaram livremente pelo laboratório, enquanto amostras da taxa de transmissão dos dispositivos eram coletadas por *scripts* em execução naquelas máquinas. Diferentes intervalos de coleta foram definidos para os experimentos. Nos dois primeiros, amostras foram coletadas a cada segundo, durante 5 minutos. No último experimento, amostras foram coletadas a cada 30 segundos, durante 25 minutos. Os dados coletados nesses experimentos foram utilizados pelos dispositivos sem fio, definidos no modelo de simulação, para representar o ajuste automático da taxa de transmissão.

Os valores dos parâmetros do algoritmo utilizados na simulação foram: (i) as sondas são geradas a uma taxa de 20 pares de pacotes por segundo (i.e., 48KBps); (ii) $W = 20$ e $\beta = 1$ quando foi usado um trace com amostras geradas a cada segundo; (iii) $W = 160$ e $\beta = 1$ quando foi usado o trace com amostras geradas a cada 30 segundos. Com esses parâmetros, após a chegada dos W primeiros pares, um novo C_{tx} é estimado a cada novo par de sonda recebida.

As Figuras 4.12(A) e 4.13(A) mostram os dois primeiros resultados de simulação. Nessas duas rodadas de simulação, foram utilizados *traces* com amostragens a cada segundo da taxa de transmissão. É possível verificar, visualmente, em ambos os gráficos, a proximidade das duas linhas: a linha sólida, que representa a taxa de

transmissão estimada pelo algoritmo, e a linha tracejada, que representa a taxa real, coletada pelos experimentos e utilizada para alimentar os modelos. Note que o algoritmo foi capaz de capturar com grande precisão o comportamento dinâmico da taxa de transmissão do dispositivo sem fio, durante a simulação. São poucos os intervalos em que a taxa estimada difere da taxa real. Como já foi mencionado, os erros podem ser atribuídos à interferência de tráfego concorrente entre os pares de sondas e/ou longos períodos de *backoff*, ocorridos nas transmissões do segundo pacote dos pares.

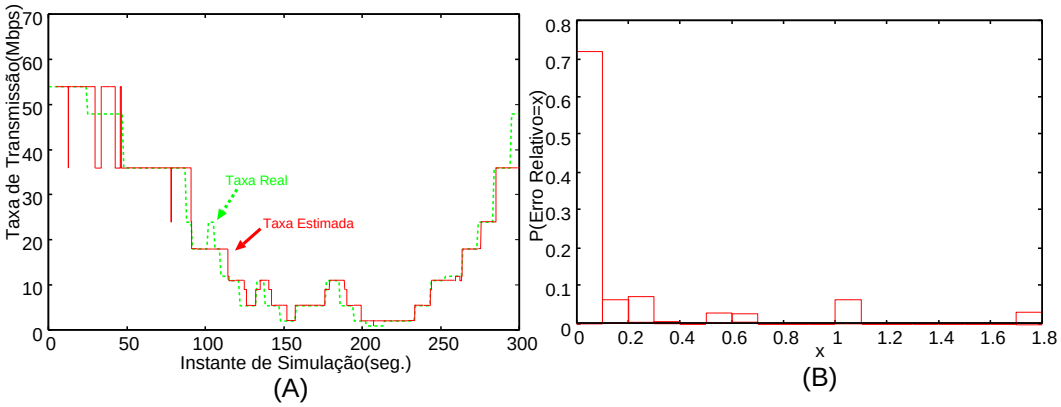


Figura 4.12: Resultados de simulação utilizando ajuste automático de taxa - intervalo de 1 segundo por amostragem (rodada 1).

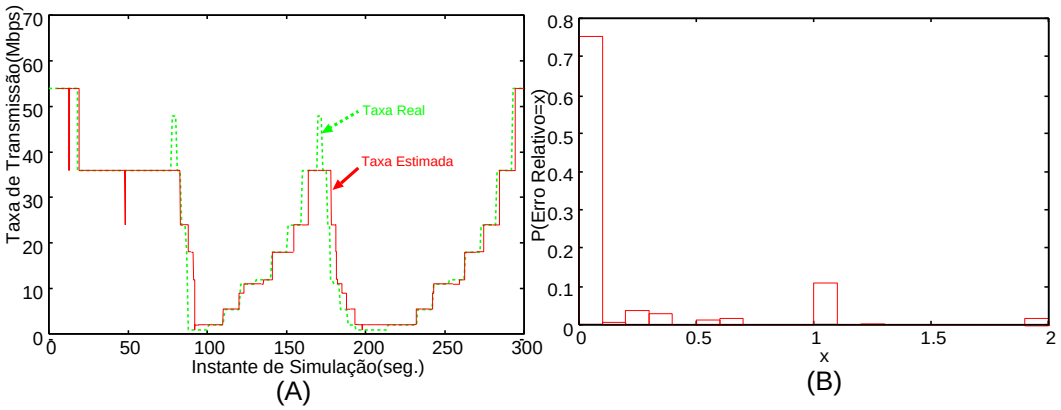


Figura 4.13: Resultados de simulação utilizando ajuste automático de taxa - intervalo de 1 segundo por amostragem (rodada 2).

Para ilustrar a precisão das estimativas, os erros relativos computados são mostrados nas Figuras 4.12(B) e 4.13(B). Cada barra representa um intervalo de 10%. Pelos gráficos é possível observar que o erro relativo foi inferior a 10% em mais

de 70% das estimativas, e menos de 20% das estimativas apresentam erro relativo superior a 20%.

Os resultados de simulação, utilizando o *trace* com amostragem de maior intervalo, são mostrados nas Figuras 4.14(A) e 4.14(B). Neste cenário, o intervalo entre coletas foi de 30 segundos e um número maior de amostras foi utilizado pelo algoritmo para estimar a taxa de transmissão. Enquanto os resultados mostrados anteriormente (Figuras 4.12 e 4.13) foram obtidos utilizando 20 pares de pacotes ($W = 20$), o resultado da Figura 4.14 é baseado em 160 pares ($W = 160$). Nota-se nos gráficos o aumento na precisão das estimativas, para um valor maior de W. Pela Figura 4.14(B), por exemplo, verifica-se que 87% das estimativas possuem um erro relativo inferior a 20%.

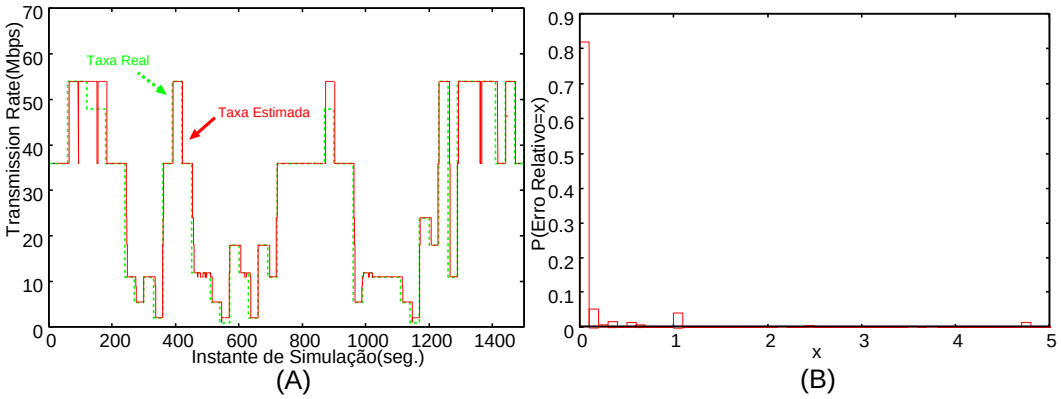


Figura 4.14: Resultados de simulação utilizando ajuste automático de taxa - intervalo de 30 segundos por amostragem.

Por fim, considerando ainda os resultados mostrados na Figura 4.14, é possível verificar que o algoritmo é mais acurado para detectar transições de aumento da taxa de transmissão. Para compreender esse fato, considere um evento de transição da taxa de transmissão de um enlace que operava a $5.5Mbps$ e passou a operar a $11Mbps$. Suponha também que, no momento do algoritmo estimar o novo C_{tx}, existem ainda algumas amostras de dispersão que foram coletadas quando a taxa estava operando a $5.5Mbps$. Logo, como essas dispersões "antigas" são maiores, elas não serão selecionadas, entre as de menor dispersão, para a estimativa do algoritmo. Por outro lado, se houver um decrescimento da taxa de transmissão, as "antigas" amostras de dispersão serão utilizadas até que não existam mais amostras em W referentes à essa dispersão. Assim, durante um curto intervalo de tempo, a taxa

real de transmissão estará sendo subestimada pelo algoritmo. Algumas ocorrências como essa podem ser identificadas nos gráficos, especialmente nas Figuras 4.12(A) e 4.13(A).

4.5 Conclusão

O desenvolvimento de técnicas fim-a-fim para inferir o tipo de conexão do último salto de um caminho de rede pode ser útil para diversos serviços e protocolos na Internet. Na literatura, já foram propostas técnicas que permitem uma diferenciação entre os tipos de acesso mais comuns: *ADSL*, *Cable Modem*, *Ethernet* e *WLAN* [124, 123, 125]. No entanto, o objetivo destas propostas se limita apenas à classificação do tipo de acesso. Outras informações importantes como a taxa de transmissão dos dados transmitidos nas redes locais sem fio não são estimadas por estas técnicas.

Nesta seção foi apresentada uma técnica de medição fim-a-fim, proposta para inferir a taxa de transmissão de uma máquina conectada à Internet, através de uma rede sem fio IEEE 802.11. A técnica desenvolvida é baseada em uma variação do método de pares de pacotes, com um filtro de seleção de pares e uma equação para definir a dispersão dos pares em uma rede IEEE 802.11a/b/g.

Resultados de experimentos e simulação comprovaram a eficiência do método proposto. Experimentos realizados em ambientes reais demonstraram a precisão da técnica. Já os resultados de simulação mostraram também que o algoritmo tem capacidade de determinar de forma eficiente a taxa de transmissão, mesmo quando a opção de ajuste automático de taxa estiver habilitada pelo dispositivo sem fio.

Capítulo 5

O uso de aplicações peer-to-peer para aumentar a disponibilidade e reduzir o custo da distribuição de conteúdo na Internet

ESTE capítulo discorre sobre uma análise experimental de larga escala realizada para avaliar o desempenho de protocolos P2P, como o BitTorrent, na disseminação de conteúdo na Internet. Os resultados obtidos demonstram que a distribuição de arquivos de forma agrupada, ao invés de arquivos isolados, pode aumentar significativamente a disponibilidade deste conteúdo e que um conteúdo muito popular pode ser distribuído a custo (quase) zero, sem degradação de desempenho para o usuário. Uma visão geral do protocolo BitTorrent é descrita na Seção 5.1. A Seção 5.2 apresenta uma análise sobre as implicações da popularidade do *swarm* na disponibilidade dos blocos e no custo para disseminação do conteúdo pelo BitTorrent. Um estudo experimental sobre o aumento da disponibilidade com a disseminação de arquivos agrupados é apresentado na Seção 5.3. Finalmente, na Seção 5.4, é apresentada uma avaliação sobre soluções para a redução de custos na distribuição de conteúdo via sistemas P2P.

5.1 Visão geral do protocolo BitTorrent

Dentre as diversas aplicações P2P existentes para disseminação de arquivos, BitTorrent é sem dúvida a mais popular de todas. Resultados de trabalhos recentes, já comentados na Seção 2.2 desta tese, denotam que mais de um terço de todo o tráfego atualmente gerado na Internet seria oriundo de aplicações BitTorrent. A popularidade dessas aplicações está relacionada às características fundamentais inerentes à arquitetura P2P, como auto-escalabilidade e maior robustez, que não são encontradas em aplicações de arquitetura cliente/servidor. No entanto, as políticas de reciprocidade instantânea, prioridade na recuperação de blocos mais raros (rarest-first) e incentivo de compartilhamento (tit-for-tat), inerentes e exclusivas do protocolo BitTorrent, tornam esse sistema ainda mais eficiente e mais robusto do que as outras redes P2P existentes (como, Naspter, Gnutela ou eDonkey2000), o que pode ser uma possível explicação para o imenso sucesso deste sistema.

A distribuição de um conteúdo no BitTorrent é feita por meio de um *swarm* (termo em inglês para "enxame"). O *swarm* é formado pelo conjunto de usuários (*peers*) interessados em recuperar ou disseminar um conteúdo, que pode consistir de um ou mais arquivos. Os *peers* que se encontram conectados ao *swarm* e que já possuem 100% do conteúdo recuperado são chamados de Seeders. Aqueles que ainda não recuperaram todo o conteúdo são denominados Leechers. Ao concluírem o *download*, os Leechers se tornam automaticamente Seeders e apenas fazem *upload* dos blocos do conteúdo. Antes de se tornarem Seeders, os Leechers recebem dados de outros *peers* conectados ao *swarm* e também fazem *upload* para outros *Leechers* das partes já recebidas do conteúdo.

Devido à falta de incentivos no protocolo BitTorrent para que os *peers* permaneçam cooperando com o sistema, é comum que os Leerchers abandonem o *swarm*, assim que finalizarem o *download*. Os Seeders que possuem algum incentivo para a disseminação do conteúdo são chamados de Publishers. Uma lista de todos os *peers* conectados ao *swarm* é mantida atualizada por uma espécie de coordenador no sistema, chamado de Tracker.

O processo completo de distribuição de conteúdo, através de um *swarm* BitTorrent, pode ser dividido em três etapas distintas. A Figura 5.1 ilustra cada uma dessas etapas.

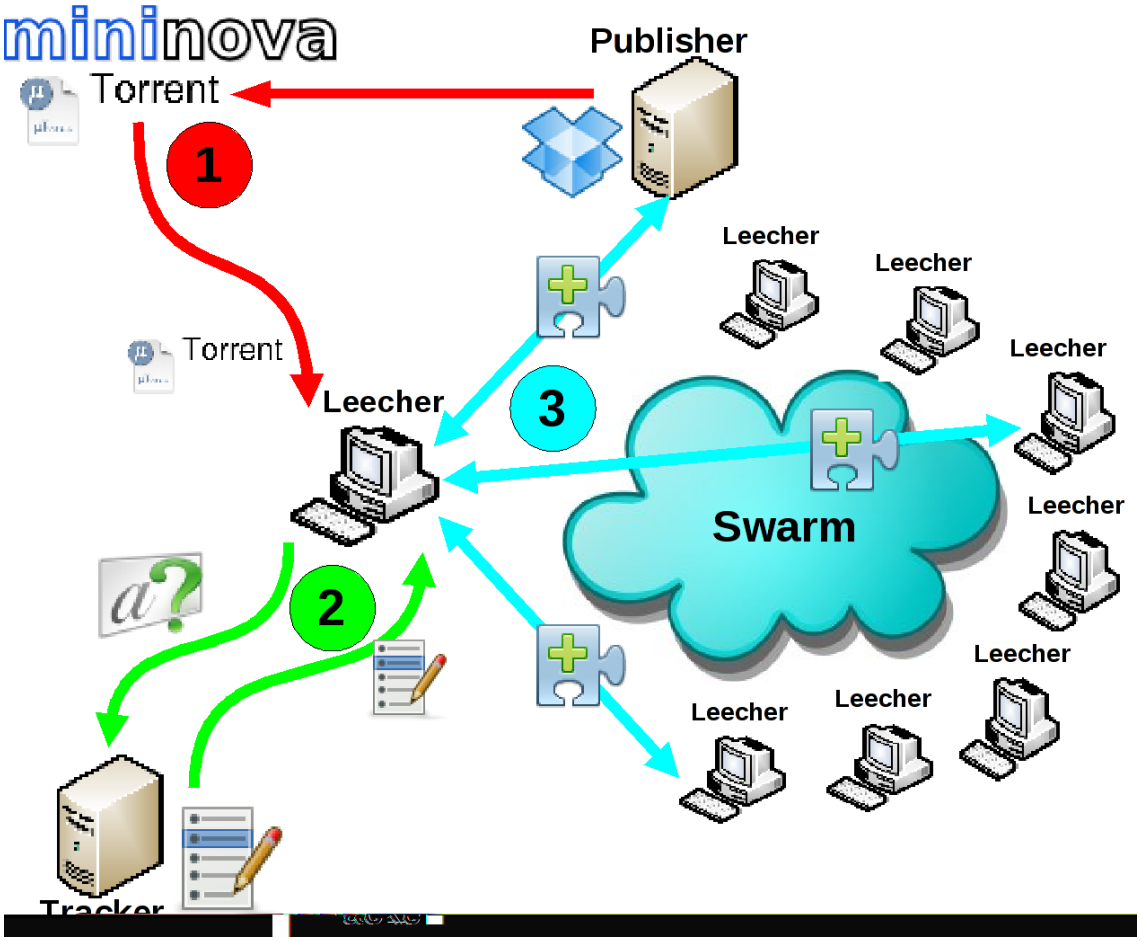


Figura 5.1: Etapas do processo completo de distribuição de conteúdo através de um *swarm* BitTorrent.

A primeira etapa (representada na figura em vermelho) consiste na definição de um *swarm* através da criação de um arquivo de referência, popularmente conhecido como ".torrent" (ou simplesmente torrent). Este arquivo pode ser criado por um usuário qualquer interessado em disseminar o conteúdo. Um torrent contém as informações necessárias para o funcionamento do *swarm* BitTorrent, como número e tamanho dos blocos (*chunks*) no qual foi dividido o conteúdo, quantidade e tamanho dos pedaços que formam um bloco, *hash* dos blocos e o endereço IP dos *Trackers* definidos para este *swarm*. Usuários interessados em fazer o *download* deste conteúdo devem obter o arquivo ".torrent" e utilizá-lo para que a aplicação BitTorrent possa se conectar ao *swarm*. Em geral, os torrents são disponibilizados pelos seus criadores em sites de busca e divulgação de *swarms* BitTorrent, como Mininova, The Piratebay e TorrentZ. Para que as etapas seguintes ocorram e o conteúdo possa ser recuperado pelos Leechers no sistema, o usuário criador do torrent deve dar início à operação

do Tracker e conectar pelo menos um Publisher ao *swarm*.

Na segunda etapa (representada em verde), os *peers* contactam o(s) Tracker(s) definido(s) pelo torrent em busca de conhecer outros *peers* também conectados àquele *swarm*. Os Trackers, sempre que solicitados, encaminham uma lista contendo os endereços IP's de um subconjunto aleatório dos *peers* conectados ao *swarm*. Periodicamente, os *peers* podem voltar a solicitar uma nova lista ao Tracker, atualizando a sua lista anterior. Essas listas também podem ser trocadas entre os *peers*, através do processo chamado PEX(*Peers Exchange*), definido pelo protocolo para que o sistema seja menos dependente dos Trackers.

De posse da lista com endereços de outros *peers* participantes do *swarm*, os nós passam para a terceira etapa do processo (representada em azul). É nesta etapa que os dados do conteúdo são, de fato, transmitidos e recebidos pelas aplicações BitTorrent. Os Leechers tentam estabelecer uma conexão com cada um dos *peers* existentes na sua lista. Todos aqueles que aceitarem a conexão formarão a sua vizinhança. Em seguida, esse Leecher envia uma mensagem para todos os seus vizinhos solicitando os seus respectivos *bitmaps* (mapas de bits, que representam a lista de blocos já recuperados e disponíveis por aquele *peer* para serem transmitidos).

O protocolo BitTorrent define que a troca de dados seja orientada a solicitações. Isto é, os dados são transmitidos pelos *peers* apenas à medida que são requisitados. Baseado na política *rarest-first*, os Leechers continuamente solicitam aos seus vizinhos, através da mensagem "Interested", aqueles blocos que se mostram mais raros dentre todos os *bitmaps* recebidos dos nós vizinhos.

Para agilizar o processo inicial de *download*, a política de *rarest-first* não é utilizada na recuperação dos primeiros blocos. Leechers que acabaram de se conectar ao *swarm*, e que ainda não possuem blocos em seus mapas de bits, requisitam aleatoriamente os blocos até que os n primeiros tenham sido recuperados. (Em geral, n é igual a quatro blocos.) Sempre que um *peer* concluir o *download* de um bloco, ele notifica todos os vizinhos, através da mensagem "Have".

Na fase final de recuperação dos últimos blocos, existe uma tendência de redução da taxa de *download*. Para tentar minimizar esse problema, o protocolo BitTorrent prevê um mecanismo de finalização (denominado "*End Game mode*"). Neste mecanismo, os Leechers na fase final do *download* devem enviar as mensagens de

"Interested" dos blocos restantes para todos os seus vizinhos. A finalidade deste mecanismo é agilizar a conclusão do *download*. No entanto, estudos apresentados em [130] contestam a eficiência do método, por não apresentar uma melhora significativa no tempo de *download* e aumentar a sobrecarga de mensagens.

Ao receber do vizinho a mensagem de interesse por um bloco, um *peer* deve decidir se irá ou não transmitir o bloco solicitado. Nem todas as solicitações podem ser contempladas. Isso porque, para que uma taxa de transmissão razoável seja alcançada, as aplicações BitTorrent limitam o número de *uploads* em paralelo. A política *tit-for-tat*, como mecanismo de incentivo instantâneo de compartilhamento do protocolo BitTorrent, define que *peers* devem, periodicamente, identificar os seus vizinhos mais generosos e retribuir fazendo *upload* dos dados solicitados por eles. Por isso, restringir o número máximo de vizinhos a servir por vez, possibilitando transmitir dados a uma taxa mais alta, pode influenciar positivamente no desempenho do tempo total de *download*, pois aumentam as chances de que *peers* estejam dispostos a retribuir pela generosidade enviando os blocos de interesse. Ao passo que, se o Leecher oferecer taxas muito baixas de *upload*, os *peers* que receberam os dados darão preferência a retribuir a generosidade daqueles outros vizinhos, de quem receberam dados a taxas mais altas. Cada *peer* deve gerenciar o estado das relações com todos os seus vizinhos, classificando cada uma das conexões como bloqueada (*choked*) ou desbloqueada (*unchoked*) para *upload* dos blocos.

A política *tit-for-tat* em sua forma pura inviabiliza a inicialização de novos Leechers, pois esses *peers* não possuem qualquer conteúdo para "barganhar" pelo compartilhamento. A forma pura dessa política também impossibilita a expansão da relação entre os vizinhos, uma vez que a decisão sobre compartilhar ou não no futuro dependeria da ocorrência de uma troca de dados prévia. Para solucionar essas duas questões, a política de compartilhamento *tit-for-tat* do BitTorrent opera juntamente com uma política de desbloqueio otimista (*optimistic unchoking)*, onde *peers* agem de forma altruísta, dedicando uma fração de sua capacidade de transmissão para servir Leechers em sua vizinhança, mesmo sem nunca ter sido servido por eles. Já os Seeders são em sua essência altruístas, pois estão sempre fazendo *upload* dos dados sem exigir reciprocidade.

5.2 Popularidade de um conteúdo e suas implicações nos *swarms* BitTorrent

Considere o caso em que um provedor deseja disseminar um ou mais arquivos para todos os usuários interessados, de forma que esse conteúdo fique o máximo de tempo disponível e a distribuição tenha o menor custo (em termos de consumo de banda) possível. Nesse contexto, sistemas P2P são, sem dúvida, uma opção natural para os provedores e, devido às particularidades inerentes ao protocolo BitTorrent (e.g, auto-escalabilidade, eficiência e robustez), este sistema aparece como uma solução razoável para os provedores. No entanto, a popularidade do conteúdo tem implicações diretas na disponibilidade e no custo para a distribuição do conteúdo em *swarms* BitTorrent.

Para compreender melhor os impactos da popularidade do *swarm*, simulações foram realizadas no ambiente de modelagem Tangram-II [74]. O modelo de simulação utilizado foi desenvolvido em [131] e trata-se de uma implementação detalhada do protocolo BitTorrent e seus elementos, como Tracker, Seeder e Leecher.

Diversas rodadas de simulação foram executadas, variando os valores dos seguintes parâmetros: tamanho do arquivo (S Bytes, dividido em B blocos de 256 KBytes cada), taxas máximas de *upload* dos *Leechers* (μ KBytes/segundo) e do Publisher (p KBytes/segundo), taxa de chegada dos Leechers (λ peers/segundo) e tempo total de simulação (T_{simul} segundos). Nas simulações, as chegadas dos Leechers ao *swarm* ocorrem em intervalos exponenciais e, ao se conectarem, os *peers* sempre encontram o Tracker e apenas um único Publisher em operação no sistema. Os Leechers permanecem conectados até a conclusão do *download*, quando, então, abandonam o *swarm*, sem atuarem como Seeders, e não mais retornam ao sistema. As ocorrências de chegada e partida dos Leechers, assim como todas as trocas de mensagens, são gravadas em um log, de onde são extraídas as medidas de interesse.

5.2.1 Impactos da popularidade do *swarm* na disponibilidade

Em sistemas P2P, um conteúdo é definido como disponível quando todas as partes dele estão à disposição dos usuários para serem recuperadas. Para isso, esse conteúdo

deve estar localizado por completo em um único *peer* ou em partes complementares e distribuídas entre os *peers* da rede.

A dinâmica da disponibilidade do conteúdo em um *swarm* é ilustrada pela Figura 5.2. Na figura, cada linha horizontal representa o intervalo de tempo que um *peer* ficou no sistema. Como no exemplo ilustrado, assume-se que os Leechers abandonam o sistema assim que recuperam 100% do conteúdo e, portanto, a linha associada a um Leecher representa o seu tempo total de *download*. No caso do Publisher, as linhas representam os intervalos de tempo que esteve conectado ao sistema. O *swarm*, que tem início no instante t_0 da figura com a chegada do primeiro Publisher, alterna entre períodos de disponibilidade e indisponibilidade de seu conteúdo. O conteúdo permanece disponível no sistema, enquanto o Publisher estiver conectado ao *swarm*. Quando o Publisher sair do sistema, como no caso ilustrado no instante t_1, o período de disponibilidade irá perdurar, se todas as partes do conteúdo estiverem disponíveis entre os Leechers conectados ao sistema. Leechers que chegarem ao sistema, mesmo que não encontrem um Publisher conectado, conseguirão concluir os seus respectivos *downloads*, enquanto todos os blocos estiverem disponíveis entre os Leechers do *swarm*. Eventualmente, um Leecher, ao concluir o seu *download*, pode deixar o sistema levando consigo a única réplica de um dos blocos do conteúdo. Neste caso, como ilustrado no instante t_2 da Figura 5.2, o conteúdo do *swarm* passa para o estado de indisponível. Os demais Leechers que já tinham iniciado o seu *download*, assim como outros que possam vir a se conectar ao sistema, permanecerão "bloqueados" no sistema e só conseguirão concluir a recuperação do conteúdo quando um Publisher retornar ao *swarm*, como é o caso ilustrado em t_3.

Devido à política de reciprocidade instantânea, não há incentivo para os *peers*, que já concluíram o *download*, permanecerem conectados fazendo *upload* e beneficiando o *swarm*. Por isso, a disponibilidade do conteúdo em um *swarm* tem forte dependência na existência de Publishers e na popularidade do conteúdo. Resultados de simulações, apresentados a seguir, evidenciam exatamente essa dependência e também sugerem que, juntamente com a popularidade do *swarm*, a política *rarest-first* exerce um papel fundamental para a manutenção da alta disponibilidade do conteúdo no sistema BitTorrent.

As simulações foram executadas considerando *swarms* de popularidades distin-

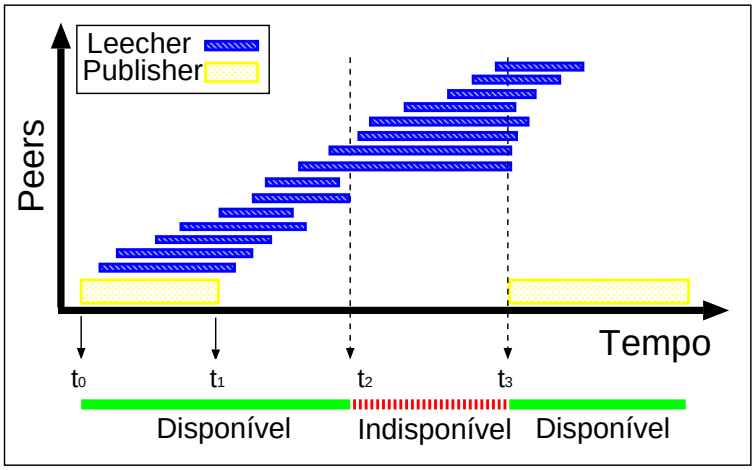


Figura 5.2: Dinâmica da disponibilidade de conteúdo em um *swarm*.

tas, variando a taxa de chegada dos Leechers (λ) entre 1 e 9 peers/minuto. Inicialmente, foi considerado um arquivo de tamanho $S \approx 4$MB ($B = 16$). Para cada valor de λ, foram realizadas 21 rodadas de simulação, cada uma com duração $T_{simul} = 10000$ segundos. As taxas máximas de *upload* definidas para os Leechers e para o Publisher foram, respectivamente, $\mu = 39$KBps e $p = 39$KBps, em todas as simulações. As simulações foram executadas também para um arquivo de tamanho $S \approx 13$MB ($B = 50$), considerando os mesmos valores para os demais parâmetros.

A Figura 5.3 mostra a média da fração de tempo em que se encontravam disponíveis, entre os Leechers conectados ao *swarm*, ao menos uma cópia de todos os 16 blocos (vermelho), de 15 blocos (verde), de 14 blocos (azul) e de 13 ou menos blocos (lilás). Os valores apresentados na figura representam a média dos tempos computados considerando as 21 rodadas, para cada um dos valores utilizados para λ nas simulações. Pelo gráfico, é possível notar que, quando a popularidade do *swarm* é baixa ($\lambda = 1/60$ peers/seg., por exemplo), na maior parte do tempo, apenas 13 ou menos blocos distintos encontravam-se replicados entre os *bitmaps* dos Leechers do *swarm*. No entanto, à medida que a popularidade aumenta, a fração de tempo em que é possível encontrar ao menos uma cópia de todos os 16 blocos distribuídos pelo sistema também cresce significativamente. Quando a taxa de chegada dos Leechers é maior que 4 peers/minuto, essa fração de tempo disponível é superior a 85%.

As medidas também foram computadas para as simulações considerando um arquivo maior ($S \approx 13$MB e $B = 50$). A Figura 5.4 apresenta os valores computados das frações de tempo em que estavam disponíveis entre os Leechers do *swarm* 50,

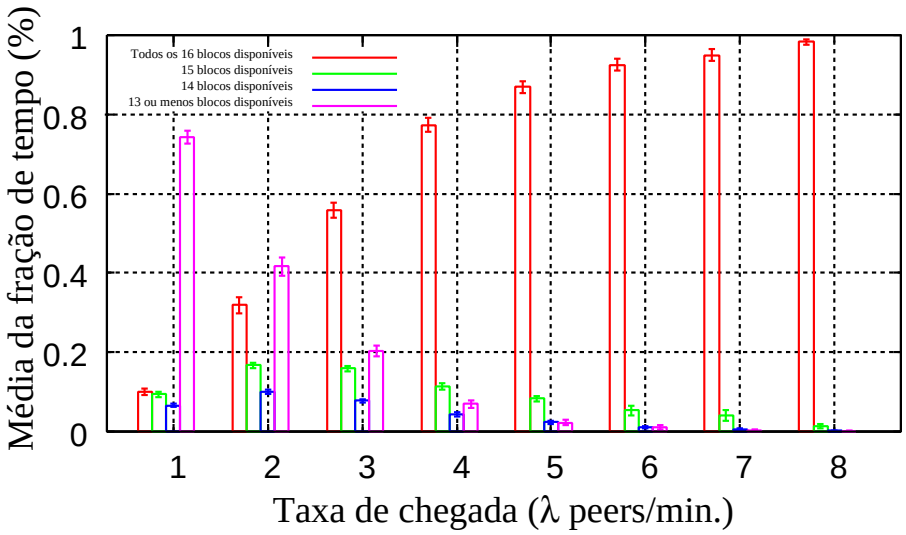


Figura 5.3: Fração de tempo que todos os 16 blocos encontravam-se replicados entre os Leechers do *swarm*.

49, 48 e 47 ou menos blocos. Os resultados também demonstram a tendência de crescimento da disponibilidade dos blocos entre os Leechers, com o aumento da popularidade.

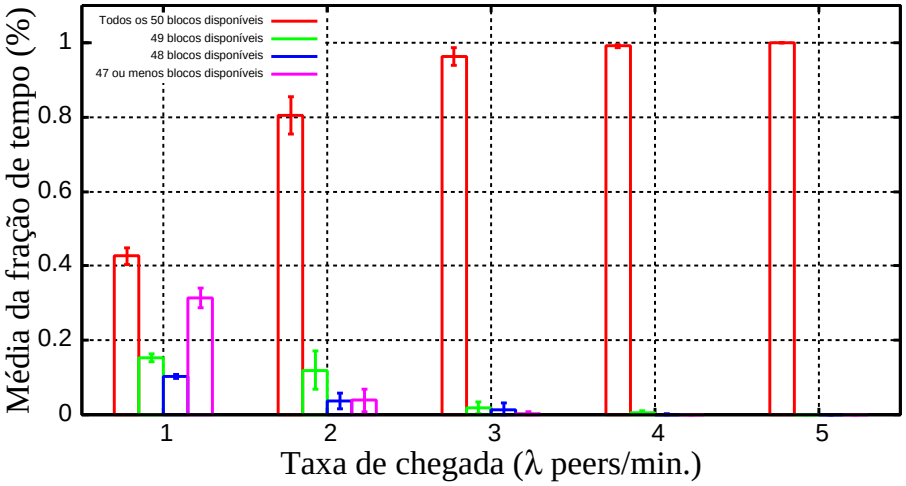


Figura 5.4: Fração de tempo que todos os 50 blocos encontravam-se replicados entre os Leechers do *swarm*.

Comparando os resultados apresentados nas Figuras 5.3 e 5.4, é possível constatar que, para uma mesma taxa de chegada, a disponibilidade é muito superior para o caso de $B = 50$. Para uma taxa de chegada dos Leechers $\lambda \geq 4$ peers/minuto, por exemplo, todos os 50 blocos do arquivo estiveram disponíveis em praticamente 100% do tempo de simulação. A razão para isso é que, aumentando o tamanho do

arquivo, mas mantendo as taxas de *upload* dos *peers*, o tempo médio de permanência dos Leechers no sistema, até concluírem o *download*, é maior e, consequentemente, aumenta o número médio de usuários no sistema e o número de blocos replicados pelo *swarm*.

O uso do mecanismo *rarest-first*, para seleção dos blocos a serem recuperados pelos *peers* no BitTorrent, possibilita uma distribuição balanceada dos blocos dentro do *swarm*. Esse mecanismo exerce um papel fundamental no crescimento da disponibilidade, em função do aumento da popularidade do conteúdo. Isso porque, apenas o aumento da população, sem a distribuição balanceada dos blocos, não garante uma uniformidade na disseminação e no número de réplicas dos blocos no sistema.

A eficiência do algoritmo *rarest-first* para a disseminação balanceada dos blocos pode ser verificada no gráfico da Figura 5.5. Nele são mostrados os números médios de réplicas no sistema de cada um dos 16 blocos, para simulações com λ igual a 1, 4 e 7 peers/minuto e com tamanho do arquivo $S \approx 4$MB ($B = 16$). Os valores apresentados no gráfico correspondem a um sistema bem balanceado. Embora apenas a média final seja mostrada na figura, esse comportamento foi observado durante todo tempo de simulação.

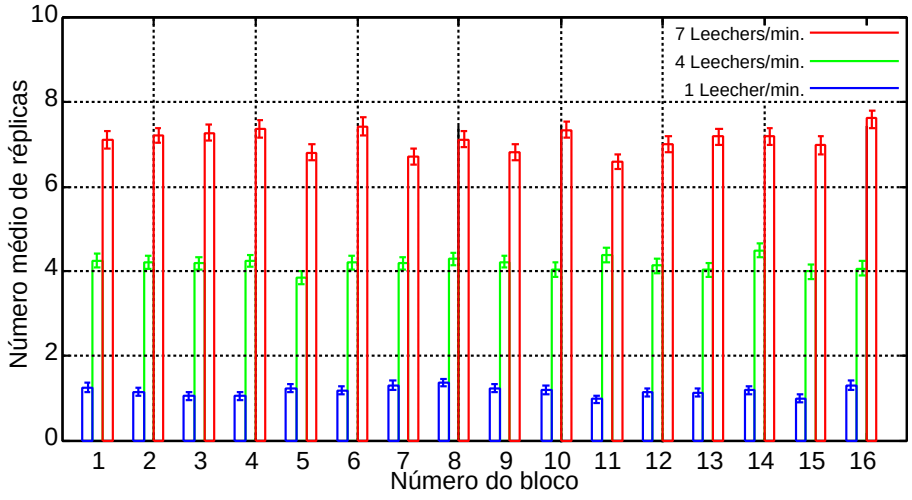


Figura 5.5: Número de réplicas de cada bloco no *swarm*.

5.2.2 Impactos da popularidade do *swarm* no custo para disseminação dos blocos

Considere um Publisher que está constantemente monitorando os mapas de bits dos *peers*, conectados ao sistema, e que só faça *upload* dos blocos que estiverem indisponíveis entre os Leechers do *swarm*. Neste cenário, o Publisher passa a ter dois estados distintos de operação: (i) "Ativo", em que atua transmitindo dados à sua taxa máxima de *upload*; (ii) "Inativo", quando para de transmitir blocos e permanece ocioso. Atuando dessa forma, o Publisher reduz a zero o consumo da banda, durante todo o período em que o conteúdo estiver disponível entre os Leechers do sistema.

A economia total de consumo da banda alcançada por um Publisher, que opera alternando entre estados de ativo e inativo, está relacionada à popularidade do conteúdo. Vejamos, como exemplo, os resultados obtidos pelas simulações apresentadas na subseção anterior. No modelo simulado do BitTorrent, o Publisher não implementa o modo de operação em dois estados e permanece contribuindo com *upload* durante todo o tempo de simulação. No entanto, se assumirmos que a capacidade do Publisher é uma contribuição marginal para a manutenção da disponibilidade dos blocos entre os Leechers do sistema, podemos analisar o impacto da popularidade do *swarm* na redução do consumo de banda do provedor, se este Publisher estivesse operando no modo "ativo/inativo".

A Figura 5.6 ilustra a fração de tempo que o Publisher precisa se manter ativo para prover blocos ao *swarm*. Além dos valores de $B = 16$ e $B = 50$ já mencionados na subseção anterior, a figura inclui também os resultados para as simulações considerando arquivos de tamanhos ainda maiores ($B = 100$ e $B = 200$). Nota-se que *swarms* impopulares são altamente dependentes do serviço do Publisher. À medida que a popularidade aumenta, a fração de tempo que o Publisher precisa permanecer ativo diminui, chegando próximo de zero para $\lambda \geq 8$ no caso de B=16 ($\lambda \geq 4$ no caso de B=50, $\lambda \geq 2$ no caso de B=100 e $\lambda \geq 1$ no caso de B=200). O tamanho do arquivo também exerce um papel crucial para a disponibilidade do arquivo. Quanto mais dados os usuários precisam baixar, mais tempo eles permanecem conectados cooperando com o sistema e, com isso, menor é a taxa de chegada necessário para a manutenção da disponibilidade de todos os blocos do conteúdo entre os Leechers

do *swarm*.

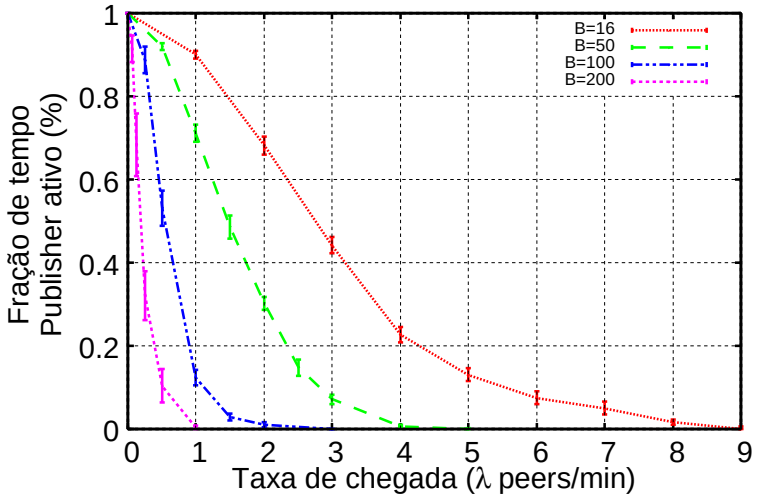


Figura 5.6: Implicações da popularidade do *swarm* na redução do custo para disseminação do conteúdo.

5.2.3 Tempo médio de *download* dos blocos

As subseções anteriores mostraram, através de simulações, que o aumento da popularidade do *swarm* tem implicações significativas no aumento da disponibilidade dos blocos e na redução do consumo de banda de Publisher. No entanto, é importante verificar ainda se a diferença na popularidade dos *swarms* influencia também o desempenho do sistema (tempo de *download* dos blocos pelos usuários). Para isso, foram realizadas simulações e os resultados mostram que o aumento da popularidade é inconsequente para a performance experimentada pelos usuários.

A Figura 5.7 ilustra a distribuição do tempo necessário para que os Leechers concluíssem o *download* do i-ésimo bloco. Os resultados apresentados na figura são referentes às simulações para três valores distintos de popularidade (λ igual a 1 peer/min., 4 peers/min. e 7 peers/min.). O gráfico mostra, para cada uma dessas popularidades, os valores estimados para os percentuais de 25%, 50% e 75%, além da média e dos valores mínimos e máximos, da distribuição do tempo de *download* de cada bloco.

O tempo médio de *download* para todos os blocos, exceto o primeiro, é aproximadamente igual para os três valores de λ mostrados no gráfico. A explicação para a diferença no tempo médio para recuperar o primeiro bloco é que os Leechers novos

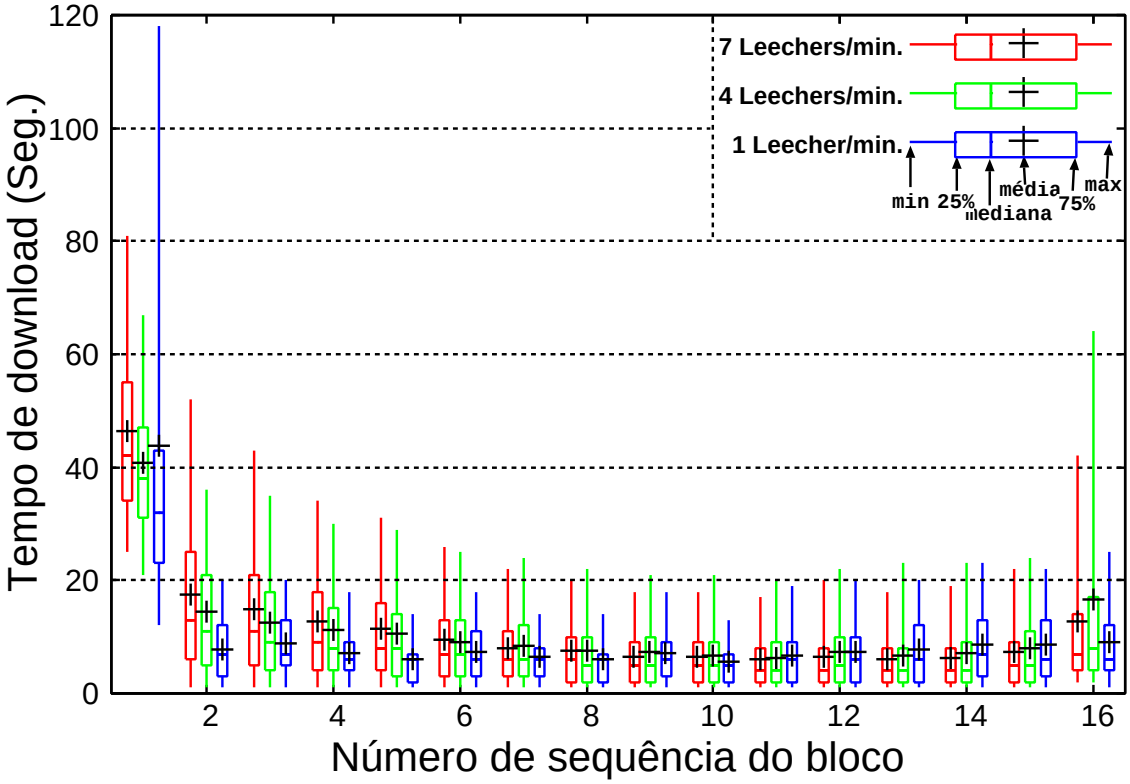


Figura 5.7: Distribuição do tempo médio de download de cada bloco no *swarm*.

no sistema dependem de um desbloqueio altruísta para iniciar o *download*. Pelos resultados, é possível verificar que o desempenho experimentado pelos usuários não foi afetado pelo crescimento na popularidade do conteúdo.

5.3 Aumento da disponibilidade do conteúdo através do agrupamento de arquivos

Os resultados, apresentados na seção anterior, sugerem que sistemas como o BitTorrent são altamente escaláveis e eficientes na disseminação de conteúdos muito populares. No entanto, esse sistema pouco pode fazer para auxiliar provedores e usuários na disponibilidade dos conteúdos em *swarms* pouco populares. Por isso, esses *swarms* são altamente dependentes da existência de um Publisher. Apesar do imenso sucesso do BitTorrent, a análise do monitoramento de milhares de torrents, apresentados em [25], demonstrou que os *swarms* BitTorrent sofrem de grande períodos de indisponibilidade, e.g., 75% de todos os *swarms* monitorados ficaram indisponíveis por mais de 80% do tempo medido (vide Figura 2.12 mostrada na

Subseção 2.2.2).

Uma prática comum, atualmente, no BitTorrent, identificada pelos experimentos de monitoramento apresentados em [25], é a disseminação de arquivos agrupados. Em uma análise feita em mais de 1 milhão de torrents disponibilizados pelo site Mininova, no período de maio de 2009, foram detectados diversos *swarms* formados por múltiplos arquivos. Das novas categorias definidas pelo site para classificação de conteúdo, em pelo menos três destas categorias (Música, Série de TV e Livros), o número de *swarms*, no qual foram identificados agrupamentos de arquivos, é significativo.

Na categoria Música, diversos arquivos de extensão ".mid", ".mp3" e ".wav" são disponibilizados em um único torrent. Um *swarm* que disponibiliza, por exemplo, todas as músicas de um álbum, trata-se de um agrupamento de arquivos. Dos 267 mil *torrents* de música examinados, mais de 193 mil eram formados por múltiplos arquivos. Já na categoria Série de TV, é muito comum encontrar um conjunto de episódios de uma mesma série. Foram identificados 25 mil *swarms* com dois ou mais arquivos do tipo "avi" ou "mpeg", dentre os 164 mil torrents examinados. Na categoria Livros, um agrupamento pode ser um conjunto de arquivos "pdf" ou "djvu", e dos 66 mil torrents examinados, mais de 6 mil continham uma coletânea de 2 ou mais arquivos.

5.3.1 Evidências de benefícios com agrupamentos

Evidências, identificadas durante o monitoramento de milhares de torrents, sugerem que existe uma forte correlação entre o agrupamento de arquivos e uma maior disponibilidade do conteúdo dentro de *swarms* reais do BitTorrent. Um exemplo foi identificado em torrents da categoria Livros. Considerando todos os *swarms* desta categoria monitorado nos experimentos, em 62% deles não foi identificado sequer um único Seeder durante toda sessão de monitoramento. No entanto, esse número cai drasticamente para 36%, se forem considerados apenas os torrents de coletâneas de livros. Além disso, considerando todos os *swarms* da categoria livro, o número médio de *downloads* por *swarm* computado durante o monitoramento foi de 2578. Essa média sobe para 4216 *downloads*, se considerarmos apenas os *swarms* que ofereciam coletâneas de livro.

Outro exemplo são as evidências identificadas na categoria Séries de TV. Em uma busca pelos episódios da primeira temporada da série de TV "Friends", foram identificados 52 torrents relacionados a esse tema. Destes 52 *swarms*, em 23 foram identificados ao menos um Seeder conectado, enquanto que nos outros 29, o conteúdo esteve indisponível durante todo o monitoramento. Dos 23 *swarms* disponíveis, 21 eram constituídos por arquivos agrupados. Ao passo que, dos 29 torrents, cujo conteúdo estava indisponível, 22 consistiam de arquivos isolados.

A venda de produtos agregados (ou *Product Bundling*, em inglês) é uma estratégia de comércio bastante utilizada no mercado [132]. A estratégia consiste em ofertar dois ou mais produtos para venda, como se fosse um único produto. Esta prática é muito comum na venda de *softwares* (e.g., pacote Office da Microsoft), TV's a cabo (por exemplo, pacotes básico, intermediário e avançado de canais, ou combo agregando diferentes produtos, como TV, Telefone e Internet) e alimentação (com a venda de pacotes de refeições combinando alguns itens do cardápio).

Na literatura, existem duas formas diferentes de agregação (ou *bundling*): Agrupamento Simples, quando o consumidor pode apenas optar pela compra do pacote inteiro; e, Agrupamento Misto, quando os consumidores têm a opção de selecionar quais as partes do pacote desejam comprar. Essas duas estratégias também podem ser implementadas no sistema BitTorrent. Torrents podem ser criados contendo diversos arquivos agrupados em um único arquivo (i.e., ZIP, RAR ou ISO) ou agrupados de forma aberta. No primeiro caso, todos os usuários seriam obrigados a recuperar e compartilhar todas as partes do conteúdo. No segundo caso, os usuários poderiam optar por apenas parte do conteúdo. As duas formas de agrupamento foram extensamente identificadas no monitoramento dos torrents reais na Internet. Por questões de simplicidade, a análise dos benefícios desenvolvida neste trabalho considera apenas a forma simples de agrupamento de arquivos no BitTorrent. No entanto, é possível supor que existem benefícios também para a forma mista do agrupamento.

5.3.2 Modelo de disponibilidade do BitTorrent

Os benefícios do agrupamento de arquivos para a disponibilidade do conteúdo em *swarms* BitTorrent foram analisados com um modelo desenvolvido por Menasche

et al. em [25]. No trabalho, um *swarm* BitTorrent é modelado por um sistema de filas $M/G/\infty$. A chegada de um Publisher dá início à operação do *swarm*. Os Leechers chegam de acordo com um processo de Poisson com taxa λ e encontram o conteúdo disponível ou indisponível. Enquanto o conteúdo estiver disponível, o tempo de permanência dos *peers* no *swarm* é exponencial com média S/μ, onde S é o tamanho do conteúdo e μ a capacidade de *download* dos Leechers. Após a partida do Publisher, os *peers* continuam trocando dados entre si e concluindo o *download* até que o conteúdo se torne indisponível, o que ocorre quando o número de Leechers conectados (n) atingir um valor inferior a um determinado "limite de cobertura" (m).

Para o modelo descrito acima, o período de disponibilidade do conteúdo corresponde ao *busy period* de uma fila $M/G/\infty$. Considerando um caso de alta indisponibilidade do Publisher (i.e., taxa de chegada r e tempo médio de permanência u pequenos) e o "limite de cobertura" igual a um peer ($m = 1$). Então, o período de disponibilidade de um arquivo de tamanho S e de popularidade λ é dado por:

$$\frac{e^{\lambda S/\mu} - 1}{\lambda}. \tag{5.1}$$

Considere, agora, que um agrupamento formado por K arquivos, todos de tamanho igual a S e popularidade λ, sejam oferecidos por um *swarm* BitTorrent. A oferta de arquivos agrupados, ao invés de isolados, aumentaria o tamanho do conteúdo a ser recuperado pelos *peers*, para KS, e a taxa de chegada dos Leechers, para $K\lambda$, uma vez que todos os Leechers interessados em um dos K arquivos deverão recuperar todo o agrupamento. Com isso, o tempo necessário para cada *peer* recuperar todo o conteúdo será agora KS/μ e o período de disponibilidade do agrupamento no *swarm* será, então:

$$\frac{e^{K^2\lambda S/\mu} - 1}{K\lambda}. \tag{5.2}$$

Comparando as Equações 5.1 e 5.2, é possível notar que o período em que o conteúdo fica disponível cresce exponencialmente com K, quando todos os arquivos são agrupados e oferecidos em um único *swarm*.

Não é difícil notar que o aumento no tempo em que o conteúdo fica disponível implica na redução da indisponibilidade do conteúdo (fração de tempo em que o

bloco do conteúdo esteve indisponível no *swarm*). O Teorema 3.1 apresentado [25] demonstra que agrupamentos de K arquivos permitem reduzir a indisponibilidade do conteúdo por um fator $e^{\Theta(K^2)}$.

O agrupamento de K arquivos implica, ainda, no aumento do tempo ativo de *download*. Isto é, o tempo necessário para que o usuário recupere todo o conteúdo, se considerarmos uma taxa constante de *download* μ, é superior no caso do *swarm* com agrupamento ($KS/\mu > S/\mu$). No entanto, a depender do tempo que o conteúdo fique indisponível, o tempo total de *download* dos arquivos isolados pode ser superior ao tempo total para recuperar todos os K arquivos agrupados. Por exemplo, *peers* que chegam e encontram o conteúdo indisponível no sistema devem aguardar o retorno do Publisher para concluírem o seu *download*. Assim, se o acréscimo no tempo que o conteúdo fica disponível, causado pelo agrupamento dos K arquivos, for maior que o aumento no tempo ativo de *download*, o agrupamento de arquivos reduz o tempo total de *download*, mesmo aumentando a quantidade efetiva de dados recuperados. Isso também é demonstrado em [25], no Teorema 3.2.

5.3.3 Experimentos

As conclusões obtidas a partir do modelo $M/G/\infty$ do BitTorrent são no mínimo intrigantes: em *swarms* muito populares, os Leechers podem recuperar mais dados em menos tempo. O que será apresentado nessa subseção é uma série de experimentos, realizados na Internet, envolvendo máquinas do PlanetLAB, com o objetivo de validar os resultados sugeridos pelo modelo em questão. Os experimentos são também utilizados para analisar a prática do agrupamento de arquivos quando as suposições do modelo não são válidas, como, por exemplo, para um processo de chegadas de Leechers diferente de Poisson.

Detalhes dos experimentos

Os experimentos foram realizados utilizando 200 nós do PlanetLAB (de um total de aproximadamente 1000 máquinas disponibilizadas pelo ambiente), selecionadas a partir de medições prévias de estabilidade e desempenho. Uma máquina localizada na UMass-Amherst foi utilizada como controlador do experimento e outra como o Tracker dos *swarms*. Os experimentos consistem na criação de *swarms* privados,

i.e., os torrents não são publicados em sites de divulgação. Com isso, garante-se que apenas máquinas envolvidas no experimento estariam conectadas ao *swarm*.

O controlador mantém uma lista de eventos a serem executados no experimento: (i) ação (chegada dos Leechers, chegada ou partida do Publisher); (ii) instante de ocorrência do evento; e, (iii) nome da máquina. Na ocorrência do evento, o controlador dispara, via *ssh* , o comando apropriado para iniciar a aplicação cliente BitTorrent instalada nas máquinas do PlanetLAB. A aplicação cliente BitTorrent 4.0.2, desenvolvido por Legout et al. [114], foi escolhida por tratar-se de uma versão instrumentada, que permite a geração de logs de eventos da ferramenta, tais como blocos enviados e recebidos pelos *peers*, conteúdo das mensagens de controle e os mapas de bits recebidos dos vizinhos. Ao final do experimento, o controlador recupera os logs armazenados nas máquinas do PlanetLAB, de onde as métricas de interesse são estimadas.

Os parâmetros dos experimentos são os mesmos utilizados nas simulações (Subseção 5.2.1) e no modelo (Subseção 5.3.2). (A Tabela 5.1 sintetizada a descrição de cada um desses parâmetros.) Nos experimentos, os torrents são formados por um único arquivo de tamanho S ou um agrupamento de K arquivos com tamanho total de $S_K = K \cdot S$MB. A chegada dos Leechers ocorre inicialmente por um processo de Poisson, mas, em seguida, são apresentados resultados considerando outros processos de chegada. A taxa de chegada dos Leechers em um *swarm* de arquivo isolado é λ. Já a taxa de chegada de um *swarm* de K arquivos agrupados é a soma das taxas de chegada dos *swarms* isolados, $\Lambda = \sum_{i=1}^{K} = \lambda_i$. Diferentes taxas de *upload* dos *peers* foram consideradas nos experimentos, onde a capacidade dos Leechers é dada por μKBps e dos Publishers pKBps. Os Leechers abandonam o sistema assim que concluem o *download*. Os *swarms* possuem um único Publisher, que alterna entre dois estados: ativo e inativo. O comportamento do Publisher, definido pela distribuição do tempo de permanência em cada um desses estados, variou de acordo com os objetivos experimentais. Intervalos determinísticos e exponenciais foram considerados nos experimentos, com médias de A segundos para o estado ativo e I segundos para o estado inativo. Os valores utilizados para cada um desses parâmetros serão informados no decorrer da descrição dos resultados.

Parâmetro	Descrição
λ	Taxa de chegada dos Leechers (peers/min.)
S	Tamanho do arquivo (Bytes)
μ	Taxa máxima de *upload* definida para os Leechers (KBytes/seg.)
p	Taxa máxima de *upload* definida para o Publisher (KBytes/seg.)
A	Tempo médio que o Publisher permanece ativo (seg.)
I	Tempo médio que o Publisher permanece Inativo (seg.)

Tabela 5.1: Parâmetros dos experimentos.

Sobrevida do *swarm* após partida do Publisher

O primeiro conjunto de experimentos investiga a dinâmica do *swarm*, após a partida do Publisher. Para isso, foi considerado um Publisher que chega ao sistema no instante de tempo 0, aguarda a chegada do primeiro Leecher, o que ocorre no instante de tempo t_1, permanece ativo servindo os *peers* do *swarm* e fica inativo tão logo o primeiro Leecher conclua o *download* do conteúdo no instante $t_1 + g_1$, onde g_1 é o tempo necessário para o primeiro Leecher ser servido pelo sistema. Os Leechers também saem do sistema, logo após concluírem o *download*. Um total de 100 Leechers foram considerados nesse primeiro experimento e os parâmetros utilizados foram: $\lambda = 1/150$ peers/seg., $S = 4$MB, $\mu = 33$KBps, $p = 50$KBps.

A Figura 5.8 ilustra a dinâmica do *swarm* durante três diferentes rodadas de experimento. O eixo Y representa o identificador do Leecher e o eixo X representa o tempo do experimento. O período de permanência do Publisher no sistema também é indicado no gráfico. Cada segmento de linha começa no instante em que o *peer* se conecta ao *swarm* e termina quando ele deixa o sistema.

A Figura 5.8(A) representa a dinâmica de um *swarm* de arquivo isolado ($K = 1$). Na figura é possível observar que apenas um único Leecher (com identificador 1) foi capaz de concluir o *download* do arquivo. Todos os demais Leechers permaneceram bloqueados no sistema, após a saída do Publisher juntamente com o primeiro Leecher.

Por outro lado, no *swarm* com $K = 10$ arquivos agrupados, a situação é invertida, como mostra a Figura 5.8(B). Neste caso, apenas um único Leecher (com identificador 98) não foi capaz de concluir o download do conteúdo ao final do exper-

imento. Isso acontece porque, após terminar de servir o bloco final ao último Leecher a deixar o sistema, este *peer* ficou sozinho no *swarm*, sem ter de quem receber as partes restantes para concluir o *download* do conteúdo. Esse resultado indica que, quando $K = 10$, o *swarm* tem uma grande sobrevida, mesmo sem a presença de um Publisher. Conclusão semelhante ao que é sugerido pelo modelo $M/G/\infty$ descrito na Seção 5.3.2. Mais adiante, na Seção 5.4, é apresentada uma análise experimental mais detalhada para esse tipo de *swarm*, denominados de auto-sustentáveis.

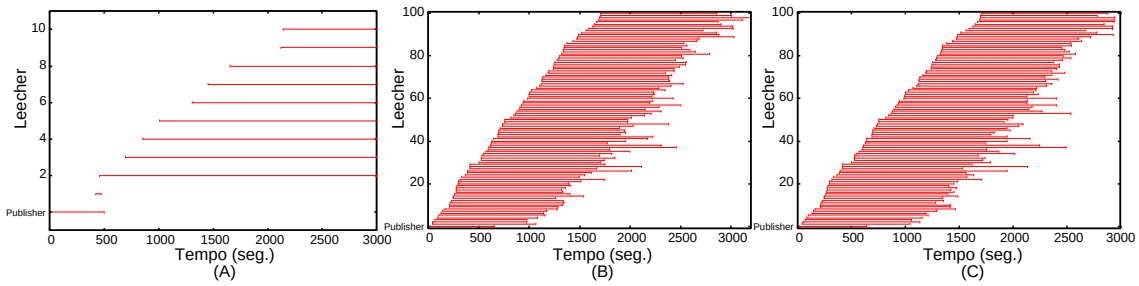


Figura 5.8: Dinâmica do *swarm* em três diferentes configurações de experimentos: (A) K=1; (B) K=10, sem tempo de espera; e, (C) K=10, com tempo de espera.

Uma terceira configuração foi utilizada para esses experimentos. O objetivo foi analisar o que aconteceria ao *swarm*, em especial ao último *peer*, se os *Leechers* continuassem no sistema por algum tempo atuando como Seeder. Este seria o caso, por exemplo, se os *peers* tivessem incentivos para permanecer no sistema, mesmo depois de completarem os seus downloads, ou fossem de alguma forma forçados a isso até que a razão (total de *upload*)/(total de *download*) fosse igual a 1. Embora tal incentivo ou exigência não ocorra atualmente no BitTorrent, existe uma racionalidade real para essa hipótese. Não é difícil imaginar que, em geral, há um intervalo de tempo entre a conclusão do *download* e a intervenção do usuários para finalizar a aplicação BitTorrent. Dessa forma, os *peers* permaneceriam por um período de tempo atuando como Seeder no sistema. A questão analisada aqui é, será que neste caso o último Leecher é capaz de concluir o *download*? O resultado do experimento mostrado na Figura 5.8(C) indica que a resposta é sim. Se os Leechers, depois de concluírem o *download*, permanecerem no sistema por um tempo exponencial com média de apenas 40 segundos (3% do tempo médio de download do experimento (B)), então também o último Leecher consegue concluir o *download* do conteúdo.

A Figura 5.9 mostra a taxa média agregada de *download* no *swarm* (eixo Y)

em função do tempo de experimento (eixo X). O gráfico ainda mostra com pontos os instantes de chegada dos Leechers ao sistema. Observamos que, após a saída do Publisher do sistema (após aproximadamente 600 segundos de experimento), a taxa média agregada de download varia em torno de 33 KBps, que equivale à capacidade de *upload* (μ) definida para os Leechers no experimento. Isso indica que o BitTorrent é extremamente eficiente na divulgação do conteúdo e é capaz de saturar a capacidade de *upload* dos seus *peers*.

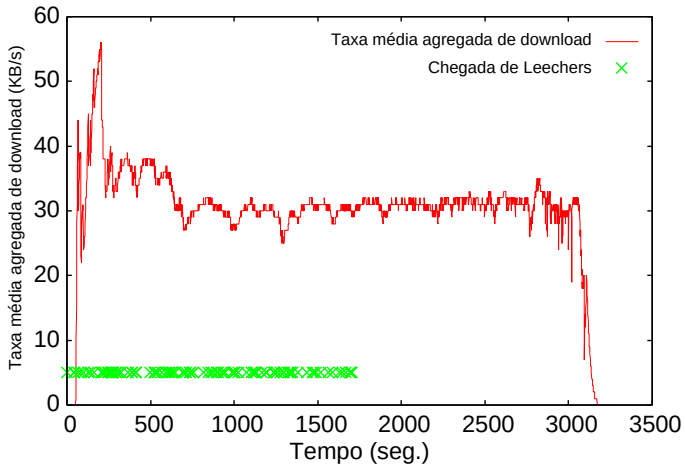


Figura 5.9: Taxa média de *download* agregada dos *peers* durante o funcionamento do *swarm*.

Pelos resultados apresentados nas Figuras 5.8(B) e (C), é possível notar a seguinte característica na progressão dos *peers* dentro dos *swarms*: Leechers que chegam próximos um do outro tendem também a terminar o *download* em instantes próximos. Isso é uma característica importante para o sistema, já que é desejável evitar rupturas do *swarm*. Considere, por exemplo, o caso extremo em que todos os *peers* dedicam as suas capacidades de *upload* para servir um único Leecher (digamos, L^a) e esse Leecher retribuí a generosidade de apenas um dos vizinhos (e.g., L^b). Neste cenário, se L^a e L^b concluírem seus respectivos *downloads* e saírem do sistema em seguida, partes do conteúdo podem ficar indisponíveis no *swarm*. Os resultados experimentais demonstram que esse tipo de distorção não ocorre no BitTorrent, uma vez que as progressões dos *peers* apresentadas nas Figuras 5.8(B) e (C) se mostraram semelhantes ao longo de toda a vida do *swarm* e a taxa média agregada de *download* é mantida quase constante por todo o tempo de experimento, como mostra a Figura 5.9.

A última análise feita para essa primeira configuração de experimentos tem como objetivo analisar o serviço do sistema, durante a sobrevida do *swarm*, para diferentes tamanhos de agregações de arquivos. O gráfico ilustrado na Figura 5.10 representa o número de Leechers servidos (eixo Y) entre os instantes de tempo 0 e 1500 segundos de experimento (eixo X), para K =1, 2, 4, 6, 8 e 10. No extremo esquerdo do gráfico (para X variando entre 0 e 300 segundos de experimento), nenhum Leecher conclui o *download*. Nesse período, o Publisher ainda está aguardando a chegada dos primeiros Leechers ou fazendo o *upload* dos primeiros blocos aos recém-chegados. Após o primeiro Leecher concluir o *download*, no entanto, as curvas para K igual a 1, 2 e 4 apresentam uma tendência muito diferente, em comparação à curvas para K igual a 6, 8 e 10. Isso porque, para os menores valores de K, após o primeiro Leecher ser servido e sair juntamente com o Publisher do sistema, partes do conteúdo se tornam indisponíveis e nenhum outro Leecher consegue concluir o *download*. Por outro lado, para os valores maiores de K, o número de Leechers servidos aumenta linearmente em função do tempo de experimento.

Considerando o extremo direito do gráfico da Figura 5.10 (para X igual a 1500 segundos de experimento), é possível notar que quanto maior for o valor de K, menos será o número total de Leechers servidos até esse instante do experimento. Esse resultado sugere que existe um delicado *trade-off*, que deve ser considerado para a escolha do valor ideal de K. O número de arquivos agregados deve ser grande o suficiente para que o *swarm* tenha alta disponibilidade, no entanto, valores muito grandes podem afetar o desempenho do usuário final com grandes tempos de *downloads*. Esse *trade-off* será discutido em mais detalhes a seguir.

Agrupamento de arquivos reduz o tempo total de *download*

Na segunda sessão de experimentos, foi considerado um *swarm* com Publisher intermitente. Durante todo o tempo de experimento, o Publisher alterna entre períodos de atividade e inatividade. Cada um dos c ciclos é formado por um período de atividade, seguido por um período de inatividade, cujos tempos de duração desses períodos são determinísticos e iguais a $A = 600$ e $I = 1800$ segundos, respectivamente. Foram considerando, ainda, os seguintes parâmetros: $\lambda = 1/60$ peers/seg, $S = 4$MB, $\mu = 50$KBps, $p = 50$KBps, $K = 1, .., 8$. Para cada valor de K, foi exe-

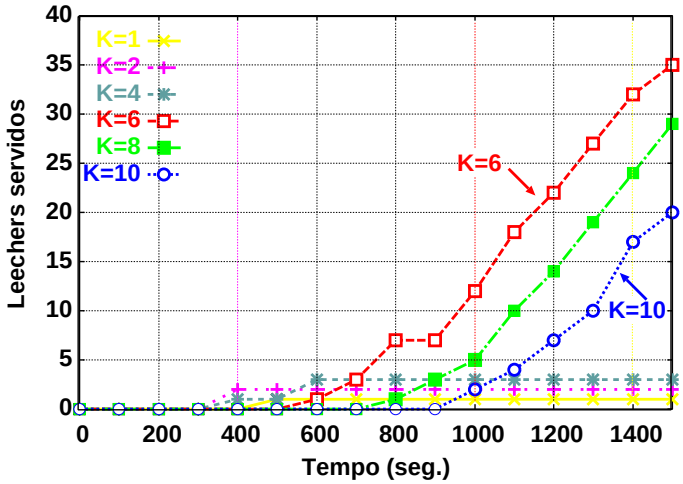


Figura 5.10: Número de Leechers servidos, para diferentes tamanhos de agrupamento.

cutado um experimento com $c = 2.5$ ciclos, ou seja, ocorreu uma sequência de três períodos de atividade do Publisher, intercalados por dois períodos de inatividade.

A Figura 5.11 ilustra a dinâmica dos *swarms* em experimentos para três dos oito valores de K. Na Figura 5.11(A), que mostra os resultados para $K = 1$, é possível observar que muitos Leechers concluem o *download* aproximadamente no mesmo instante (por exemplo, aproximadamente 40 Leechers deixam o *swarm* em torno do instante 2400). Isso ocorre porque, em algum momento antes do conteúdo se tornar indisponível no *swarm*, os Leechers ficam "bloqueados" à espera do retorno do Publisher para completarem os seus *downloads*. Na Figura 5.11(B)(K = 4), por outro lado, o bloqueio acontece apenas uma vez, e por um pequeno período de tempo. Na Figura 5.11(C) é possível notar que não ocorrem bloqueios para o experimento com $K = 5$. O mesmo aconteceu com os demais experimentos realizados para valores de $K > 5$, não mostrados em gráficos.

O fato de os Leechers não ficarem bloqueados quando $K \geq 5$, por si só, já é uma propriedade positiva para os usuários. Isso representa uma alta disponibilidade dos blocos no sistema e evita que usuários se sintam desmotivados em continuar conectados ao *swarm*, por não observarem uma evolução no processo de recuperação do conteúdo. No entanto, pode-se argumentar que, um usuário não está muito interessado em saber se ele está bloqueado ou não. Para o usuário final, o que importa mesmo é o desempenho do sistema, isto é, o tempo de download de um arquivo.

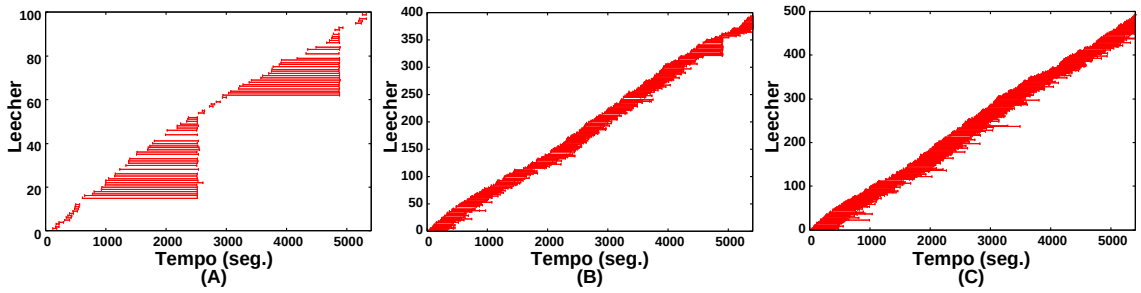


Figura 5.11: Dinâmica do *swarm* com um Publisher intermitente e ciclos determinísticos: (A) $K = 1$; (B) $K = 4$; e, (C) $K = 5$

As médias dos tempos totais de *download* computadas dos experimentos, com K variando de 1 a 8 arquivos no *swarm*, são ilustradas na Figura 5.12(A). Fica claro no gráfico o *trade-off* existente na escolha do valor de K. Para valores de $K < 4$, as médias computadas para os tempos totais de *download* são fortemente influenciadas pelos tempos de bloqueio dos Leechers. Isso porque, a probabilidade dos Leechers ficarem bloqueados no *swarm* é significativa. No entanto, para valores de $K \geq 4$, as chances do conteúdo ficar indisponível reduz significativamente e o tamanho do arquivo passa a ser o fator dominante no tempo total de *download* do conteúdo. À medida que K cresce, o tempo médio para que os Leechers recuperem todo o conteúdo passa a crescer linearmente em função do tamanho do conteúdo. Portanto, $K = 4$ é o valor ótimo do tamanho da agregação para o cenário experimentado.

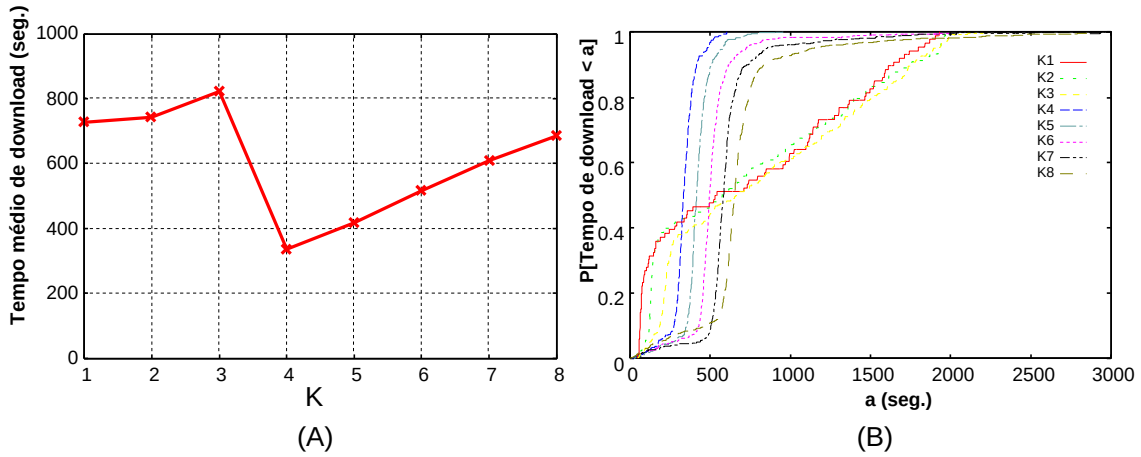


Figura 5.12: Tempos totais de *download* para $K = 1, \ldots, 8$: (A) Média; (B) Distribuição.

As distribuições (CDF's) dos tempos de *download* para cada rodada do experimento são mostradas na Figura 5.12(B). É possível notar que existem dois com-

portamentos distintos para as curvas mostradas no gráfico. As curvas referentes aos experimentos com $K = 1, 2, 3$ podem ser divididas em duas regiões. A primeira região ($Tempo\ de\ download < 100$ seg.) representa os Leechers que, em nenhum momento do processo de recuperação do conteúdo, foram afetados pela indisponibilidade de blocos no *swarm*. Em geral, esse *peers* chegaram ao sistema, encontraram o conteúdo disponível e concluíram o *download* antes que alguma parte dele se tornasse indisponível. A segunda região ($Tempo\ de\ download > 100$ seg.), o *download* representa os Leechers que em algum momento da recuperação do conteúdo tiveram seus *downloads* bloqueados. Já nas curvas referentes aos experimentos com $K = 4, \ldots, 8$, é possível observar que o tempo total de *download* da grande maioria dos Leechers (cerca de 90%) é aproximadamente igual a $(K \cdot S)/\mu$. Isso porque, se o conteúdo está sempre disponível, o tempo necessário para o *peer* recuperar o conteúdo é proporcional ao tamanho do conteúdo $(K \cdot S)$ e sua capacidade (μ).

Períodos exponenciais de Intermitência

Os resultados apresentados acima são referentes a um Publisher intermitente, mas com um comportamento bastante previsível, uma vez que os intervalos dos períodos de atividade e inatividade eram determinísticos. O que será analisado a seguir é o comportamento da dinâmica do *swarm*, quando os períodos de intermitência não são determinísticos, como no experimento anterior, mas sim exponenciais. O número de ciclos definido para cada rodada dos experimentos é também maior.

Nos experimentos foram considerados $c = 10$ ciclos de operação de um Publisher de capacidade $p = 100$KBps, que alternou entre intervalos de atividade e inatividade, exponencialmente distribuídos, com médias $A = 300$s e $I = 900$s, respectivamente. A chegada dos Leechers segue um processo de Poisson com taxa $\lambda = 1/60$ peers/seg. e a capacidade desses Leechers é de $\mu = 50$KBps. Os arquivos têm $S = 4$MB e o tamanho dos agrupamentos variou de $K = 1, \ldots, 8$ arquivos.

As dinâmicas de alguns dos *swarms* estão ilustradas nas Figuras 5.13(A)-(D). O gráfico (A) mostra o resultado para um agrupamento com $K = 2$ arquivos. Assim como foi visto nos gráficos dos experimentos para um Publisher intermitente com períodos determinísticos, aqui também podemos observar diversos "bloqueios" na progressão dos Leechers e o efeito de partidas em rajada. Isso sugere que o *swarm*

com $K = 2$ não é auto-sustentável. Leechers frequentemente têm que esperar o retorno do Publisher a fim de concluírem seus *downloads*. No caso em que $K = 3$, são bem menores as ocorrências de "bloqueios" dos Leechers, como mostrado na Figura 5.13(B). Quando $K \geq 4$ não há bloqueios, como pode ser visto nas Figuras 5.13(C)($K = 4$) e (D)($K = 5$).

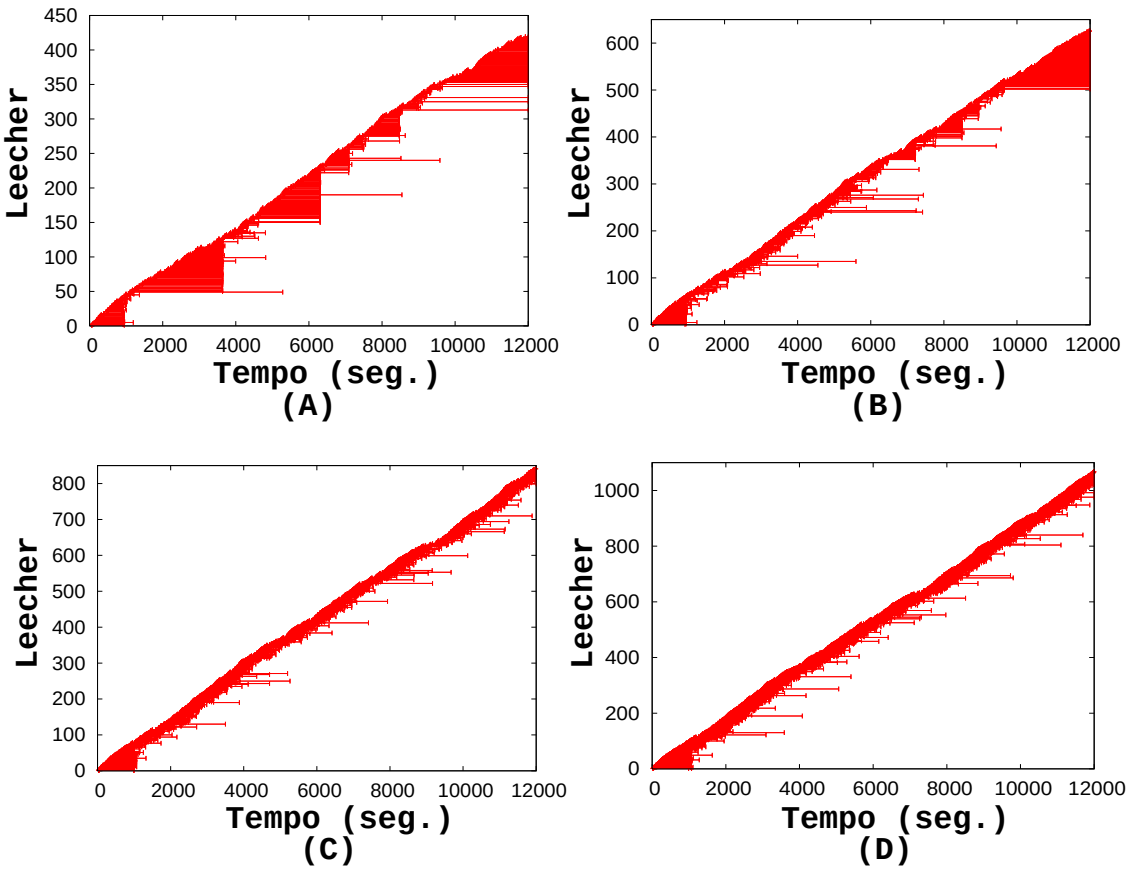


Figura 5.13: Dinâmica do *swarm* com um Publisher intermitente e ciclos exponenciais: (A) $K = 2$; (B) $K = 3$; (C) $K = 4$; e, (D) $K = 5$.

A Figura 5.14 apresenta a média e os percentis da distribuição do tempo total de *download* (eixo Y) em função do tamanho de K. Considerando os experimentos com $K = 1$ e 2, é possível observar que as médias do tempo total de *download* são altas. Os períodos de indisponibilidade dos Publishers e a baixa popularidade dos *swarms* exercem grande impacto nesses valores. Quando $K = 3$, a média tem uma redução significativa. No entanto, assim como no caso de $K < 3$, a variabilidade é ainda muito alta, uma vez que existe uma possibilidade não desprezível do conteúdo ficar indisponível e os Leechers terem que esperar pelo retorno do Pub-

lisher para concluírem seus *downloads*. O tamanho ótimo da agregação é $k = 4$, a média e a mediana são as menores entre todos os valores experimentados. Nesse ponto, a variância também diminui, sugerindo que neste caso o *swarm* independe da disponibilidade do Publisher. A partir dos experimentos com valores de $K \geq 5$, os tempos totais de *download* são dominados pelo tamanho de K e as médias passam crescer proporcionalmente ao tamanho do agrupamento. Já a variabilidade permanece baixa, uma vez que o *swarm* se mantém auto-sustentável com o aumento de K.

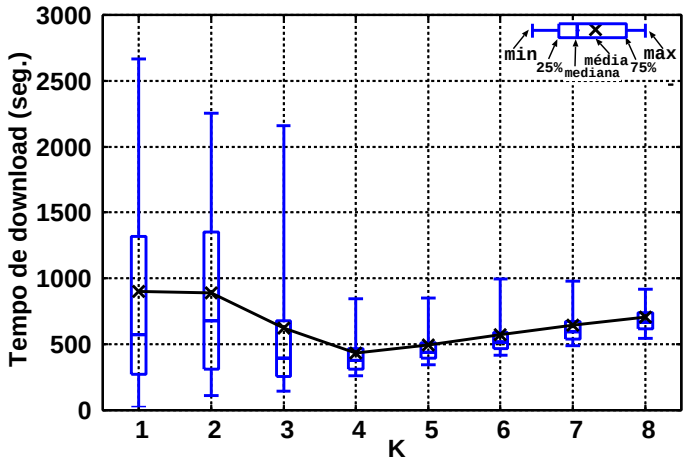


Figura 5.14: Distribuição do tempo total de *download*.

Capacidades de *upload* heterogêneas

O mesmo experimento descrito acima foi repetido, considerando desta vez diferentes capacidades de *upload* para os Leechers do sistema. Os valores de μ são agora definidos por uma distribuição de capacidade com média e mediana iguais a 280KBps e 50KBps, respectivamente. Essa distribuição foi estimadas através de medições de *swarms* reais do BitTorrent realizadas por Piatek et al. para o trabalho do BitTyrant[133][1].

O objetivo desses experimentos foi analisar os impactos de heterogeneidade das taxas de *upload* nos tempos totais de *download*. Os resultados obtidos para $K = 1, \ldots, 8$ são apresentados na Figura 5.15. Comparando com os resultados anteriores (com $\mu = 50$KBps), é possível notar que não existe uma alteração qualitativa no comportamento do sistema. Mas, é possível verificar uma diferença no

[1]Os autores de [133] gentilmente cederam os *traces* para serem utilizados nos experimentos.

tamanho ótimo para o número de arquivos agregados e um aumento na variância da distribuição dos tempos totais de *download*. A diferença no tamanho ótimo da agregação com $K = 5$ é justificada pelo aumento da capacidade dos Leechers utilizados no experimento com taxas de *upload* heterogêneas. Já o aumento na variância da distribuição do tempo total de *download* é causada pela variação das taxas de *download* atribuída aos Leechers.

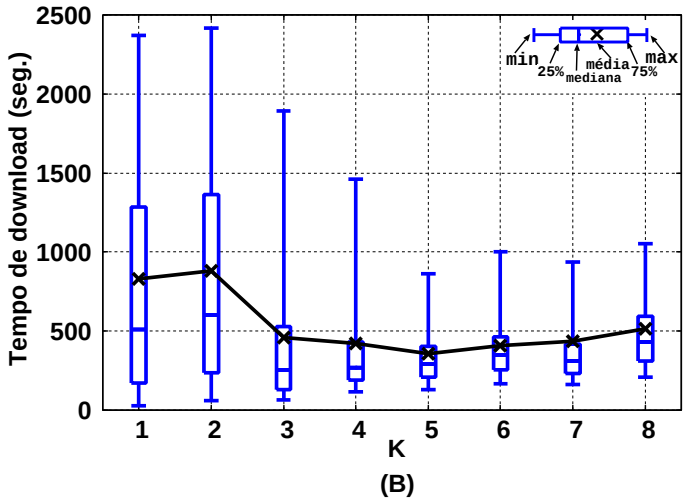


Figura 5.15: Distribuição do tempo total de *download* considerando *peers* com capacidades heterogêneas.

Popularidades heterogêneas

Os últimos experimentos desta seção consideram o caso em que diferentes arquivos podem ter popularidades distintas. O objetivo é compreender como e quando a agregação pode ajudar aos usuários neste caso. Foram analisados resultados de experimentos executados em dois cenários distintos. No primeiro foi considerado um conjunto de 4 arquivos cuja distribuição da popularidade entre eles segue uma lei de potência: $\lambda_1 = 1/8$, $\lambda_2 = 1/16$, $\lambda_3 = 1/24$ e $\lambda_4 = 1/32$. (A popularidade do agrupamento é $\Lambda = \sum_{i=1}^{4} \lambda_i = 1/3.84$.) Neste primeiro caso foi considerado o processo de chegada Poisson. Já no segundo cenário, foram considerado 2 arquivos cujo processo de chegada dos Leechers foi definido por *traces* coletados em dois *swarms* reais do BitTorrent.

No primeiro cenário, foram executados cinco rodadas de experimentos: experimentos 1, 2, 3 e 4 com arquivos isolados ($K = 1$) e o experimento 5 considerando

a agregação dos quatro arquivos ($K = 4$). A taxa de chegada dos *peers* utilizada no experimento i (para $i = 1, 2, 3, 4$) foi λ_i e no experimento 5 foi Λ, conforme definido no parágrafo anterior. Todos os demais parâmetros foram mantidos dos experimentos de períodos exponenciais de intermitência ($p = 100$KBps, $\mu = 50$KBps, $S = 4$MB, $c = 10$, $A = 300$s e $I = 900$s).

A média e a distribuição do tempo total de *download* em cada rodada do experimento são mostradas na Figura 5.16(A). Considerando apenas os arquivos isolados ($K = 1$), à medida que o índice do experimento cresce (isto é, à medida que a popularidade dos arquivos diminui), a média do tempo total de *download* aumenta. Comparando esses valores ao obtido com o agrupamento de arquivos ($K = 4$, experimento 5), verifica-se que apenas o experimento 1 apresenta uma média inferior ao experimento com agregação. Todos os demais experimentos (2, 3 e 4) têm médias superiores, mesmo se tratando de arquivos 4 vezes menores. Os resultados dos experimentos demonstram ainda que, embora a agregação de arquivos com diferentes popularidades possa ocasionar um aumento no tempo total de *download* do arquivo mais popular, isso pode trazer benefícios significativos no desempenho para os usuários interessados nos arquivos menos populares.

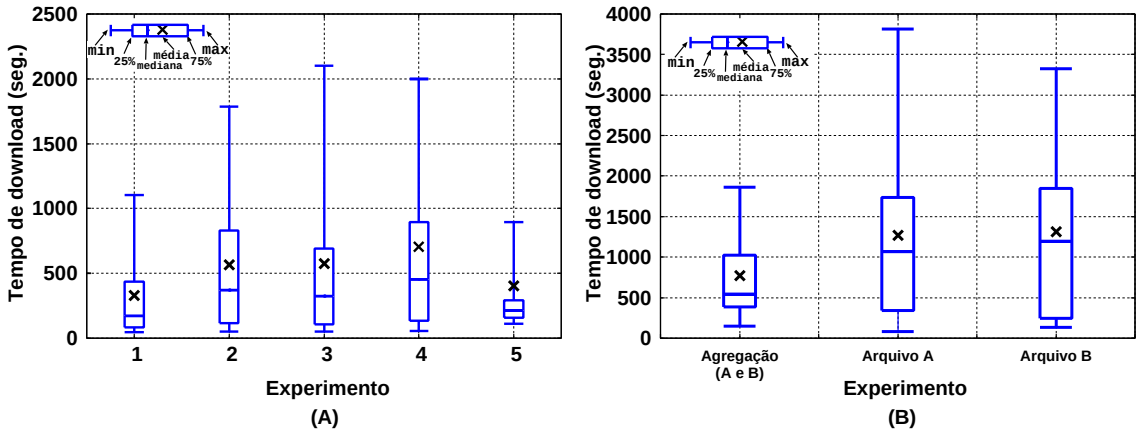


Figura 5.16: Distribuição do tempo total de *download* considerando conteúdos de popularidades heterogêneas.

Para os experimentos do segundo cenário, foram utilizados os *traces* do processos de chegada em dois *swarms* reais do BitTorrent. Os arquivos (denominados A e B) oferecidos pelos *swarms* são trechos complementares dos melhores momentos da cerimônia de abertura dos Jogos Olímpicos de 2008. Esses *swarms* foram criados

no dia seguinte à realização do evento e a popularidade deles foi monitorada ininterruptamente, nas 12 primeiras horas de vida. Os arquivos isolados têm tamanho iguais ($S_A = S_B = 10$MB). O tamanho do arquivo agregado (chamado de AB) é igual a soma dos dois arquivos isolados ($S_{AB} = 20$MB). O processo de chegada utilizado para o arquivo AB também é a soma dos dois processos isolados. O experimento considerou ainda o Publisher com capacidade $p = 100$KBps e períodos de intermitência exponenciais com médias $A = 500$ e $I = 1500$ segundos, e Leechers com capacidade $\mu = 50$KBps. Para cada um dos três arquivos, uma rodada de experimento com duração de 12 horas foi executada.

A média e a distribuição dos tempos totais de *download* computadas nos experimentos é mostrada na Figura 5.16(B). A redução da média do tempo total de *download* para a agregação, em relação aos valores dos arquivos isolados A e B, são de 39% e de 41%, respectivamente. A variância no caso dos arquivos agregados é também significativamente menor do que nos casos de arquivos isolados. Esses resultados demonstram que as implicações da agregação de arquivos são válidas inclusive para processos de chegada diferentes de Poisson.

5.4 Redução de custo para distribuição de conteúdo

Empresas de comércio eletrônico para mídia digital têm se deparado com uma crescente demanda por títulos que isoladamente não são considerados sucessos de vendas, mas juntos representam um montante significativo do total de arrecadação da empresa. Por exemplo, análises apresentadas em [134, 135] indicam que 57% dos produtos vendidos pela Amazon.com, no ano de 2004, foram de produtos que não estavam disponíveis em lojas tradicionais; 15% do total da demanda de filmes no Netflix são por títulos fora da lista dos 3 mil mais solicitados; e, 20% da receita da Rhapsody, em Novembro de 2008, foram geradas pela venda de músicas que não figuravam na lista das 52 mil mídias mais populares do site. Portanto, essas e outras empresas devem considerar o fato de que conteúdos impopulares podem desempenhar um papel fundamental no seu faturamento. Nesse contexto, provedores comerciais de disseminação de conteúdo devem passar a dedicar uma atenção

especial (e uma parcela significativa do seus recursos) para a distribuição de mídias impopulares.

A possibilidade de disseminar conteúdos populares a baixo custo faz com que provedores comerciais estejam cada vez mais interessados em integrar soluções baseadas em *swarms* P2P aos seus serviços tradicionais de distribuição. Dessa forma, os recursos economizados pelo provedor poderiam ser utilizados para servir clientes de conteúdos impopulares. Os resultados de simulação analisados na Seção 5.2.2, por exemplo, sugerem que o uso da arquitetura BitTorrent pode ser bastante conveniente neste cenário, se o Publisher operar alternando entre estados de atividade e inatividade. Assim, é fundamental que os provedores criem mecanismos que permitam definir a forma mais adequada para a utilização dos recursos, segundo os objetivos e restrições desejadas pela empresa. Um opção promissora é a utilização de mecanismos que reduzam o consumo da largura de banda do Publisher para a disseminação de conteúdo, sem afetar a qualidade do desempenho experimentado pelos usuários.

O que apresentaremos a seguir é uma série de experimentos realizados com *swarms* privados do BitTorrent que comprovam os benefícios do uso de mecanismos estratégicos para atuação do Publisher. Os resultados demonstram que *swarms* populares podem ser auto-sustentáveis por longos períodos de tempo e, neste caso, é possível reduzir o consumo da largura de banda dos Publishers a (quase) zero. Experimentos foram realizados para analisar em que condições os *swarms* são auto-sustentáveis, os impactos dessa estratégia do Publisher em termos de redução da largura de banda e atraso experimentado pelos usuários do sistema.

Os resultados apresentados nesta seção foram obtidos de experimentos realizados na Internet, envolvendo máquinas do PlanetLAB, e em um cluster da UMass-Amherst formado por 25 máquinas interconectadas pela rede local da universidade. Diferentes configurações foram usadas. Os parâmetros são semelhantes aos utilizados nas simulações e nos experimentos descritos nas seções anteriores. (Vide Tabela 5.1, na Seção 5.3.3, para relembrar a descrição dos parâmetros mais relevantes para os experimentos desta seção.) Os valores atribuídos a cada um dos parâmetros são informados no decorrer do texto, juntamente com a descrição dos cenários experimentados.

Análise experimental de *swarms* auto-sustentáveis

Em linhas gerais, o regime operacional de um *swarm* BitTorrent pode ser classificado como: (i) Impopular, quando o número médio de *peers* conectados ao sistema é tão pequeno que todos os Leechers fazem o *download* do conteúdo diretamente do Publisher. (Nesse regime, o desempenho dos usuários no sistema é altamente dependente da capacidade do Publisher e a operação é semelhante a de um serviço cliente/servidor.); (ii) Auto-sustentável, neste caso, se o Publisher for desligado, há grandes chances dos blocos permanecerem disponíveis por um longo tempo e os Leechers podem, ainda assim, concluírem com êxito os seus *downloads*, apesar da ausência de qualquer outro Seeder no *swarm*; e, (iii) Intermediário, quando Leechers, ocasionalmente, dependem de Publishers para recuperar blocos do conteúdo que ficaram indisponíveis, mas boa parte do tempo dependem apenas de outros Leechers para concluírem os seus respectivos *downloads* do conteúdo.

Os resultados de simulação apresentados na Seção 5.2.2 demonstram que há um crescimento da disponibilidade dos blocos entre os Leechers do *swarm* com o aumento da popularidade do conteúdo. No entanto, embora a métrica de disponibilidade utilizada na ocasião levasse em consideração apenas os blocos distribuídos por entre os Leechers do sistema, durante todo o tempo o Publisher permaneceu conectado e servindo blocos no *swarm*. Já nos experimentos apresentados da análise de agrupamento de arquivos (Subseção 5.3), embora o Publisher permanecesse desconectado do *swarm*, ele retornava ao sistema após curtos períodos de inatividade. Assim, duas questões fundamentais são: (i) seria possível um *swarm* sobreviver por longos períodos de tempo, sem a presença de Publishers ou Seeders? e, (ii) quais as condições necessárias para que isso ocorra? Essas são questões fundamentais para auxiliar na definição de estratégias de redução do custo na disseminação de conteúdos via *swarms* BitTorrent.

Experimentos em larga escala foram realizados justamente para responder essas questões e ajudar a compreender melhor as condições em que um *swarm* se torna auto-sustentável, em função da popularidade de seu conteúdo. O objetivo do experimento é estimar a distribuição do tempo de sobrevida do *swarm*, após a partida do Publisher. Portanto, esse tempo de sobrevida do *swarm* representa o tempo em que todos os blocos do conteúdo permanecem disponíveis distribuídos entre os Leechers

do sistema, sem a presença do Publisher. Para cada valor de taxa de chegada (λ), que variou de 1 a 8 peers/min., foram executadas 50 rodadas. À medida que os Leechers concluíam seus respectivos *downloads*, eles abandonavam o sistema, sem se tornarem Seeders. Cada rodada teve início com a chegada de um Publisher e dos primeiros Leechers ao *swarm*, e foi interrompida quando uma das duas condições fosse verdadeira: (i) o número de peers no sistema chegasse ao limite de 100; ou, (ii) o tempo da rodada do experimento chegasse a 10000 segundos, desde a partida do Publisher. As razões para a escolha do limite de 100 *peers* (condição (i)) e do tempo máximo de experimento (condição (ii)) são explicadas a seguir.

Quando todos os blocos estão disponíveis no *swarm*, o número médio de usuários no sistema é dado por $N = \lambda T$, onde T representa o tempo médio que os usuários levam para recuperar o conteúdo (i.e., $T = S/\mu$). Se o número real de *peers* conectados no sistema for muito superior ao valor esperado de N (considerando os valores utilizados no experimento para λ, S e μ), isso indica que os Leechers estão bloqueados no sistema e que a sobrevida do *swarm* chegou ao fim. Assim, devido aos valores dos parâmetros utilizados nesse experimento descritos abaixo, o limite definido para a condição (i) foi de 100 *peers*.

Para analisar a distribuição do tempo de sobrevida do *swarm*, foi preciso repetir os experimentos diversas vezes (neste caso, foram 50 rodadas, para cada valor de taxa de chegada utilizada). Com isso, foi necessário impor um limite máximo para a duração dos experimentos. Isso porque, a depender dos valores definidos para os parâmetros do experimento, existiria uma probabilidade não desprezível de que a sobrevida do *swarm* fosse muito grande. Assim, o tempo máximo de 10000 segundos foi escolhido por se tratar de um valor suficientemente alto para a análise desejada. Esse valor é 125 vezes maior do que o tempo de permanência do Publisher no sistema e do que o tempo necessário para que os Leechers recuperem todos os blocos, como sugerem os demais parâmetros utilizados no experimento descritos abaixo.

No experimento, o Publisher permaneceu conectado no sistema, servido a uma taxa de $p = 100KBps$, por um tempo $A = 80$ segundos. Esse tempo é duas vezes o necessário para fazer o *upload* do conteúdo que tinha tamanho $S = 4$MB. Após sair, o Publisher não mais retornava ao sistema (i.e., $I = \infty$). A capacidade de *download* atribuída aos Leechers foi de $\mu = 50KBps$ e o processo de chegada Poisson. As

distribuições dos tempos de sobrevida dos *swarms*, após a desconexão do Publisher, estão representadas nos gráficos da Figura 5.17.

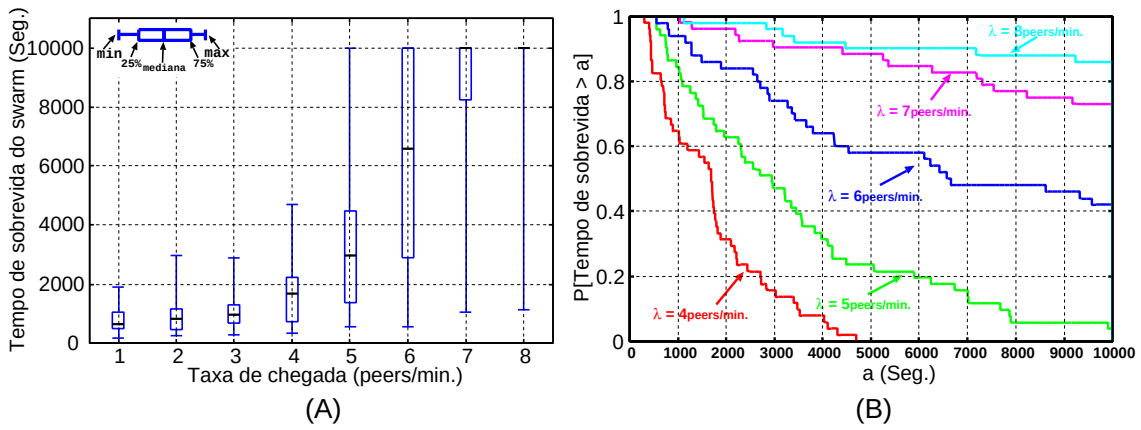


Figura 5.17: Análise dos limites para *swarms* auto-sustentáveis: (A) CDF's dos tempos de sobrevida, para $\lambda = 1, \ldots, 8$; (B) CDF complementar dos tempos de sobrevida, para $\lambda = 4, \ldots, 8$.

A Figura 5.17(A) mostra os percentuais de 25%, 50% e 75% da distribuição do tempo de sobrevida dos *swarm*, bem como os valores mínimos e máximos observados nos experimentos. Nota-se no gráfico que o tempo médio de sobrevida do *swarm* cresce com a popularidade do conteúdo. Para casos de taxas de chegada $\lambda = 5, 6, 7, 8$, houveram *swarms* que sobreviveram até o limite de 10000 segundos. A Figura 5.17(B) mostra a distribuição complementar para $\lambda = 4, 5, 6, 7, 8$. Considerando os resultados para as taxas de chegada maiores, as chances dos *swarms* serem auto-sustentáveis, e com isso os Leechers não ficarem bloqueados antes do final do experimento, são altas. (Para os casos de $\lambda = 6, 7$ e 8, os valores da $P[Tempo\ de\ sobrevida > 10000]$ são, respectivamente, 42%, 74% e 85%.)

Eficiência e economia em *swarms* auto-sustentáveis sem Publisher

Com a finalidade de analisar a eficiência do *swarm*, em termos de desempenho experimentado pelos usuários, e a economia no consumo de banda do provedor de conteúdos populares em *swarms* BitTorrent, experimentos foram realizados considerando dois *swarms* com Publishers atuando de formas distintas. Em um dos *swarms*, o Publisher esteve em atividade durante todo o tempo de experimento ($A = \infty$). No outro *swarm*, o Publisher ficou servindo os Leechers apenas nos

primeiros 80 segundos e depois permaneceu inativo até o final do experimento ($A = 80 \Rightarrow I = \infty$). Os experimentos ocorreram em paralelo e o processo de chegada dos Leechers foi o mesmo, Poisson com taxa $\lambda = 12$ peers/minuto. O valor alto definido para λ foi para garantir que o *swarm* se mantivesse auto-sustentável durante toda duração do experimento. A cada evento do processo de chegada, o controlador do experimento requisitava a uma das máquinas envolvidas no experimento que iniciasse dois processos da aplicação BitTorrent, conectando simultâneamente um novo Leecher a cada um dos dois *swarms*. Os demais parâmetros do experimentos foram semelhantes ao descritos na subseção acima.

As curvas mostradas na Figura 5.18(A) representam as taxas de *upload* servidas por cada Publisher dos dois *swarms* ao longo dos experimentos. No caso em que o Publisher atua estrategicamente(representado no gráfico pela curva azul), verifica-se que a taxa servida cresce rapidamente no início, alcançando os 100KBps definidos como a taxa máxima (p), mas essa taxa reduz a zero logo após os 80 segundos de experimento. Se compararmos com o caso em que o Publisher permanece ativo por todo o experimento (curva vermelha), é possível notar um comportamento semelhante no início do experimento. No entanto, após 80 segundos, a utilização da banda do provedor para o segundo caso é muito superior, pois o Publisher permanece servindo dados a uma taxa que varia entre 20 e 80KBps.

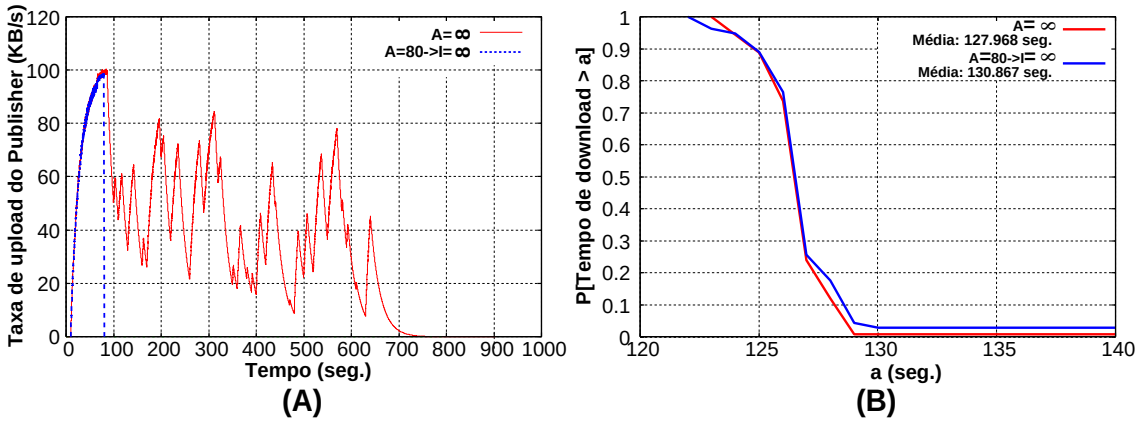


Figura 5.18: Eficiência e economia com Publisher estratégico em *swarms* auto-sustentável.

A economia alcançada pela estratégia de tornar o Publisher inativo quando o *swarm* for auto-sustentável é significativa. No entanto, é desejável que os efeitos dessa estratégia no desempenho experimentado pelos usuários do sistema sejam

pequenos. Para analisar essa questão, a Figura 5.18(B) apresenta a distribuição do tempo de *download* obtido pelos usuários de cada um dos *swarms*, além das médias das distribuições. Os resultados demonstram que não houve uma degradação significativa de desempenho para os usuários.

Nem todo *swarm* é auto-sustentável e, para estes casos, é necessário um esquema mais sofisticado do que simplesmente "desligar" o Publisher. Neste contexto, observa-se a necessidade de desenvolvimento de um controlador que tenha como objetivo definir dinamicamente a taxa máxima de *upload* que um Publisher deve utilizar para servir *swarms* que se encontram operando em regimes críticos (impopular ou intermediário). Possíveis soluções para um algoritmo de controlador estão sendo analisadas e são, sem dúvida, parte dos trabalhos futuros desta tese. Para demonstrar a viabilidade do uso de um controlador para redução do consumo de banda do Publisher, a seção 5.6 discorre sobre um trabalho, ainda em fase preliminar, que está relacionado a essa questão.

5.5 Conclusão

Nesta seção foi apresentada uma análise experimental de larga escala realizada para avaliar o desempenho de protocolos P2P, como o BitTorrent, na disseminação de conteúdo na Internet. Inicialmente, neste capítulo, foi apresenta uma análise sobre as implicações da popularidade do *swarm* na disponibilidade dos blocos e no custo para disseminação do conteúdo pelo BitTorrent. Resultados de simulação demonstram a relação entre o crescimento da popularidade do conteúdo e o aumento da sua disponibilidade entre os Leechers do *swarm*.

Na segunda parta deste capítulo, foi analisada a eficiência da distribuição de arquivos de forma agrupada, ao invés de arquivos isolados. Os resultados apresentados através de uma série de experimentos mostram que é possível aumentar significativamente a disponibilidade de conteúdos impopulares se estes forem oferecidos através de agrupamentos. Ficou demonstrado, inclusive, que em determinadas situações é possível reduzir o tempo de *download* dos arquivos, se eles forem ofertados de forma agrupada. Esses resultados reforçam as suposições apresentadas por um modelo analítico proposto por Menasche et al. em [25].

Na última parte deste capítulo, foi definido e analisado o conceito de *swarms* auto-sustentáveis, os quais têm pouca dependência da presença de um Publisher para que os blocos do conteúdo permaneçam disponíveis por um longo tempo. Resultados de experimentos reais demonstram que é possível reduzir a (quase) zero a banda utilizada pelo provedor para a disseminação de conteúdos muito populares, sem afetar o desempenho experimentado pelos usuários. Para o caso em que o *swarm* não é auto-sustentável, observou-se a necessidade de um esquema mais sofisticado, por exemplo, utilizando um método para automatizar a definição da taxa máxima de *upload* a ser dedicada pelo Publisher a esses *swarms*. A seguir será apresentado um trabalho preliminar nesse sentido.

5.6 Trabalhos preliminares para um controlador de banda dos Publishers de *swarms* em regimes críticos

O objetivo aqui é apresentar uma versão, ainda em fase de estudos, de um controlador que tem como finalidade definir dinamicamente a taxa máxima de *upload* que um Publisher deve dedicar para servir *swarms* que estão operando em um dos regimes críticos (impopular ou intermediário). A razão para incluirmos esta seção no texto da tese é mostrar a viabilidade do uso de um controlador para alcançar economias significativas no consumo de banda de Publishers. Apesar dos estudos estarem em fase preliminar e estejam sendo feitos estudos teóricos para ajudar a alcançar uma solução ótima para o problema, um algoritmo de controlador muito simples já vem sendo testado. Os experimentos realizados com esta versão do algoritmo já indicam que é possível reduzir o consumo de banda do Publisher, sem afetar de forma significativa o desempenho experimentado pelos usuários.

O algoritmo em estudo, neste momento, foi definido empiricamente baseado no conceito fundamental de sistemas P2P: quanto mais *peers* conectados, maior a capacidade agregada do sistema. Assim, a capacidade oferecida pelo Publisher ao *swarm* deve ser reduzida à medida que aumenta o número de Leechers com potencial para contribuir ativamente com o sistema.

O procedimento definido para o controlador é simples. Em um determinado instante (t), o controlador utiliza a Equação 5.3 para determinar $B(t)$, que representa a taxa máxima de *upload* a ser oferecida pelo Publisher ao *swarm* pelos próximos w segundos. O valor determinado para $B(t)$ é uma fração do limite superior da taxa máxima de *upload*, representada aqui por pKBps, e deve ser recomputado a cada w segundos. A racionalidade para o cálculo de $B(t)$ é a seguinte: Leechers recém chegados (chamados de imaturos) têm muito pouco (ou ainda nada) do conteúdo a oferecer aos demais *peers* do sistema, em contrapartida, Leechers que já recuperaram alguns blocos (denominados maduros) são semeadores em potencial destas partes. Logo, o valor definido para $B(t)$ com a Equação 5.3 é diretamente proporcional à fração do número de Leechers imaturos em relação ao número total de Leechers existentes no sistema.

$$B(t) = p * \max([1 - \frac{(N(t) - a(t) - 1)^2}{N(t)^2}], 0.20) \tag{5.3}$$

onde, p é o limite superior da taxa máxima de *upload* definido para o Publisher; N(t) é o número total de *peers* conectados ao sistema naquele instante, incluindo o Publisher; e, a(t) é o número de *peers* imaturos existentes atualmente no sistema, determinado conforme é descrito a seguir.

A cada intervalo de w segundos, o controlador consulta junto ao Tracker as informações de quantos *peers* chegaram ao *swarm* e quantos partiram, desde o início da operação até o instante atual (t). Considere $C(t)$ e $D(t)$ como sendo, respectivamente, os números totais de *peers* que chegaram e partiram do sistema até o instante de tempo t. Assim, o número de *peers* conectados ao *swarm* no instante t, definido como $N(t)$, é dado por $N(t) = C(t) - D(t)$. Se assumirmos $c(t-w,t)$ e $d(t-w,t)$ como sendo, respectivamente, os totais de chegadas e partidas ocorridas no intervalo $[t-w,t)$, então esses valores podem ser obtidos da seguinte forma: $c(t-w,t) = C(t) - C(t-w)$ e $d(t-w,t) = D(t) - D(t-w)$. A Figura 5.19 ilustra os valores computados para $C(t)$, $D(t)$, $c(t-w,t)$ e $d(t-w,t)$, em diferentes instantes de tempo da operação do controlador em um *swarm*.

O cálculo de $a(t)$ (número de *peers* imaturos existentes no sistema no instante de tempo t) depende da ocorrência de partidas nos intervalos anteriores. Seja $l(t)$ o número de intervalos de w segundos, anteriores ao instante de tempo atual t, em que

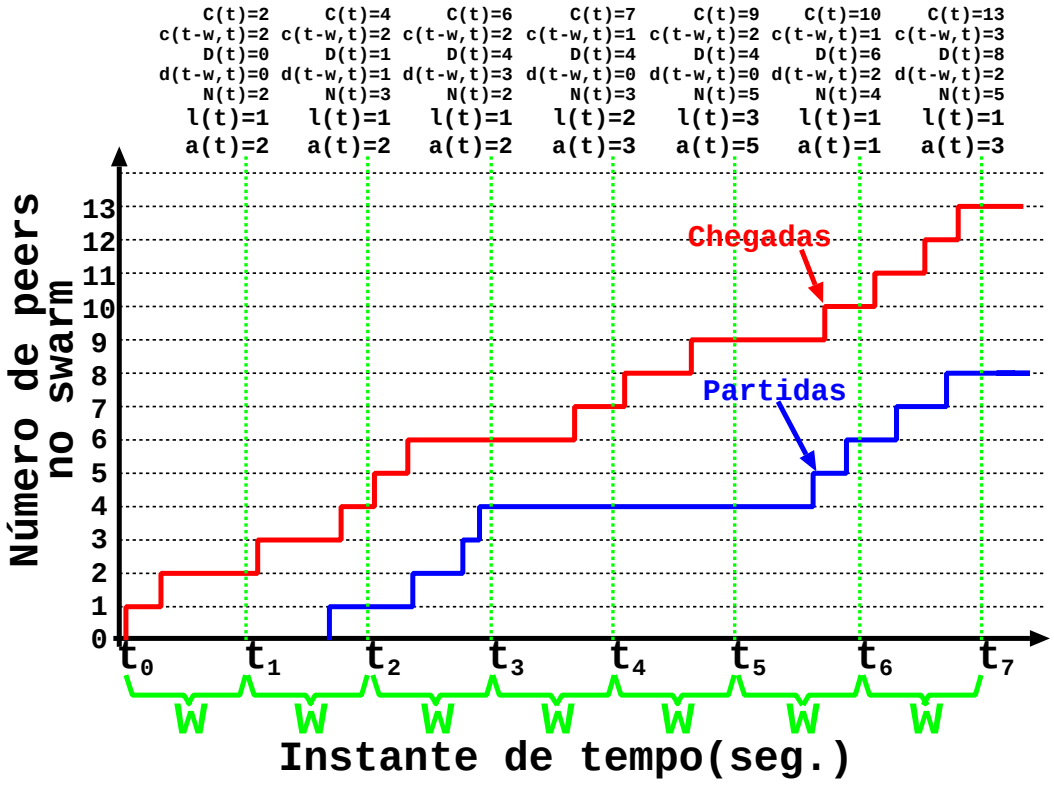


Figura 5.19: Processo de chegada e partida dos *peers* ao *swarm* e as variáveis computadas pelo controlador.

houve ao menos uma partida de Leechers do sistema. Então, o valor assumido por $l(t)$ é igual a 1 caso tenham ocorrida ao menos uma partida de Leechers do sistema no intervalo $[t-w,t)$, $l(t)=2$ se nenhum Leechers deixou o sistema no intervalo $[t-w,t)$, mas ocorreu partida no intervalo $[t-2w,t-w)$, $l(t)=3$ se a última ocorrência de partida foi no intervalo $[t-3w,t-2w)$, e assim por diante. O valor assumido pela variável $l(t)$ é determinante para o controlador definir o número de *peers* considerados imaturos no sistema, representado por $a(t)$.

O valor de $a(t)$ é dado por: $a(t) = C(t) - C(t - l(t) * w)$. Por exemplo, se ocorreu ao menos uma partida de Leechers no intervalo $[t-w,t)$ (i.e., $d(t-w,t) > 0$ e $l(t) = 1$), então $a(t) = C(t) - C(t-w)$. Neste caso, apenas os Leechers que chegaram no último intervalo são considerados imaturos. É o caso ilustrado na Figura 5.19, por exemplo, nos instantes t_2, t_3, t_6 e t_7. No entanto, se nenhum Leecher concluiu o *download* no intervalo $[t-w,t)$ (i.e., $d(t-w,t) = 0$), então são considerados imaturos todos os Leechers que chegaram ao *swarm* nos últimos $l(t)$ intervalos. Os instantes t_5 e t_6, do gráfico mostrado na Figura 5.19, ilustram

duas situações em que não ocorreram partidas no intervalo $[t-w, t)$. Dessa forma, o valor atribuído à variável $l(t)$ e o número de *peers* considerados imaturos nesses casos foram $l(t_4) = 2$ e $a(t_4) = C(t_4) - C(t_4 - 2 * w) = 3$, no instante t_4, e $l(t_5) = 3$ e $a(t_5) = C(t_5) - C(t_5 - 2 * w) = 5$, no instante t_5. (O instante t_1 reflete o período de inicialização do algoritmo do controlador e durante esse período, até que ocorra a partida do primeiro Leecher do sistema, todos os *peers* são considerados imaturos.)

O valor escolhido para o tamanho do intervalo foi definido como $w = 0.2 * S/p$, onde S é o tamanho do conteúdo disponibilizado e p o limite superior da taxa máxima de *upload* definida para o Publisher. Esse valor foi definido empiricamente, mas representa uma fração do conteúdo que permite ao Leecher iniciar o processo de barganha por troca de dados com os demais *peers* do sistema. No entanto, esse valor pode ser facilmente alterado.

O procedimento definido para o controlador é resumido no Algoritmo 5.1.

Algoritmo 5.1 Controlador para determinar a taxa máxima de *upload* do Publisher.

Passo 1: A partir das informações obtidas do Tracker, computa o número total de chegadas e partidas ocorridas no *swarm* no último intervalo de w segundos;

Passo 2: Se ocorreram partidas no intervalo $[t-w, t)$ (i.e., $D(t) \geq 1$), então $l(t) = 1$; senão, $l(t) = l(t-w) + 1$.

Passo 3: Calcula o número de Leechers imaturos existentes no sistema através de $a(t) = C(t) - C(t - l(t) * w))$;

Passo 4: Estimar $B(t)$ utilizando a Equação 5.3, para determinar a taxa máxima de *upload* a ser utilizada pelo controlador do Publisher pelos próximos w segundos;

Passo 5: Aguarda w segundos e retorna ao **Passo 1**;

As Figuras 5.20(A) e (B) ajudam a compreender melhor como o controlador define o valor de B. Considere um *swarm* que contenha $N = 100$ Leechers e suponha que o limite definido para a taxa máxima a ser oferecida pelo Publisher seja $p =$ 100KBps. O gráfico mostrado na Figura 5.20(A) ilustra o valor atribuído a B pelo controlador (no eixo Y), em função do número total de Leechers considerados imaturos (no eixo X) dentre os $N = 100$ existentes no sistema. Se o número de Leechers imaturos for muito grande (por exemplo, se praticamente todos os 100 forem imaturos), o valor de B será alto, muito próximo do limite superior definido por pKBps. (Esse caso é representado na extremidade esquerda do gráfico da Figura

5.20(A).) Porém, à medida que a fração do número de imaturos decresce em relação ao total de Leechers do sistema (ou seja, que cresce o valor do eixo X, aproximando-se da extremidade direita do gráfico), o valor atribuído a B também diminui, podendo chegar ao limite inferior definido de 0.2*p=20KBps.

A Figura 5.20(B) mostra os valores definidos para B (eixo Y), no caso em que o número de Leechers imaturos é fixo (igual a 10) e a população do sistema (eixo X) decresce (de 120 até 10 Leechers). O gráfico demonstra o comportamento desejável para o valor atribuído a B, onde a taxa de *upload* deve ser baixa se o número de Leechers imaturos também for baixo (neste caso 10), em relação ao número total de *peers* existentes no sistema. (Caso representado na extremidade esquerda do gráfico da Figura 5.20(B).). No entanto, a taxa de *upload* deve aumentar à medida que o número total de Leechers no sistema diminuir, em relação ao total de 10 *peers* imaturos existentes no sistema (i.e., à medida que valor do eixo X se deslocar para a direita).

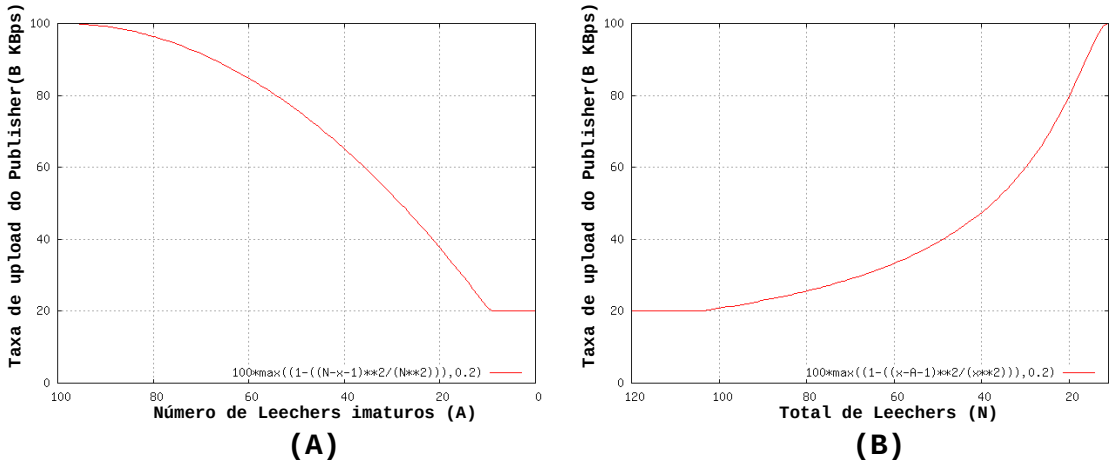


Figura 5.20: Análise para os valores definidos pelo controlador: (A) para um valor de N(t)=100 e a(t) variando de 1-100 Leechers; (B) para a(t)=10 e N(t) variando de 120-10 Leechers.

Para analisar a eficiência desta versão preliminar desenvolvida para o controlador, experimentos vêem sendo realizados no PlanetLAB. O controlador (descrito no Algoritmo 5.1) foi implementado na aplicação cliente BitTorrent 4.0.2 [114] utilizada pelos Publishers dos experimentos. Os demais *peers* permanecem utilizando a versão original da mesma aplicação. Nesses experimentos, o Publisher operou com a taxa de *upload* igual ao limite superior de p = 100KBps até a partida do

primeiro Leecher. Em seguida, o Publisher passou ajustar a taxa de *upload* através do controlador, variando entre $20 - 100$KBps. Em paralelo, experimentos foram executados para o caso em que o Publisher permaneceu operando a 100KBps, sem alterar a taxa de *upload* até o final do experimento.

Para cada configuração foram executadas 7 rodadas de experimentos, com duração de 8000 segundos cada. O tamanho do arquivo utilizado foi de $S = 20$MB e a capacidade dos Leechers foi de $\mu = 40$KBps. Diferentes popularidades foram atribuídas aos *swarms* para cada configuração de experimento. Os valores utilizados para λ foram: 1/10, 1/15, 1/20, 1/40, 1/80 e 1/200 peers/segundo.

As Figuras 5.21(A)-(F) comparam o volume total de dados servido pelos Publishers (com e sem controlador), em função do tempo decorrido nos experimentos, para cada configuração utilizada. Observando o gráfico da Figura 5.21(A) é possível notar que a economia da largura de banda do Publisher alcançada com o uso do controlador é de aproximadamente 75%, ao final dos 8000 segundos de experimento, e com uma tendência de continuar crescendo ao longo do tempo.

A diferença no consumo de banda do Publisher é significativa, principalmente no caso de um *swarm* muito popular. No entanto, essa diferença reduz à medida que o *swarm* se torna menos popular, como pode ser visto na sequência dos gráficos (A)-(F) mostrados na Figura 5.21. Isso ocorre porque, quando um Leecher chega ao *swarm* impopular, ele encontra um sistema vazio e o serviço é semelhante a cliente/servidor. Apenas o Publisher pode contribuir para a recuperação do conteúdo para esse Leecher e neste caso o controlador prevê que o valor de B fique próximo do limite superior definido por pKBps.

Os gráficos acima demonstram uma redução significativa do volume de tráfego gerado pelos Publishers com o uso do controlador. No entanto, uma restrição desejada com o uso do controlador é que o desempenho experimentado pelo usuário não sofra degradação significativa com a alocação dinâmica da banda do Publisher. Para analisar essa questão a Tabela 5.2 apresenta a média dos tempos de *download* obtido pelos usuários nos experimentos. Pelo valores apresentados, nota-se que a perda de desempenho é marginal, em relação ao ganho, na redução do custo para a disseminação do conteúdo, alcançado pelos Publishers com o uso do controlador.

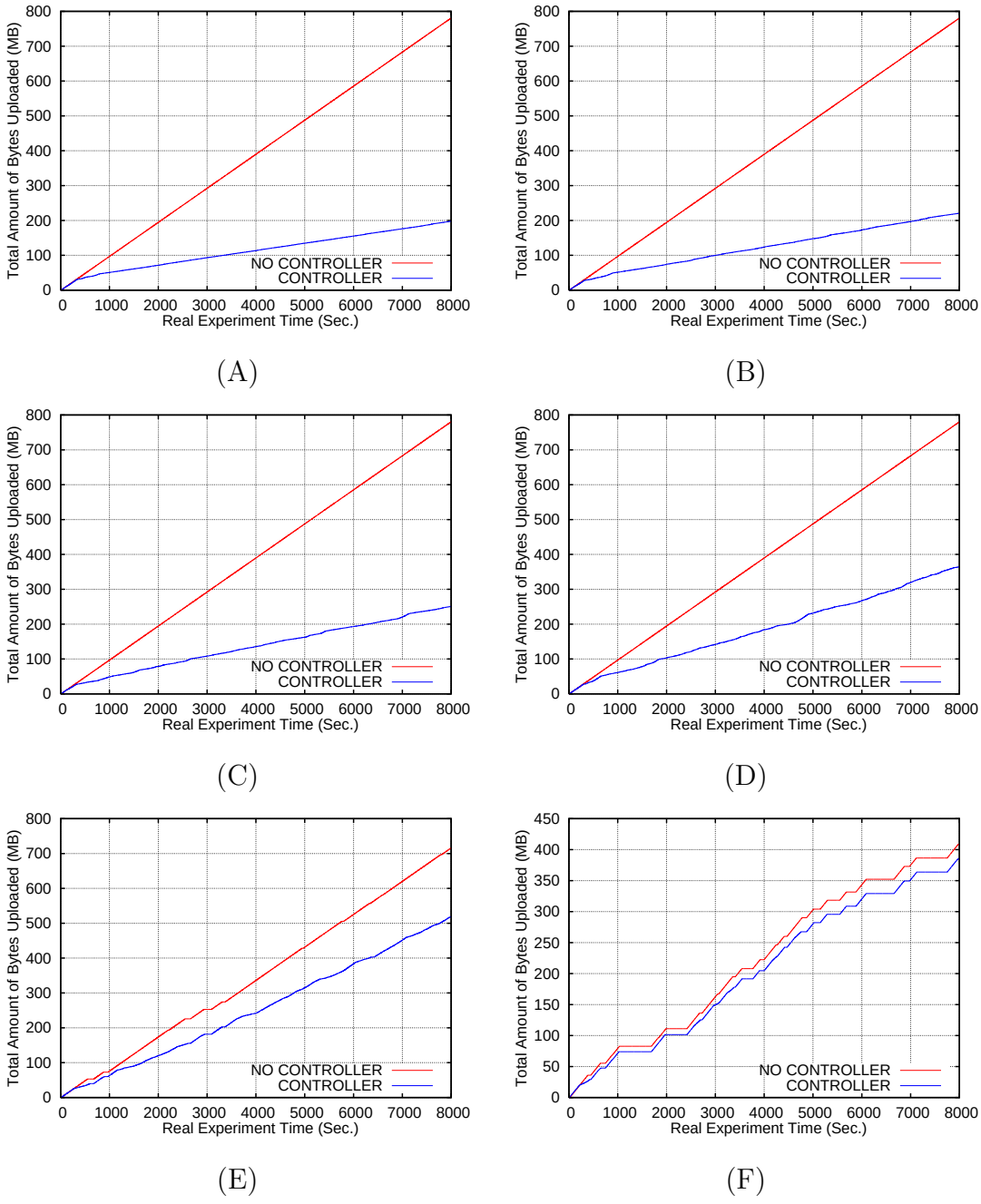


Figura 5.21: Experimentos usando controlador: (A) λ=1/10 peers/s; (B) λ=1/15 peers/s; (C) λ=1/20 peers/s; (D) λ=1/40 peers/s; (E) λ=1/80 peers/s; e, (F) λ=1/200 peers/s.

Tabela 5.2: Desempenho médio obtido pelos usuários nos experimentos.

Popularidade (λ peers/seg.)	Tempo médio de download(seg.) COM controlador — SEM controlador
1/10	423.13 — 414.98
1/15	415.86 — 405.14
1/20	408.71 — 387.25
1/40	351.11 — 326.85
1/80	268.59 — 256.36
1/200	182.87 — 183.36

Capítulo 6

Considerações finais

NOS capítulos anteriores (3, 4 e 5) são descritos em detalhes todos os trabalhos desenvolvidos nesta tese. As conclusões referentes a cada um dos problemas estudados foram apresentadas ao final dos respectivos capítulos. As duas seções existentes neste último capítulo sintetizam as principais contribuições alcançadas no desenvolvimento desta tese (Seção 6.1) e discorre sobre as perspectivas de possíveis trabalhos futuros (Seção 6.2).

6.1 Resumo das contribuições

Sobre medição de atraso unidirecional

- [i] Uma nova técnica de medição ativa não cooperativa para estimar a média e a variância da distribuição do atraso em um único sentido. A proposta não requer permissões de acesso à máquina remota para executar qualquer processo de coleta das sondas e contorna o problema da falta de acesso à máquina remota explorando valores do IPID dos pacotes recebidos ou fazendo *IP spoofing* com os pacotes enviados;

- [ii] Uma extensão da técnica que permite tratar problemas como *Skew* e *Offset*, decorrentes da falta de sincronização entre os relógios das máquinas;

- [iii] A realização de diversos experimentos executados na Internet para avaliação e validação dos algoritmos propostos;

- [iv] O uso de modelos de simulação para analisar os algoritmos quando as medições são aplicadas sobre diferentes cargas de utilização da largura de banda nos canais da rede;

- [v] Análise quantitativa do erro causado pela suposição de igualdade nos tempos de propagação dos caminhos de ida e volta na rede;

A variação da técnica que utiliza o IPID das sondas para estimar a média e a variância da distribuição do atraso em um único sentido (contribuição [i]), juntamente com um conjunto limitado dos resultados de simulação (contribuição [iv]), foram apresentados pela primeira vez no SBRC'2006 [72]. Este trabalho foi premiado como melhor artigo da conferência naquele ano. Uma versão ampliada deste artigo, que incluía resultados de experimentos reais (contribuição [iii]), novos resultados de simulação (contribuição [iv]), e a extensão da técnica para o caso de relógios não sincronizados (contribuição [ii]) fizeram parte do trabalho [73] publicado no IFIP/Networking 2007.

Um artigo a ser submetido a uma revista está em processo de revisão final pelos autores. A versão mais recente do trabalho, em forma de relatório técnico, está disponível em [136]. Este trabalho apresenta a variação da técnica utilizando *IP Spoofing* (contribuição [i]), os novos resultados de experimentos reais (contribuição [iii]) e de simulação (contribuição [iv]), além da análise quantitativa do erro causado pela suposição de igualdade dos tempos de propagação, nos caminhos de ida e volta da rede (contribuição [v]).

Sobre medição de taxa de transmissão em redes sem fio 802.11

- [vi] Uma técnica de medição fim-a-fim para inferir a taxa de transmissão de uma máquina conectada através de uma rede sem fio IEEE 802.11;

- [vii] Experimentos realizados na Internet e em ambientes controlados para validar a técnica;

- [viii] Validações do algoritmo para estimar dinamicamente a taxa de transmissão, quando a opção de ajuste automático de taxa estiver habilitada pelo dispositivo sem fio;

O método proposto para estimar a taxa de transmissão de um enlace conectado a uma rede sem fio IEEE 802.11 (contribuição [vi]), com os primeiros resultados experimentais (contribuição [vii]), foram apresentados em [35], publicado no SBC/WPerformance'2006. Uma versão estendida incluindo os demais resultados experimentais (contribuições [vii e viii]) foram publicados em [36], aceito no IEEE/ICC'2007.

Sobre disponibilidade e custo para distribuição de conteúdo em aplicações P2P como BitTorrent

- [ix] Experimentos de simulação para analisar a relação entre a popularidade de um conteúdo do BitTorrent e a sua disponibilidade entre os Leechers do *swarm*, o custo para sua disseminação e o desempenho experimentado pelos usuários;

- [x] Avaliação experimental dos benefícios da prática de agrupamento de arquivos na disseminação de conteúdo. Os resultados comprovam que é possível aumentar significativamente a disponibilidade e reduzir o tempo total de download do conteúdo se os arquivos foram distribuídos na forma agrupada;

- [xi] Análise dos custos da distribuição de conteúdo em função da popularidade dos *swarms*. Os resultados demonstram que a disponibilidade do conteúdo em *swarms* auto-sustentáveis (i.e., muito populares) podem perdurar por um tempo muito grande e o custo de disseminação para os provedores é (quase) zero;

- [xii] Observação da possibilidade do uso de um controlador para alocação dinâmica da taxa máxima de *upload* do Publisher que reduz o custo da disseminação de conteúdo, a depender da popularidade do *swarm*;

A análise sobre as implicações da popularidade do conteúdo na disponibilidade entre os Leechers, custo de disseminação e desempenho (contribuição [ix]) são parte do trabalho apresentado [137], publicado na revista Performance Evaluation Review. Uma versão estendida deste trabalho foi submetida ao Performance 2010 e um relatório técnico encontra-se em [138]. Os resultados de simulação apresentados nesta tese foram essenciais para o desenvolvimento dos modelos analíticos apresentados em [138].

Os resultados de experimentos que comprovam o aumento da disponibilidade e a redução no tempo de *download* de conteúdos disseminados de forma agrupada (contribuição [x]) foram publicados em [25, 26]. O trabalho [25] recebeu o prêmio de melhor artigo do ACM/CoNext 2009 e, por isso, uma versão estendida deste artigo será publicada no IEEE/ACM Transactions on Networking.

A análise dos custos da distribuição de conteúdo em função da popularidade dos *swarms*, assim como a observação da possibilidade do uso de soluções que possibilitem a redução do consumo de banda de provedores para a distribuição de conteúdo via sistemas P2P (contribuições [xi e xii]), foram apresentadas no artigo [24], aceito no SBRC'2009. Foi apresentado um estudo preliminar que demonstra a viabilidade do uso de um controlador para redução de custo. As próximas etapas deste trabalho encontram-se detalhadas na descrição de trabalhos futuros desta tese.

6.2 Possibilidades de trabalhos futuros

Relacionados à área de medições

Duas importantes métricas de interesse para aplicações na Internet são a capacidade de contenção e a largura de banda disponível. A utilidade dessas medidas para as aplicações na Internet já foram amplamente discutidas nos Capítulos 1 e 2 desta tese. Portanto, o desenolvimento de métodos não cooperativos, semelhantes ao proposto nesta tese, que possibilitem estimar a largura de banda disponível e a capacidade de conteção são possíveis trabalhos futuros. Do meu conhecimento, até o presente momento, na literatura, apenas o trabalho de Antoniades et. al [139] se propõe a estimar uma dessas duas métricas (largura de banda disponível) através de métodos não cooperativos de medição ativa. No entanto, possui limitações, como a dependência de um servidor web em operação na máquina alvo, além de conhecimento prévio de objetos web disponibilizados por esse servidor.

A técnica desenvolvida neste trabalho, que estima a taxa de transmissão de um enlace conectado por uma rede de acesso sem fio, trata-se de um método da forma ativa de medição. Um possível trabalho futuro, relacionado à proposta apresentada nesta tese, e com importantes aplicações na área de redes, seria a definição de uma versão passiva para a técnica. Neste caso, a estimativa da taxa de transmissão seria

feita sem a necessidade de geração de novas sondas, apenas a partir de pacotes originados de aplicações convencionais que são coletados de forma estratégica em algum ponto da rede. Uma das aplicações para esta técnica passiva está relacionada ao trabalho apresentado em [140]. Neste trabalho os autores avaliam os dispositivos e aplicativos "sniffers" específicos para monitorar enlaces 802.11 e evidenciam a ineficiência na coleta obtida por esses equipamentos. O trabalho sugere que novos métodos para inferência sejam desenvolvidos com o objetivo de reconstruir, com maior precisão, a lista de eventos de enlaces 802.11. Para isso, conhecer a taxa de transmissão utilizada pelos equipamentos sem fio, conectados ao enlace monitorado, é, sem dúvida, uma informação importante para auxiliar os métodos de inferência sugeridos pelos autores daquele trabalho.

Relacionados às aplicações P2P

Os resultados dos experimentos comprovam que o agrupamento de arquivos pode aumentar significativamente a disponibilidade de conteúdos que não sejam muito populares. No entanto, algumas questões ainda sem resposta servem de motivação para possíveis trabalhos futuros relacionados a esta área. Uma questão a ser considerada seria, como agrupar os arquivos de forma ótima para que sejam alcançados os objetivos de disponibilidade e desempenho desejados pelo provedor? A construção de um modelo que nos permita responder essa questão, assim como a realização de experimentos que comprovem a validade desse modelo, são dois importantes problemas em aberto nesta área. Um outra questão importante seria, qual o impacto da prática do agrupamento de arquivos no BitTorrent no volume de tráfego da rede?

Um estudo teórico para axiliar na definição de um controlador ótimo é um dos trabalhos de continuidade desta tese já em andamento. Os indícios de que é possível reduzir o custo para um Publisher na distribuição de conteúdo na Internet motivam este trabalho. Porém, o algoritmo utilizado até o momento tem embasamento apenas empírico. É necessária uma formalização do problema para que se possa determinar um algoritmo próximo da um modelo ótimo desejado.

Um outro possível trabalho futuro, relacionado aos estudos desenvolvidos com aplicações P2P, é a definição de uma versão de controlador para múltiplos *swarms*. Neste caso, o mecanismo, que definirá as taxas de *upload* de um Publisher para di-

versos *swarms*, pode ter o objetivo de maximizar o desempenho global (considerando todos os usuários de todos os *swarms* servidos), mas limitado a uma fração mínima dedicada a cada um dos *swarms*.

Referências Bibliográficas

[1] CERF, V., KAHN, R., "A protocol for packet networks intercomunication", *IEEE Transaction on Communications*, v. 22, n. 5, pp. 637–648, May 1974.

[2] "Internet System Consortiun", `http://www.isc.org`, 2009, [Último acesso: 01/02/2010].

[3] "Internet World Stats", `http://www.internetworldstats.com/stats.htm`, 2009, [Último acesso: 01/02/2010].

[4] SALTZER, J. H., REED, D. P., CLARK, D. D., "End-to-End Arguments in System Design", *ACM Transactions in Computer Systems*, v. 2, n. 4, pp. 277–288, November 1984.

[5] "Skype", `http://www.skype.com`, 2009, [Último acesso: 01/02/2010].

[6] AZEVEDO, J. A., NETTO, B. C., E. A. DE SOUZA E SILVA, R. M. L., "FreeMeeting: um ambiente para trabalho cooperativo e ensino a distância". In: *7th International Free Software Forum*, pp. 319–323, April 2006.

[7] LAND, "FreeMeeting", `http://www.land.ufrj.br/tools/fm/index.php`, 2009, [Último acesso: 01/02/2010].

[8] SCHULZE, H., MOCHALSKI, K., *Internet Study 2008/2009*, Tech. rep., Ipoque, 2009, `http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009`.

[9] COHEN, B., "BitTorrent", `http://www.bittorrent.com/`, 2009, [Último acesso: 01/02/2010].

[10] "Emule", http://www.emule-project.net/, 2009, [Último acesso: 01/02/2010].

[11] "PPLive", http://www.pplive.com/en/index.html, 2009, [Último acesso: 01/02/2010].

[12] "Sopcast", http://www.sopcast.org/, 2009, [Último acesso: 01/02/2010].

[13] TORRENT FREAK, "Comcast throttles bittorrent traffic, seeding impossible", http://torrentfreak.com/comcast-throttles-bittorrent-traffic-seeding-impossible, August 2007, [Último acesso: 01/02/2010].

[14] THE NEW YORK TIMES, "Comcast adjusts way it manages internet traffic", http://www.nytimes.com/2008/03/28/technology/28comcast.html, March 2008, [Último acesso: 01/02/2010].

[15] INTERNATIONAL HERALD TRIBUNE, "Who will pay as the Internet grows?" http://www.iht.com/articles/2008/06/08/technology/neutral09.php, June 2008, [Último acesso: 01/02/2010].

[16] BRADEN, R., CLARK, D., SHENKER, S., *RFC 1633: Integrated services in the Internet architecture: an overview*, IETF, June 1994.

[17] BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., Z.WANG, W.WEISS, *RFC 2475: An architecture for differentiated services*, IETF, December 1998.

[18] "IP Performance Metrics", http://www.ietf.org/dyn/wg/charter/ippm-charter.html, [Último acesso: 01/02/2010].

[19] DE CICCO, L., MASCOLO, S., PALMISANO, V., "An Experimental Investigation of the Congestion Control Used by Skype VoIP". In: *5th international conference on Wired/Wireless Internet Communications*, pp. 153–164, Coimbra, Portugal, May 2007.

[20] HARATCHEREV, L., TAAL, J., LANGENDOEN, K., LAGENDIJK, R., SIPS, H., "Optimized video streaming over 802.11 by cross-layer signal-

ing", *IEEE Communications Magazine*, v. 44, n. 1, pp. 115–121, January 2006.

[21] FILHO, F. S., WATANABE, E. H., DE SOUZA E SILVA, E. A., "Adaptive forward error correction for interactive streaming over the Internet". In: *IEEE Globecom*, pp. 1–6, San Francisco, CA, USA, November 2006.

[22] WATANABE, E. H., MENASCHÉ, D. S., DE SOUZA E SILVA, E. A., LEÃO, R. M., "Modeling Resource Sharing Dynamics of VoIP users over a WLAN Using a Game-Theoretic Approach". In: *IEEE INFOCOM*, pp. 915–923, Phoenix, AZ, USA, April 2008.

[23] SUH, K., FIGUIEREDO, D. R., KUROSE, J., TOWSLEY, D., "Characterizing and Detecting Skype-Relayed Traffic". In: *IEEE INFOCOM*, pp. 1–12, Barcelona, Spain, April 2006.

[24] ROCHA, A. A., MENASCHÉ, D. S., TOWSLEY, D. F., VENKATARAMANI, A., "On P2P systems for enterprise content delivery". In: *XVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 379–392, Maio 2009.

[25] MENASCHE, D. S., ROCHA, A. A., LI, B., TOWSLEY, D. F., VENKATARAMANI, A., "Content Availability and Bundling in Swarming Systems". In: *ACM CoNext*, pp. 121–132, December 2009.

[26] MENASCHÉ, D. S., ROCHA, A. A., DE SOUZA E SILVA, E. A., LEÃO, R. M., TOWSLEY, D. F., VENKATARAMANI, A., "Modeling Chunk Availability in P2P Swarming Systems", *ACM SIGMETRICS Performance Evaluation Review*, v. 37, September 2009.

[27] MUUSS, M., "Ping Tool", `http://ftp.arl.army.mil/pub/ping.shar`, [Último acesso: 01/02/2010].

[28] JACOBSON, V., "Traceroute Tool", `ftp://ftp.ee.lbl.gov/traceroute.tar.Z`, [Último acesso: 01/02/2010].

[29] PAXSON, V., "End-to-end routing behavior in the Internet", *IEEE/ACM Transaction on Networking*, v. 5, 1997.

[30] ZHANG, M., ZHANG, C., PAI, V., PETERSON, L., WANG, R., "PlanetSeer: internet path failure monitoring and characterization in wide-area services". In: *6th Symposium on Operating Systems Design and Implementation*, pp. 167–182, San Francisco, CA, USA, October 2004.

[31] STEENBERGEN, R. A., "A practical guide to (correctly) troubleshooting with traceroute". In: *North American Network Operators Group*, pp. 1–49, Santo Domingo, Dominican Republic, 2009, `http://www.nanog.org/meetings/nanog45/presentations/Sunday/RAS_traceroute_N45.pdf`.

[32] MADHYASTHA, H. V., ISDAL, T., PIATEK, M., DIXON, C., ANDERSON, T., KRISHNAMURTHY, A., VENKATARAMANI, A., "iPlane: An information plane for distributed services". In: *7th Symposium on Opearting Systems Design and Implementation*, pp. 367–380, Seattle, WA, USA, November 2006.

[33] KATZ-BASSETT, E., MADHYASTHA, H. V., JOHN, J. P., KRISHNAMURTHY, A., AMD T. ANDERSON, D. W., "Studying Black Holes in the Internet with Hubble". In: *5th USENIX Symposium on Networked Systems Design and Implementation*, pp. 247–262, San Francisco, California, USA, December 2008.

[34] "Hubble: Monitoring Internet Reachability in Real-Time", `http://hubble.cs.washington.edu/`, 2007, [Último acesso: 01/02/2010].

[35] ROCHA, A. A., LEÃO, R. M., DE SOUZA E SILVA, E., "Estimando a taxa de transmissão de redes de acesso sem fio através de medições fim-a-fim na Internet". In: *V WPerformance/XXVI SBC*, pp. 1–18, Campo Grande, Brasil, Agosto 2006.

[36] ROCHA, A. A., LEÃO, R. M., DE SOUZA E SILVA, E. A., "An End-to-End Technique to Estimate the Transmission Rate of an IEEE 802.11 WLAN". In: *IEEE ICC*, pp. 1–6, Glasgow, Scotland, June 2007.

[37] PAPAGIANNAKI, K., TAFT, N., ZHI-LI, Z., DIOT, C., "Long-term forecasting of internet backbone traffic: observations and initial models". In:

IEEE INFOCOM, v. 2, pp. 1178–1188, San Francisco, CA, USA, March 2003.

[38] DE SOUZA E SILVA, E. A., LEÃO, R. M., TRINDADE, M., ROCHA, A. A., RIBEIRO, B., DUARTE, F., AZEVEDO, J., "Um método para projeção de tráfego usando wavelets e fecho convexo". In: *XXI Simpósio Brasileiro de Telecomunições*, pp. 1–6, Belém, PA, Brasil, Setembro 2004.

[39] IEEE STANDARD 802.11, "LAN/MAN standards of the IEEE Computer Society. Wireless LAN medium access control (MAC) and physical layer (PHY) specification", 1997.

[40] IEEE STANDARD 802.11A/B/G, "IEEE 802.11, 802.11a, 802.11b, 802.11g standards for wireless local area netowrks", `http://standards.ieee.org/getieee802/802.11.html`.

[41] PAXSON, V., *Measurements and analysis of end-to-end Internet dynamics*, Ph.D. Thesis, Computer Science Division, and Information and Computing Sciences Division, Lawrence Berkeley National Laboratory, University of California, Berkeley, April 1997.

[42] SPRING, N., WSTHERALL, D., ANDERSON, T., "Reverse Engineering the Internet", *SIGCOMM Computer Communications Review*, v. 34, n. 1, pp. 3–8, 2004.

[43] ZIVIANI, A., DUARTE, O. C. M., *Metrologia na Internet*, Minicurso do SBRC, Fortaleza, CE, Brasil, Maio 2005.

[44] CROVELLA, M., KRISHNAMURTHY, B., *Internet Measurement: Infrastructure, Traffic And Applications*. 1st ed. John Wiley and Sons: New York, NY, USA, 2006.

[45] MEASUREMENT SYSTEM, E., "Guaranteed Packet Capture with DAG cards", `http://www.endace.com/guaranteed-packet-capture.html`, 2001, [Último acesso: 01/02/2010].

[46] IPOQUE, "Ipoque's DPX Network Probe", `http://www.ipoque.com/products/dpx-network-probe`, 2008, [Último acesso: 01/02/2010].

[47] TECHNOLOGIES, C., "AirPcap: USB-Based WLAN packet capture solutions", http://www.cacetech.com/products/airpcap.html, 2005, [Último acesso: 01/02/2010].

[48] "Tcpdump and libpcap programs", http://www.tcpdump.org/, 2008, [Último acesso: 01/02/2010].

[49] "Wireshark: network protocol analyzer", http://www.wireshark.org/, 1998, [Último acesso: 01/02/2010].

[50] CISCO SYSTEMS, INC., "Cisco Netflow", http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html, [Último acesso: 01/02/2010].

[51] POSTEL, J., *RFC 792: Internet Control Message Protocol*, IETF, September 1981.

[52] MAHAJAN, R., SPRING, N., WETHERALL, D., ANDERSON, T., "User-level internet path diagnosis". In: *19th ACM SOSP*, pp. 106–119, 2003.

[53] SAVAGE, S., "Sting: a TCP-based Network Measurement Tool". In: *USENIX Symposium on Internet Technologies and Systems*, pp. 71–79, 1999.

[54] BELLARDO, J., SAVAGE, S., "Measuring Packet Reordering". In: *2nd ACM SIGCOMM IMW*, pp. 97–105, 2002.

[55] CHEN, W., HUANG, Y., RIBEIRO, B., SUH, K., ZHANG, H., DE SOUZA E SILVA, E., KUROSE, J., TOWSLEY, D., "Exploiting the IPID Field to Infer Network Path and End-System Characteristics". In: *Passive and Active Measurement (PAM)*, pp. 108–120, Boston, MA, USA, March 2005.

[56] ZHAO, Y., CHEN, Y., BINDEL, D., "Toward Unbiased End-to-End Network Diagnosis". In: *ACM SIGCOMM*, pp. 219–230, 2006.

[57] GOVINDAN, R., PAXSON, V., "Estimating Router ICMP Generation Time". In: *Passive and Active Measurement (PAM)*, pp. 6–13, Fort Collins, CO, USA, March 2002.

[58] POSTEL, J., *RFC 791: Internet Protocol*, IETF, September 1981.

[59] INSECURE.ORG, "Remote OS detection via TCP/IP Stack FingerPrinting", `http://www.insecure.org/nmap/nmap-fingerprinting-article.txt`, Outubro 1998, [Último acesso: 01/02/2010].

[60] INSECURE.ORG, "Idle Scanning and related IPID games", `http://www.insecure.org/nmap/idlescan.html`, Setembro 1997, [Último acesso: 01/02/2010].

[61] BELLOVIN, S., "A Technique for Counting NATed Hosts". In: *ACM SIGCOMM IMW*, pp. 267–272, Marseille, France, November 2002.

[62] BEVERLY, R., BAUER, S., "THE Spoofer Project: Inferring the Extent of Source Address Filtering on the Internet". In: *USENIX - The Steps to Reducing Unwanted Traffic on the Internet Workshop* , pp. 53–59, Cambridge, USA, July 2005.

[63] D. MILLS, *RFC 1305: Network Time Protocol (Version 3) - Specification Implementation and Analysis*, IETF, Março 1992.

[64] PAXSON, V., "On Calibrating Measurements of Packet Transit Times". In: *ACM/Sigmetrics*, pp. 11–21, Madison, Wisconsin, USA, Junho 1998.

[65] MOON, S., SKELLY, P., TOWSLEY, D., "Estimation and Removal of Clock Skew for Network Delay Measurements". In: *IEEE Infocom*, pp. 227–234, New York, USA, Março 1999.

[66] LOUNG, D., BIRO, J., "Needed Services for Network Performance Evaluation". In: *IFIP Workshop on Performance Modeling and Evaluation of ATM Networks*, pp. 501–510, Inglaterra, Julho 2000.

[67] TSURU, M., TAKINE, T., OIE, Y., "Estimation of Clock Offset from One-way Delay Measurement on Asymmetric Paths". In: *SAINT International Symposium on Applications and the Internet*, pp. 126–133, Nara, Japão, Fevereiro 2002.

[68] ZHANG, L., LIU, Z., XIA, C., "Clock Synchronization Algorithms for Network Measurements". In: *IEEE/Infocom*, pp. 160–169, New York, USA, Junho 2002.

[69] PÁSZTOR, A., VEITCH, D., "PC based precision timing without GPS". In: *ACM/Sigmetrics*, pp. 1–10, Marina del Rey, California, USA, Junho 2002.

[70] VEITCH, D., BABU, S., PÁSZTOR, A., "Robust synchronization of software clocks across the internet". In: *ACM SIGCOMM IMC*, pp. 219–232, Taormina, Italy, October 2004.

[71] ROCHA, A. A., LEÃO, R. M., DE SOUZA E SILVA, E. A., "Metodologia para Estimar o Atraso em um Sentido e Experimentos na Internet". In: *XXII Simpósio Brasileiro de Redes de Computadores*, pp. 589–602, Gramado, Brasil, Maio 2004.

[72] ROCHA, A. A., LEÃO, R. M., DE SOUZA E SILVA, E. A., "Estimando a média e a variância do atraso em um sentido utilizando o IPID da máquina remota". In: *XXIV Simpósio Brasileiro de Redes de Computadores*, pp. 147–162, Curitiba, Brasil, Maio 2006.

[73] ROCHA, A. A., LEÃO, R. M., DE SOUZA E SILVA, E. A., "A Non-cooperative Active Measurement Technique for Estimating the Average and Variance of the One-way Delay", *IFIP/Metworking, Lecture Notes in Computer Science*, v. 4479, pp. 1084–1095, Maio 2007.

[74] LAND, "Tangram-II v.3.1", `http://www.land.ufrj.br/tools/tangram2/tangram2.html`, 2009, [Último acesso: 01/02/2010].

[75] DE SOUZA E SILVA, E., LEÃO, R., MUNTZ, R., DA SILVA, A., ROCHA, A., DUARTE, F., FILHO, F., JAIME, G., "Modeling, Analysis, Measurement and Experimentation with the Tangram-II Integrated Environment". In: *In: International Conference on Performance Evaluation Methodologies and Tools, 2006*, v. 180, pp. 1–10, Pisa, 2006.

[76] ROCHA, A., JAIME, G., MURAI, F., ALVES, B., FIGUEIREDO, D., LEÃO, R., DE SOUZA E SILVA, E., "Novas evoluções integradas à ferramenta Tangram-II v3.1". In: *Salão de Ferramentas / XXVII Simpósio Brasileiro de Redes de Computadores*, pp. 33–40, Recife, PE, Maio 2009.

[77] DE SOUZA E SILVA, E. A., RATTON, D., LEÃO, R. M., "The TANGRAMII Integrated Modeling Environment for Computer Systems and Networks", *Performance Evaluation Review*, v. 36, pp. 64–69, 2009.

[78] JACOBSON, V., "Pathchar - A tool to Infer Network Characteristics of Internet Paths", `ftp://ftp.ee.lbl.gov/pathchar/`, 1997, [Último acesso: 01/02/2010].

[79] DOWNEY, A., "Clink: a tool for estimating Internet link characteristics", `http://allendowney.com/research/clink/`, 1999, [Último acesso: 01/02/2010].

[80] DOWNEY, A., "Using Pathchar to Estimate Internet Link Characteristics". In: *ACM SIGCOMM*, pp. 241–250, Cambridge, USA, Setembro 1999.

[81] LAI, K., BAKER, M., "Measuring Link Bandwidths using a Deterministic Model of Packet Delay". In: *ACM SIGCOMM*, pp. 283–294, Stockholm, Suécia, July 2000.

[82] DOVROLIS, C., RAMANATHAN, P., MOORE, D., "What do Packet Dispersion Techniques Measure?" In: *IEEE Infocom*, v. 1, pp. 905–914, Anchorage, USA, Abril 2001.

[83] DOVROLIS, C., "Pathrate: a measurement tool for the capacity of network paths ", `http://www.pathrate.org`, 2001, [Último acesso: 01/02/2010].

[84] DOVROLIS, C., "Pathload: a measurement tool for the available bandwidth of network paths", `http://www.pathload.org`, 2001, [Último acesso: 01/02/2010].

[85] COOPERATIVE ASSOCIATION FOR INTERNET DATA ANALYSIS (CAIDA), "Bandwidth / Throughput Measurement Tools", `http://www.caida.org/tools/taxonomy/perftaxonomy.xml`, 2009, [Último acesso: 01/02/2010].

[86] JACOBSON, V., "Congestion Avoidance and Control". In: *ACM SIGCOMM*, pp. 314–329, Stanford, USA, Setembro 1988.

[87] CARTER, R. L., CROVELLA, M. E., "Measuring Bottleneck Link Speed in Packet-Switched Networks". In: *Performance Evaluation*, v. 27, 28, pp. 297–318, 1996.

[88] HARFOUSH, K., BESTRAVOS, A., BYERS, J., "Measuring Bottleneck Bandwidth of Targeted Path Segments". In: *IEEE Infocom*, v. 3, pp. 2079–2089, São Francisco, CA, EUA, Abril 2003.

[89] ROESLER, V., FINZSCH, P., ANDRADE, M., LIMA, J. V., "Análise do Mecanismo de Pares de Pacotes Visando Estimar a Banda da Rede via UDP". In: *XXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pp. 103–118, Natal, Brasil, Maio 2003.

[90] AUGUSTO, M., MURTA, C., "Avaliação Experimental de Ferramentas para Medição de Capacidade em Redes de Computadores". In: *II WPerformance/XXIII SBC*, pp. 129–142, Campinas, Brasil, Agosto 2003.

[91] ROCHA, A. A., LEÃO, R. M., DE SOUZA E SILVA, E., "Proposta de uma técnica de seleção dos pares de pacotes para estimar a capacidade de contenção". In: *III WPerformance/XXIV SBC*, pp. 1–18, Salvador, Brasil, Agosto 2004.

[92] KESHAV, S., "A Control-Theoretic Approach to Flow Control". In: *ACM SIGCOMM*, pp. 3–15, Zürich, Switzerland, Setembro 1991.

[93] BOLOT, J., "Characterizing End-to-End Delay and Loss in the Internet". In: *ACM SIGCOMM*, pp. 289–298, San Fracisco, USA, Setembro 1993.

[94] KAPOOR, R., CHEN, L., LAO, L., GERLA, M., SANADIDI, M., "CapProbe: A Simple and Accurate Capacity Estimation Technique". In: *ACM SIGCOMM*, v. 34, pp. 67–78, Portland, USA, Outubro 2004.

[95] LAKSHMINARAYANAN, K., PADMANABHAN, V., PADHYE, J., "Bandwidth Estimation in Broadband Access Networks". In: *ACM SIGCOMM IMC*, pp. 314–321, Taormina, Italy, Maio 2004.

[96] KUMAR, R., ROSS, K., "Peer-Assisted File Distribution: The Minimum Distribution Time". In: *IEEE Hot Topics in Web Systems and Technologies*, pp. 1–11, Boston, MA, USA, November 2006.

[97] QIU, D., SRIKANT, R., "Modeling and performance analysis of BitTorrent-like peer-to-peer networks". In: *ACM SIGCOMM*, pp. 367–378, Portland, OR, USA, September 2004.

[98] RAMACH, A., SARMA, A. D., FEAMSTER, N., "BitStore: An Incentive-Compatible Solution for Blocked Downloads in BitTorrent". In: *Workshop on The Economics of Networked Systems and Incentive-Based Computing*, pp. 29–36, San Diego, CA, USA, June 2007.

[99] GUO, L., CHEN, S., XIAO, Z., TAN, E., DING, X., ZHANG, X., "A performance study of BitTorrent-like peer-to-peer systems", *IEEE Journal on Selected Areas in Communications*, v. 25, pp. 155–169, January 2007.

[100] POUWELSE, J., GARBACKI, P., EPEMA, D., SIPS, H., "The Bittorrent P2P File-Sharing System: Measurements and Analysis". In: *IV International Workshop on Peer to Peer Systems*, pp. 205–216, Ithaca, NY, USA, February 2005.

[101] CHU, J., LABONTE, K., LEVINE, B., "Availability and locality measurements of peer-to-peer file systems". In: *ITCom: Scalability and Traffic Control in IP Networks*, pp. 310–321, Boston, MA, USA, July 2002.

[102] BHAGWAN, R., SAVAGE, S., VOELKER, G. M., "Understanding availability". In: *III International Workshop on Peer to Peer Systems*, pp. 1–11, Berkeley, CA, USA, February 2003.

[103] GUMMADI, K., DUNN, R., SAROIU, S., GRIBBLE, S., LEVY, H., ZAHORJAN, J., "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload". In: *ACM Symposium on Operating Systems Principles*, pp. 314–329, Bolton Landing, NY, USA, October 2003.

[104] NEGLIE, G., REINA, G., ZHANG, H., TOWSLEY, D., VENKATARAMANI, A., DANAHER, J., "Availability in BitTorrent Systems". In: *IEEE Infocom*, pp. 2216–2224, Anchorage , Alaska, USA, May 2007.

[105] GKANTSIDIS, C., RODRIGUEZ, P., "Network Coding for Large Scale Content Distribution". In: *IEEE Infocom*, pp. 2235–2245, Miami , FL, USA, March 2005.

[106] TORRENT FREAK, "Interview with Bram Cohen, the inventor of BitTorrent", `http://torrentfreak.com/interview-with-bram-cohen-the-inventor-of-bittorrent/`, January 2007, [Último acesso: 01/02/2010].

[107] KONTIKI, INC., "Power of Commercial Peer-to-Peer Delivery", `http://www.kontiki.com/_download/The-Power-of-Commercial-P2P.pdf`, June 2008, [Último acesso: 01/02/2010].

[108] ABOUT.COM, "Peer-to-Peer Gets Down to Business", `http://pcworld.about.com/magazine/1905p149id44862.htm`, June 2005, [Último acesso: 01/02/2010].

[109] FORBES.COM, "Akamai Goes P2P", `http://www.forbes.com/2007/04/12/akamai-red-swoosh-tech-intel-cx_ag_0412akamai.html`, April 2007, [Último acesso: 01/02/2010].

[110] GKANTSIDIS, C., KARAGIANNIS, T., RODRIGUEZ, P., VOJNOVIC, M., "Planet Scale Software Update". In: *ACM Sigcomm*, pp. 423–434, Pisa, Italy, September 2006.

[111] AMD Y. CHEN, C. Z., LIN, C., "Experimental Analysis of Super-Seeding in BitTorrent". In: *IEEE International Conference on Communications*, pp. 65–69, Beijing, China, May 2008.

[112] HOFFMAN, J., "BitTornado", `http://www.bittornado.com/`, 2003, [Último acesso: 01/02/2010].

[113] BHARAMBE, A., HERLEY, C., PADMANABHAN, V., "Some observations on bitTorrent performance". In: *ACM SIGMETRICS*, pp. 398–399, Banff, Alberta, Canada, June 2005.

[114] LEGOUT, A., LIOGKAS, N., KOHLER, E., ZHANG, L., "Clustering and Sharing Incentives in BitTorrent Systems". In: *ACM SIGMETRICS*, pp. 301–312, San Diego, CA, June 2007.

[115] CHOW, A., GOLUBCHIK, L., MISRA, V., "Improving BitTorrent: A Simple Approach". In: *International Workshop on Peer-to-Peer Systems*, pp. 1–6, Tampa, FL, USA, February 2008.

[116] IOANNIDIS, S., MARBACH, P., "On the Design of Hybrid Peer-to-Peer Systems". In: *ACM SIGMETRICS*, pp. 157–168, Annapolis, Maryland, USA, June 2008.

[117] PETERSON, R., SIRER, E., "Antfarm: efficient content distribution with managed swarms". In: *USENIX symposium on Networked systems design and implementation*, pp. 107–122, Boston, MA, USA, April 2009.

[118] "PlanetLAB: an open plataform for developing, deploying and accessing planetary-scale services", `http://www.planet-lab.org/`, 2002, [Último acesso: 01/02/2010].

[119] TAQQU, M., WILLINGER, W., SHERMAN, R., "Proof of a Fundamental Result in Self-Similar Traffic Modeling". In: *ACM/Computer Communications Review*, pp. 5–23, Abril 1997.

[120] BAKSHI, B., KRISHNA, P., VAIDYA, N., PRADHAN, D., "Improving performance of TCP over wireless networks". In: *International Conference on Distributed Computing Systems*, pp. 365–373, Baltimore, EUA, Maio 1997.

[121] GERLA, M., BAGRODIA, R., ZHANG, L., TANG, K., L.WANG, "TCP over wireless multihop protocols: Simulation and experiments". In: *IEEE ICC*, pp. 1089–1094, Vancouver, Canadá, Junho 1999.

[122] COHEN, R., RAMANATHAN, S., "TCP for high performance in hybrid fiber coaxial broad-band access networks", *IEEE/ACM Transaction on Networking*, v. 6, pp. 15–29, 1998.

[123] CHENG, L., MARSIC, I., "Fuzzy Reasoning for Wireless Awareness", *International Journal of Wireless Information Networks*, v. 8, pp. 15–26, 2001.

[124] WEI, W., WANG, B., ZHANG, C., KUROSE, J., TOWSLEY, D., "Classification of Access Network Types: Ethernet, Wireless LAN, ADSL, Cable Modem or Dialup?" In: *IEEE/Infocom*, pp. 1060– 1071, Miami, USA, March 2005.

[125] WEI, W., JAISWAL, S., ZHANG, C., KUROSE, J., TOWSLEY, D., "Identifying 802.11 Traffic from Passive Measurements Using Iterative Bayesian Inference". In: *IEEE/Infocom*, pp. 1– 12, Barcelona, Espanha, March 2006.

[126] NICHOLS, J., CLAYPOOL, M., KINICKI, R., LI, M., "Measurements of the Congestion Responsiveness of Windows Streaming Media". In: *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, pp. 189–202, Cork, Irlanda, Junho 2004.

[127] LI, F., CHUNG, J., LI, M., WU, H., CLAYPOOL, M., KINICKI, R., "Application, Network and Link Layer Measurements of Streaming Video over a Wireless Campus Network". In: *Passive and Active Measurement (PAM)*, pp. 189–202, Boston, Massachusetts, EUA, Março 2005.

[128] BEJERANO, Y., BREITBART, Y., GAROFALAKIS, M., RASTOGI, R., "Physical topology discovery for large multi-subnet networks". In: *IEEE/Infocom*, pp. 342–352, São Francisco, EUA, Junho 2003.

[129] "The Network Simulator - ns-2", `http://www.isi.edu/nsnam/ns/`, 2009, [Último acesso: 01/02/2010].

[130] BHARAMBE, A., HERLEY, C., PADMANABHAN, V., "Analyzing and Improving a BitTorrent Network's Performance Mechanisms". In: *IEEE Infocom*, v. 1, pp. 1–12, Barcelona, Spain, Abril 2006.

[131] LUIZ JOSÉ HOFFMANN FILHO, *Algoritmos para acesso interativo em aplicações de vídeo P2P*, Master's Thesis, Universidade Federal do Rio de Janeiro, 2009.

[132] "Product Bundling", `http://en.wikipedia.org/wiki/Product_bundling`, 2009, [Último acesso: 01/02/2010].

[133] PIATEK, M., ISDAL, T., ANDERSON, T., KRISHNAMURTHY, A., VENKATARAMANI, A., "Do incentives build robustness in BitTorrent?" In: *4th USENIX Symposium on Networked Systems Design e Implementation*, pp. 1–12, Cambridge, USA, April 2007.

[134] ANDERSON, C., *The Long Tail: Why the Future of Business is Selling Less of More*. Hyperion, 2006.

[135] PAGE, W., "More Long Tail debate: mobile music no, search yes", `http://longtail.typepad.com`, 2008, [Último acesso: 01/02/2010].

[136] ROCHA, A. A., LEÃO, R. M., DE SOUZA E SILVA, E. A., *Estimating first two moments of the one-way delay with no cooperation from remote host*, Tech. rep., Federal University of Rio de Janeiro, 2010, [Último acesso: 01/02/2010].

[137] MENASCHÉ, D., A., R., DE SOUZA E SILVA, E., LEÃO, R., TOWSLEY, D., A.VENKATARAMANI, "Modeling chunk availability in P2P swarming systems", *Performance Evaluation Review*, v. 37, pp. 30–32, 2009.

[138] MENASCHÉ, D., A., R., DE SOUZA E SILVA, E., LEÃO, R., TOWSLEY, D., A.VENKATARAMANI, *Estimating Self-Sustainability in Peer-to-Peer Swarming Systems*, Tech. rep., ArXiv:1004.0395v2, 2010, [Último acesso: 10/04/2010].

[139] ANTONIADES, D., ATHANATOS, M., PAPADOGIANNAKIS, A., MARKATOS, E. P., DOVROLIS, C., "Available bandwidth measurement as simple as running wget". In: *Passive and Active Measurement (PAM)*, pp. 61–70, Adelaide, Australia, March 2006.

[140] SERRANO, P., ZINK, M., KUROSE, J., "Assessing the fidelity of COTS 802.11 sniffers". In: *IEEE INFOCOM*, pp. 1089–1097, Rio de Janeiro, Brazil, April 2009.