ERIC RODRIGUES GUIMARÃES

ROTEAMENTO DE MÍDIA CONTÍNUA EM TOPOLOGIAS REAIS DA INTERNET

Belo Horizonte, Minas Gerais
Agosto de 2006

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

ROTEAMENTO DE MÍDIA CONTÍNUA EM TOPOLOGIAS REAIS DA INTERNET

Dissertação apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ERIC RODRIGUES GUIMARÃES

Belo Horizonte, Minas Gerais
Agosto de 2006

UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Roteamento de mídia contínua em topologias reais da Internet

ERIC RODRIGUES GUIMARÃES

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D Jussara Marques de Almeida – Orientador
Universidade Federal de Minas Gerais

Ph. D Sérgio Vale Aguiar Campos
Universidade Federal de Minas Gerais

Ph. D Dorgival Olavo Guedes Neto
Universidade Federal de Minas Gerais

Belo Horizonte, Minas Gerais, Agosto de 2006

Resumo

Com a popularização das conexões de alta velocidade, cada vez mais usuários estão dispostos a utilizar aplicações baseadas em mídia contínua, motivando o desenvolvimento e implantação de diversas aplicações em uso comercial ou institucional [55, 40, 53, 49, 8, 39, 61, 62, 63, 60, 67, 14]. Mídia contínua é um tipo de aplicação de tempo real e que, por esse motivo, requer garantias de qualidade de serviço (QoS) por parte da infra-estrutura computacional. Existe uma série de abordagens para melhorar a QoS para mídia contínua em redes baseadas em "melhor esforço" como a Internet. Exemplos de abordagens são técnicas de replicação ou *caching* [66, 1, 3, 54], entrega compartilhada [19, 28, 18] e roteamento otimizado. Esta dissertação enfatiza a última abordagem e trata de mecanismos de roteamento que explorem a diversidade de caminhos [7, 6, 24] e o compartilhamento de fluxos [70, 4, 22].

A caracterização da diversidade de caminhos existente na Internet subsidia o primeiro grupo de trabalhos. No entanto, as tentativas de caracterização [59] realizadas anteriormente foram limitadas a ISPs norte-americanos e europeus. Além disso, os protocolos para otimizar roteamento baseados em fluxos compartilhados somente haviam sido avaliados em topologias sintéticas e em poucas topologias reais. Dadas as limitações dos trabalhos realizados até o momento, esta dissertação analisa mais a fundo os ganhos potenciais da aplicação de técnicas alternativas para roteamento de mídia contínua na Internet.

Para tanto, estendemos uma técnica de mapeamento de topologias em nível de roteadores [57] e a usamos para coletar topologias reais da Internet, espalhadas pelo mundo. Caracterizamos essas topologias quanto à diversidade de caminhos, a fim de subsidiar trabalhos que se baseiam na premissa da existência de tal diversidade. Nossa caracterização revelou a existência de uma alta diversidade de caminhos em topologias mais dispersas (continentais), e uma diversidade baixa para topologias mais restritas (países pequenos). Por fim, realizamos uma análise extensiva de heurísticas propostas por Almeida [2] para criação da árvore de distribuição para roteamento com fluxos compartilhados. Essa análise envolve o uso de topologias com diversos graus de dispersão e a variação de diversos parâmetros, tais como o número e a demanda dos *sites* clientes. Nossos resultados revelam que essas heurísticas são superiores na criação de árvores de distribuição para fluxos compartilhados, quando comparadas ao protocolo *default* [37] para *unicast* e similares propostos como padrão [41, 21] também para compartilhamento de fluxos, apresentando ganhos de cerca de 35%, em termos banda média de rede consumida. Ganhos da ordem de 70% foram obtidos em situações nas quais a grande maioria dos *sites* faz o papel de réplica e de cliente ao mesmo tempo.

Abstract

The spreading of high speed network connections has been driving more and more users into using streaming media applications, and therefore motivating the development and deployment of many such applications, either in commercial or institutional scenarios [55, 40, 53, 49, 8, 39, 61, 62, 63, 60, 67, 14]. Streaming media is a real time application, and for that reason it requires that the underlying computational infrastructure provide quality of service (QoS) warranties. Several approaches have been taken in improving QoS for streaming media over best-effort networks like the Internet. As examples of such approaches, there are replication or caching techniques [66, 1, 3, 54], multicast delivery [19, 28, 18], and optimized routing. This dissertation focuses on this last approach, dealing with mechanisms which exploit path diversity [7, 6, 24] and multicast routing [70, 4, 22].

Characterizing Internet's path diversity reinforces the first group of works. However, so far the attempts of such characterization [59] only cover some north-american and european ISP. Besides, protocols optimized for multicast routing had only been evaluated in synthetic topologies and a few real topologies. Given previous works' limitations, this dissertation analyzes more deeply the potential gains of using such alternative routing techniques on current Internet topologies.

In order to accomplish that, we extend a router-level topology mapping technique [57] and use it to collect real Internet topologies, spread all over the world. We characterize these topologies' path diversity, in order to give support to works which are based on the premise of the existence of such diversity. Our study reveals a high path diversity in wider-area (continental) topologies, and a low diversity in more restricted topologies, such as small countries. We then make an extensive analysis of heuristic protocols which assemble multicast routing trees, proposed in [2]. Such analysis includes using several topologies in various dispersion levels and varying several parameters, such as the number of client sites and their request rates. Our results show that these heuristics create better multicast routing trees, when compared to the default unicast protocol [37] and similar protocols for multicast proposed in [41, 21], using up to 35% less network bandwidth. Even better savings, of around 70%, where observed in scenarios where most of the sites act both as replicas and clients at the same time.

Para meus pais,
pelo amor e apoio que sempre me deram.

Agradecimentos

Gostaria de agradecer aos professores Dorgival Olavo Guedes Neto e Sérgio Vale Aguiar Campos, por terem me orientado ainda no tempo em que eu era um aluno de graduação, na minha iniciação científica e no meu projeto final de curso. Agradeço-lhes também por terem feito parte da banca examinadora da minha defesa de dissertação e por seus comentários que certamente melhoraram este trabalho.

Agradeço também aos meus colegas Felipe Augusto Dornelas e Alex Borges Vieira, pela grande colaboração, especialmente no princípio dos trabalhos que levaram a esta dissertação. Ao povo do laboratório VoD em geral, pela ótima companhia na hora do almoço.

Um agradecimento especial à minha orientadora Jussara Marques de Almeida. Sua dedicação em me orientar, suas cobranças e prazos, e sua paciência quando, em não poucas vezes, deixei cumprir tais prazos, me incentivaram a realmente me esforçar ao máximo neste trabalho. Também pelo seu perfeccionismo, que sempre encontrava algo a ser melhorado.

Tudo é melhor com música. Agradeço aos Beatles, The Doors, Pink Floyd, Nirvana, Queen, Guns 'n Roses, The Brian Jonestown Massacre, Metallica, pelos variados estilos de *rock* que tocaram em meus fones de ouvido enquanto eu trabalhava, e também a artistas brasileiros, os velhos Titãs, o mutante Skank, o irreverentíssimo Ultraje a Rigor, o ourobranquense Cartoon, o poeta Chico Buarque, a voz de Elis Regina, vários outros que estavam e a tantos outros que só não estavam na minha *playlist* por falta de espaço no disco rígido. Tudo é melhor com *sitcoms*, ou, como eu gosto de chamar, "seriados de meia hora com risadinhas para a gente saber a hora certa de rir"! Quando insone, obrigado aos Friends por me fazerem dormir.

Agradeço muito aos amigos com quem convivi mais de perto nesses últimos dois anos, os meus companheiros de república Jean, Marco Cristo e André, que me ajudaram a tomar cerveja na hora de descansar e tomar café na hora de trabalhar, também pelas discussões e dicas em relação a este trabalho e à defesa. Minhas irmãs adotadas nos últimos tempos, Pri e Lili, obrigado pelas idas ao supermercado, pelos jogos de pôquer e por tudo o mais quanto eu precisava para me distrair e manter minha cabeça sã! A tantos outros amigos que eu não via no dia-a-dia, mas pelo menos em alguns excelentes finais de semana, o Daniell, os Thiagos, o Juninho, o distante João Vidal, a Paula com quem vez ou outra eu "trombava" nos corredores do ICEx, Los Poderosos Patrulleros del Espacio e uma infinidade de outros amigos menos freqüentes mas não menos importantes. Obrigado por tudo, a toda a minha família (e nisso já se inclui a família da Bárbara), em especial a meus irmãos Túlio e Breno.

Pai e mãe, mãe e pai, meus agradecimentos, beijos e abraços. Obrigado por compreenderem o fato de eu não ter ido visitá-los tanto quanto gostaria e, das vezes em que fui, freqüentemente ter tido que voltar logo para Belo Horizonte para cumprir meus prazos. Obrigado pelo amor, carinho, conversas, conselhos, pizzas, vinhos e tudo de bom com que vocês preencheram os nossos encontros, seja nas minhas visitas a vocês, seja nas suas visitas a mim. Também agradeço por acreditarem em mim, me dando todo o apoio, inclusive financeiro, necessário aos meus estudos.

Bárbara ("bela do Norte estrela"), minha Cutinha, sempre ao meu lado e suportando a tensão dos últimos meses, dando-me ânimo para trabalhar até nos — para mim sagrados — fins-de-semana, eu não tenho palavras suficientes para agradecer-lhe devidamente, por tudo o que você fez. As preciosas palavras de apoio, as revisões dos meus textos, o fato de você me escutar falar de computação nas horas mais impróprias, as idas ao cinema, a Ouro Branco e tantos outros lugares aonde você me acompanhou e sempre acompanha me fazendo mais feliz. Às inumeráveis e inomináveis outras coisas que você fez por mim e que somente me ajudaram.

Obrigado a todos vocês.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

É crescente a demanda por aplicações multimídia na Internet. Com a popularização das conexões de alta velocidade, cada vez mais usuários estão dispostos a utilizar aplicações baseadas em vídeo e áudio sob demanda. Como exemplos de aplicações já amplamente utilizadas, existem o Skype [55], um programa de telefonia sobre IP baseado em redes P2P, além de aplicações de vídeo-conferência como o NetMeeting [40] ou o GnomeMeeting [53].

Há também transmissão de vídeo ou áudio (ao vivo ou pré-gravado) sob demanda para um grande conjunto de usuários da Internet ou de redes locais simultaneamente, tarefa essa realizada por servidores de vídeo sob demanda (VoD) como o o RealVideo [49], Darwin [8], ou o Windows Media Server [39]. Diversas aplicações que utilizam VoD estão já disponíveis na Internet, sendo exemplos a Rádio UOL [61], a TV UOL [62] e a Usina do Som [63]. A utilidade de tal tipo de serviço vai além do entretenimento (como é o caso de vídeo-locadoras virtuais ou a transmissão de jogos ao vivo), chegando até o campo da educação. Diversas universidades e centros de ensino em todo o mundo já utilizam, em maior ou menor grau, o ensino à distância. Como exemplos, citamos o sistema MANIC da Universidade de Massachusets [60], a PUC Minas Virtual [45], o programa eTeach da Universidade de Wisconsin [67] etc. Usa-se vídeo sob demanda também em treinamento de funcionários de empresas multinacionais com escritórios espalhados pelo mundo inteiro, como a Hewlett-Packard [14].

1.1 Qualidade de serviço em mídia contínua

Aplicações de vídeo e áudio sob demanda são categorias específicas do que chamamos mídia contínua, que consiste em aplicações em que, em geral, há um grande volume de dados a ser transferido do servidor para o cliente e esses dados podem ser exibidos à medida em que vão sendo recebidos pelo cliente. Esse tipo de aplicação contrasta com o *download* tradicional de arquivos, no qual o arquivo é primeiro totalmente transferido para o cliente, para somente então poder ser visualizado e/ou acessado.

Mídia contínua é um tipo de aplicação de tempo real e que, por esse motivo, requer garantias de qualidade de serviço (QoS) por parte da infra-estrutura computacional, em termos dos valores médios e da variabilidade de fatores como banda de rede disponível entre cliente

e servidor, atraso e taxa de perda de pacotes. Restrições de tempo devem ser satisfeitas para que tais aplicações funcionem corretamente: por exemplo, se dado quadro de um vídeo chegar ao cliente após o momento em que ele deveria ter sido exibido, ele é inútil e deve ser descartado, havendo então desperdício de banda.

Diversos fatores prejudicam a qualidade de serviço em aplicações de distribuição de mídia contínua pela Internet, entre eles o tráfego cruzado, a instabilidade da rede e o próprio consumo excessivo de banda de rede intrínseco desse tipo de aplicação. Devido ao modelo do tipo "melhor esforço" adotado na Internet, as aplicações atuais não podem oferecer garantia absoluta de qualidade ao usuário final. Isso porque não existem mecanismos de contrato automático de QoS amplamente implantados na Internet. Questões políticas, administrativas, de segurança e econômicas impedem a implantação de propostas já existentes para solucionar esse problema.

Existe na literatura uma série de abordagens para melhorar a qualidade de serviço em aplicações de mídia contínua na Internet. Uma das estratégias é a utilização de *caching* dos dados em servidores *proxy* próximos aos clientes [66, 1, 3, 54]. Uma outra estratégia é a redução da banda de servidor através de protocolos de entrega baseados em compartilhamento de fluxos, como *Bandwidth Skimming* [19], *Patching* [28], *Skyscraper Broadcast* [18], que provêem escalabilidade de banda de servidor sublinear com relação ao aumento do número de clientes. Esses protocolos podem usar compartilhamento de fluxos de rede tanto a nível de rede (*IP multicast*) quanto de aplicação (redes *overlay*). A redução na banda de servidor ocasiona também a redução de banda de rede e, conseqüentemente, melhor QoS.

Uma terceira estratégia diz respeito à utilização de mecanismos otimizados para roteamento de mídia contínua explorando diversidade de caminhos [7, 6, 24] e compartilhamento de fluxos [70, 4, 22]. O roteamento de mídia contínua é o tema desta dissertação.

1.2 Roteamento de Mídia Contínua

O princípio básico de um protocolo de roteamento é a formação de uma árvore de distribuição, cuja raiz é o servidor e qualquer nó pode ser um *site* cliente. Assume-se que o conjunto de clientes, bem como suas demandas, são estáticos. Na prática, o sistema seria reconfigurado sempre que houvesse alteração significativa no conjunto de clientes ou demanda. Esse tipo de abordagem é comum em redes de distribuição de conteúdo (CDNs), por ser mais barato que a reconfiguração automática a cada mínima mudança [4].

Assim, o objetivo desses protocolos é montar árvores de distribuição de forma a minimizar o custo de transmissão nessas árvores. O custo é dado pela soma dos valores de banda média consumida em cada *link* da árvore, sendo essa banda expressa em número de fluxos. Nesta dissertação, assume-se que o custo de transmissão de um fluxo em um *link* é unitário, igual para todos os *links*. Isso é uma simplificação que pode ser facilmente estendida para tratar o caso em que os *links* possuem custos diferentes.

Um grupo de estratégias para prover qualidade de serviço através do roteamento é fundamentado na suposição de que há diversidade de caminhos entre servidor e cliente, isto é, há

mais de um caminho possível e com custo razoável entre servidor e cliente. Essas estratégias utilizam técnicas de envio por múltiplos caminhos simultaneamente ou de troca de caminho em caso de contenções no caminho corrente. Dessa forma, o consumo de banda de rede excessivo inerente à mídia contínua, que prejudica a qualidade de serviço, é amenizado ao ser dividido entre vários caminhos. Esse é o princípio que permite que essa abordagem reduza atraso e perda de pacotes.

Os trabalhos que exploram entrega por múltiplos caminhos simultaneamente [7, 6, 24] tentam dar garantia estatística de que uma certa fração dos pacotes enviados chegará ao destino. Essa garantia, associada à codificação de vídeo em múltiplas camadas, leva a um aumento na qualidade de serviço percebida pelo usuário final.

O segundo grupo de estratégias explora o uso de protocolos de transmissão escaláveis (baseados em compartilhamento de fluxos), otimizando o roteamento dos fluxos compartilhados entre servidor e clientes, visando minimizar ainda mais o consumo de banda de rede. Dessa forma, espera-se obter uma redução na carga sobre a rede e a possibilidade de existência de pontos de contenção de tráfego que piorem a qualidade de serviço. Além de utilizar protocolos de transmissão escaláveis baseados em compartilhamento de fluxos que reduzem a banda de servidor e, como conseqüência, reduzem também a banda de rede, é possível também otimizar protocolos de roteamento de forma a reduzir ainda mais a banda de rede. Para o caso dos protocolos otimizados, numa árvore de distribuição, cada nó tem a possibilidade de compartilhar fluxos com seus ancestrais — daí a possibilidade de redução no consumo de banda.

O protocolo *default* usado no roteamento entre dois *sites* na Internet, o BGP [37], realiza o roteamento através do menor caminho, considerando-se a forma mais simplificada de implementação desse protocolo. Assim, a forma *default* de criar uma árvore de distribuição a partir de determinado servidor é ligar cada cliente ao servidor através do caminho mais curto. Essa abordagem minimiza o custo (assumindo-se custo unitário em cada *link*) de transmissão para *unicast*, na qual a criação dos fluxos é feita de forma independente e separada para cada cliente. A mesma abordagem também foi proposta como padrão da Internet em [21, 41] para árvores de distribuição com compartilhamento de fluxos. Entretanto, para este caso ela pode não ser ótima.

Tomemos como exemplo a topologia na figura 1.1, onde S é um servidor de VoD, R_i são roteadores que permitem compartilhamento de fluxos e A e B são *sites* clientes. Como simplificação, suponhamos que o custo de transmissão seja unitário para todos os *links* e que há apenas um usuário em cada um desses *sites* assistindo ao filme e eles estejam compartilhando fluxos de rede. O caminho oferecido pelo roteamento da Internet (isto é, o caminho mais curto) entre S e A é $SR_1R_2R_3A$, enquanto o caminho para B é $SR_1R_4R_5B$. Se A e B estão assistindo a um mesmo vídeo, o custo de transmissão total seria 7 fluxos, isto é, o custo de S até R_1 mais o custo de R_1 até A mais o custo de $R1$ até B. Se, no entanto, utilizássemos uma árvore alternativa na qual o caminho de S até A fosse o mesmo, mas o caminho de S até B fosse $SR_1R_2R_3R_5B$, teríamos um custo de transmissão total igual ligeiramente menor, igual a 6 fluxos[1].

[1] Veremos que caso as demandas de A e B não fossem unitárias, mas houvesse vários usuários em cada um

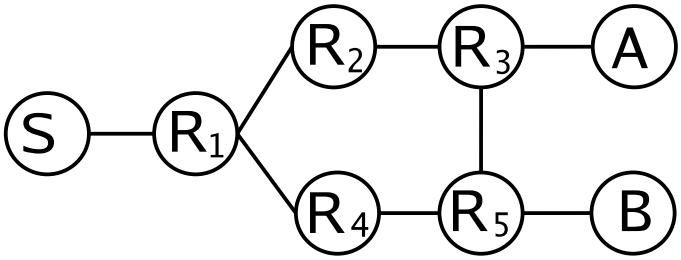


Figura 1.1: Uma topologia simples.

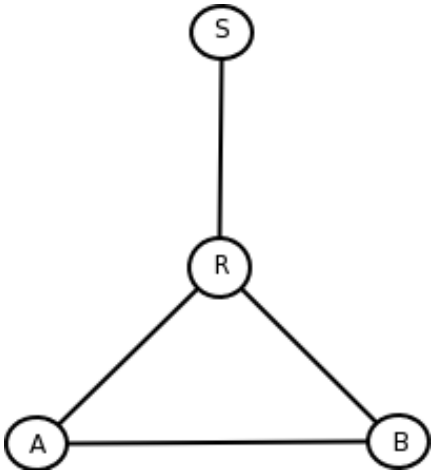


Figura 1.2: Topologia canônica com um servidor S, um roteador R, e dois *sites* clientes A e B

Dependendo da relação entre os custos de entrega na parte compartilhada e nas partes disjuntas, a entrega pelo caminho mais curto, que é o oferecido pela Internet, pode ser significativamente pior que o roteamento ótimo. Essa relação é formalizada por Almeida [2], para uma topologia simples canônica como a mostrada na figura 1.2.

Existem trabalhos [4, 70, 22] propondo heurísticas para otimizar o roteamento de fluxos compartilhados. Avaliações preliminares realizadas nesses trabalhos mostram que pode haver um ganho em termos de consumo de banda de rede média quando se usam essas heurísticas ao invés do roteamento *default*. Geralmente, a avaliação dessas heurísticas se dá através de simulações, as quais devem receber como entrada, idealmente, mapas topológicos precisos da Internet. Entretanto esses trabalhos avaliaram esses protocolos apenas com topologias sintéticas e possivelmente não realistas [70] ou com um pequeno número de topologias reais [4]. Logo, os resultados obtidos não são conclusivos.

A questão central para os dois grupos de estratégias de roteamento para melhoria da qualidade de serviço em mídia contínua é a diversidade de caminhos. Os protocolos de roteamento para compartilhamento de fluxos tentam criar árvores de distribuição entre o servidor e os

desses *sites* assistindo a um mesmo filme, a redução no custo poderia ser bem maior.

clientes alternativas à árvore dos menores caminhos e que tenham um custo total de entrega mais baixo. Isso implica na necessidade de outros caminhos viáveis entre o servidor e o cliente, além do caminho mais curto. Já com relação aos protocolos de roteamento através de múltiplos caminhos, é óbvia a necessidade de existência de diversidade de caminhos.

Apesar dos esforços para caracterizar essa diversidade [59], existe ainda uma carência de caracterização mais completa da diversidade de caminhos na Internet, analisando topologias diversas, em vários níveis de dispersão. Apenas com essa caracterização pode-se avaliar a possibilidade de aplicação prática de técnicas que visam a melhoria da qualidade da entrega através do aproveitamento da diversidade de caminhos.

Para realizar essa caracterização, bem como para realizar uma simulação extensa e mais metódica dos protocolos de roteamento, faz-se necessária a coleta de dados que permitam recriar essas topologias reais. Essa base de dados é importante não apenas para permitir a caracterização de topologias reais da Internet em termos de diversidade de caminhos, mas também para fomentar diversas outras pesquisas futuras que necessitem de testes em topologias realistas.

Tendo em vista que tanto as avaliações das heurísticas de roteamento alternativo para compartilhamento de fluxos quanto a própria caracterização da diversidade de caminhos da Internet realizadas em trabalhos anteriores foram limitadas, esta dissertação visa avançar esse conhecimento, através de análises mais abrangentes e detalhadas.

1.3 Objetivos

Dadas as limitações dos trabalhos realizados até o momento, esta dissertação tem por objetivo analisar mais a fundo os ganhos potenciais da aplicação de técnicas alternativas para roteamento de mídia contínua na Internet atual.

Para tanto, fazem-se necessários o mapeamento de topologias reais, sua caracterização quanto à diversidade de caminhos e a sua utilização em simulações de protocolos para criação de árvores de distribuição para compartilhamento de fluxos.

No mapeamento das topologias reais, além de melhorar a precisão dos mapas, outras melhorias a serem feitas em relação ao trabalho de Spring et al [56] incluem, por exemplo, a coleta de mapas em áreas mais abrangentes. Além disso, do ponto de vista deste trabalho, não se está interessado em provedores de acesso (ISP) específicos, mas sim na Internet em geral. Por esse motivo, utilizamos neste trabalho uma metodologia semelhante à adotada por Spring et al [56], porém com pontos de coleta espalhados por todo o mundo e corrigindo as fontes de imprecisão apontadas por Teixeira et al [59].

Dada a importância da diversidade de caminhos para as técnicas alternativas de roteamento, realizamos uma caracterização dos mapas coletados, utilizando dois parâmetros, a saber, o número de caminhos diferentes e o grau de diferença médio entre esses caminhos, levando em conta o fato de que não é necessário haver disjunção completa entre os caminhos que ligam dois *sites*.

Por fim, usamos as topologias coletadas, mais precisas e diversificadas, para estender o trabalho de Almeida et al [4, 2], através da realização de um número bem maior de simulações de dois protocolos otimizados de roteamento de fluxos compartilhados e em topologias de diversos graus de abrangência espalhadas por todo o mundo. Os diversos parâmetros, como a heterogeneidade de demanda, são variados de forma mais sistemática.

1.4 Contribuições

A seguir listamos as principais contribuições deste trabalho.

- Coleta e tratamento de dados de rotas reais entre pontos espalhados pela Internet, bem como geração de vários mapas topológicos, dados esses úteis para o nosso trabalho, mas que também podem ser aproveitados por outros grupos de pesquisa.
- Caracterização de topologias reais para verificar a potencialidade da aplicação de técnicas já existentes que permitiriam melhorar a qualidade de serviço e/ou prover economia de recursos na entrega de mídia contínua.
- Análise extensa e mais sistemática das potencialidades de ganho de banda média com o uso de protocolos alternativos ao caminho mais curto que exploram compartilhamento de fluxos para roteamento de mídia contínua, bem como comparação entre esses protocolos em termos do ganho obtido por cada um deles.

1.5 Organização desta dissertação

Esta dissertação está organizada da seguinte forma. No capítulo 2, explicamos os principais conceitos envolvidos nesta dissertação, estabelecendo nosso foco de trabalho. Em seguida, no capítulo 3 revisamos os trabalhos relacionados, detalhando os que serão estendidos nesta dissertação. No capítulo 4 apresentamos a coleta e filtragem dos dados necessários tanto à caracterização da Internet quanto à simulação dos protocolos de roteamento de mídia contínua. Em seguida, no capítulo 5, apresentamos a caraterização das topologias coletadas quanto a métricas importantes para a aplicação de técnicas alternativas de roteamento de mídia contínua. No capítulo 6 realizamos simulações dos protocolos de criação das árvores de distribuição sobre as topologias reais coletadas e analisamos as árvores criadas. Por fim, no capítulo 7, apresentamos as conclusões obtidas e estabelecemos possibilidades de melhorias ou extensões neste trabalho.

Capítulo 2

Definição do Problema

Este capítulo apresenta as principais definições acerca do problema de roteamento de mídia contínua, estabelecendo o foco deste trabalho. Definimos roteamento de uma maneira geral e mostramos como ocorre o roteamento hoje na Internet. Em seguida apresentamos a principal abordagem para roteamento de mídia contínua para múltiplos clientes: a criação de árvores de distribuição com compartilhamento de fluxos de rede. Mostramos a importância da diversidade de caminhos para tal abordagem, justificando assim a medição desse aspecto em topologias reais. Mostramos a diferença entre topologias em nível de roteadores e em nível de sistemas autônomos, apontando o motivo pelo qual focamos no nível de roteadores.

2.1 Roteamento entre dois pontos na Internet

Definimos **roteamento** como o processo de determinar que caminho, em uma rede de computadores, certo conjunto de dados deve percorrer ao sair de um nó origem e chegar a um nó destino. A unidade básica de transferência de dados através da Internet é o pacote IP, assim podemos falar em roteamento de certo pacote de dados. O tamanho do pacote IP é limitado. Na maioria das vezes, o conjunto total de dados a ser transferido entre dois nós na rede é maior que esse limite, portanto esses dados devem ser fragmentados em vários pacotes e enviados através da rede. Enxergando isso como um processo contínuo, faz sentido dizer que existe um **fluxo** de pacotes saindo do nó origem e chegando ao nó destino. Dependendo da forma como é feito o roteamento desses pacotes, é possível observar na rede um caminho, em termos de *links* da rede, pelo qual passa esse fluxo. Em outros casos, em que o roteamento é feito pacote a pacote e existe mais de um caminho possível entre origem e destino, não é possível precisar o caminho exato pelo qual um fluxo percorre a rede. Por possuir diversos caminhos entre cada par de nós e utilizar um esquema de roteamento dinâmico, a Internet cai nessa segunda categoria.

Uma *internet* é a interconexão, utilizando dispositivos denominados roteadores, de duas ou mais redes físicas distintas. A Internet atual pode ser vista como a interconexão das *internets* de diversas organizações, tais como universidades, centros de pesquisa, provedores de acesso e/ou de trânsito etc. Essas organizações são denominadas **sistemas autônomos**

e a cada uma delas é atribuído um número **identificador** único. A cada sistema autônomo é delegada a responsabilidade sobre uma certa fração do conjunto total de endereços IP, endereços estes que devem ser atribuídos a cada nó (computador, dispositivos móveis etc.) que deseja participar da Internet.

Cada sistema autônomo naturalmente possui liberdade para determinar como se dá o roteamento dentro de sua *internet*. Dependendo das dimensões e da finalidade dessa rede, o roteamento pode ser feito de forma estática, através da configuração manual de tabelas de roteamento, ou de forma dinâmica, através de algum protocolo específico para isso. Protocolos usados no roteamento dentro de um mesmo sistema autônomo são denominados *interior gateway protocols*, ou IGP. Para rotear pacotes entre sistemas autônomos distintos, no entanto, é interessante que todos os sistemas autônomos saibam se comunicar através de um protocolo comum. O protocolo padrão que serve a essa função na Internet atual é o *Border Gateway Protocol*, ou **BGP** [37]. Esse nome vem do fato de serem os roteadores de borda que se comunicam através desse protocolo. Um roteador de borda, em um sistema autônomo, é um roteador que possui uma conexão física direta com outro roteador de borda em um sistema autônomo adjacente. O tráfego flui entre sistemas autônomos sempre através desses roteadores.

Cada roteador de borda mantém uma tabela (atualizada periodicamente através da comunicação com roteadores de borda vizinhos), denominada **tabela BGP**, que relaciona endereços IP destino a rotas, estas dadas em termos de identificadores de sistemas autônomos. Na verdade, uma otimização é feita aproveitando-se do fato de os endereços IP serem alocados aos sistemas autônomos em faixas contíguas, geralmente de mesmo prefixo (mesmos dígitos nos campos mais significativos). Assim, as tabelas BGP relacionam prefixos destino a rotas. Ao receber um pacote com destino a um endereço de dado prefixo, o roteador de borda escolhe uma das rotas capazes de alcançar o sistema autônomo que detém aquele endereço, e envia o pacote para o roteador adjacente que corresponder a essa rota. Geralmente, em conformidade com a sugestão apresentada no padrão de funcionamento do BGP [37], escolhe-se o caminho mais curto, em termos de número de sistemas autônomos a serem percorridos.

2.2 Roteamento a partir de uma origem até diversos destinos

Muitas vezes, especialmente no contexto de mídia contínua, um nó origem quer enviar os mesmos dados para diversos nós destino. Isso pode ser feito, naturalmente, através da criação de um fluxo até cada nó destino, partindo do nó de origem. Por simplificação, podemos assumir que cada fluxo percorre um caminho fixo até seu nó destino. O conjunto dos caminhos até cada um dos nós destino pode ser visto como uma árvore, cuja raiz é o nó origem. A desvantagem dessa abordagem é o desperdício de banda de rede, já que haverá n fluxos idênticos passando por cada *link* que sirva a uma subárvore de n clientes. Se os fluxos são idênticos, seria possível enviar apenas uma cópia deles através de tal *link*, e o roteador na outra ponta recriaria as n cópias.

Baseando-se nessa idéia, existem esquemas de roteamento que permitem que um único

fluxo enviado por um nó origem chegue a diversos nós destino. Isso é feito através da "bifurcação" de um fluxo que chega a dado nó no interior da rede, na qual esse nó envia cópias dos dados recebidos para dois ou mais outros nós adjacentes. A esses esquemas de roteamento mais econômicos chamamos **roteamento com compartilhamento de fluxos** de rede (ou *multicast*). Já o esquema menos econômico, no qual cada cliente possui um fluxo de rede totalmente dedicado desde a origem, é chamado ***unicast***.

No contexto de mídia contínua, pode-se utilizar roteamento com compartilhamento de fluxos quando um servidor envia um mesmo objeto de mídia contínua (por exemplo, um vídeo) a diversos clientes espalhados pela rede. Estendendo essa idéia, pode haver vários servidores espalhados pela rede, cada um servindo a certo grupo de clientes, conforme veremos a seguir.

2.2.1 Floresta de distribuição

Uma floresta de distribuição consiste em um conjunto fixo de m servidores, um conjunto fixo de *sites* clientes (ou simplesmente clientes) com demandas fixas e um conjunto também fixo de caminhos sobre a rede IP, cada caminho ligando um servidor a um cliente. Cada cliente está ligado a exatamente um servidor. Por simplicidade, assume-se que uma floresta corresponde a um único objeto (p.ex. determinado vídeo) sendo requisitado pelos clientes e fornecido pelos servidores. Assume-se também que tal objeto está completamente replicado em cada um dos servidores, razão pela qual chamaremos os servidores também de **réplicas**.

A demanda N_i de um *site* cliente i é o agregado das demandas dos usuários dessa rede. Essa demanda é dada em termos de número de requisições efetuadas por aquele *site* cliente em uma unidade de tempo igual ao tamanho (tempo de exibição) do objeto sendo requisitado. Por exemplo, supondo que o objeto em questão seja um vídeo de 90 minutos, se os usuários do *site* cliente i requisitarem a uma taxa de 10 requisições por minuto, então serão feitas em média 900 requisições durante o período de 90 minutos (que é o tamanho do objeto sendo requisitado). Assim, neste caso, temos $N_i = 900$.

2.2.2 Cálculo do custo

Apesar de o cálculo da banda média de rede[1] ser trivial para uma floresta de distribuição que use *unicast*, para compartilhamento de fluxos o cálculo não é tão óbvio. Nesta seção mostramos como se realiza esse cálculo.

Para o cálculo do custo de rede, assumimos, que o custo de transmissão através de qualquer *link* é igual a 1. Isso é apenas uma simplificação, a análise poderia facilmente ser estendida para a situação em que os custos dos *links* são heterogêneos. Assim, o custo de rede é igual à banda média de rede, dada pela soma das bandas médias em cada um dos *links* da floresta de distribuição. A banda média em um *link* é dada, no contexto desse trabalho, pelo número médio de fluxos de rede passando por tal *link*.

[1] Banda **consumida**, não banda disponível. Nesta dissertação, toda vez que nos referirmos à banda de rede, estaremos falando no sentido de banda consumida. Assume-se que nunca há restrições na banda disponível em qualquer dos *links*.

Para *unicast*, a banda de servidor varia linearmente com a demanda total da floresta, já que não há compartilhamento de fluxos. Pelo mesmo motivo, a banda média de rede em cada *link* também varia linearmente com a demanda da subárvore servida por tal *link*. Nota-se, portanto, que, com o uso de *unicast*, a banda de rede total da floresta varia linearmente com a demanda e também com a distância dos servidores até seus clientes.

Com o uso de compartilhamento a variação **não** é linear com a demanda. Eager et al [20] derivaram um limite inferior teórico para a banda média no caso em que se usa um protocolo ótimo para compartilhamento de fluxos com entrega imediata. A banda média para um servidor [20] ou para um *link* [70] que sirva a uma demanda total de N, usando esse protocolo ótimo, é dada por:

$$ln(N+1)$$

Assim, em florestas que usam compartilhamento de fluxos, a banda média de rede total varia logaritmicamente com a demanda e linearmente com a distância dos servidores até seus clientes. A banda de servidor varia logaritmicamente com a demanda.

Foi mostrado, também em [20], que o uso do protocolo *Bandwidth Skimming* (a ser definido no capítulo 3), com banda de cliente limitada a 2 vezes a taxa de bits usada na exibição consegue atingir um desempenho muito próximo do limite inferior teórico. Por esse motivo, sempre que nos referirmos ao custo de roteamento para compartilhamento de fluxos, estaremos assumindo o limite inferior teórico acima.

Note que essa diferença entre os custos de roteamento usando *unicast* e usando compartilhamento de fluxos faz com que o roteamento ótimo para *unicast* não seja necessariamente ótimo para fluxos compartilhados. Por exemplo, em *unicast*, não há diferença, em termos de banda média de rede, se há 1 *link* ligando o servidor até um cliente de demanda $2N$ ou se há 2 *links* ligando um servidor até um cliente de demanda N. Já para compartilhamento de fluxos, o custo é menor no primeiro caso, pois $ln(2N+1) < 2 \times ln(N+1)$.

2.2.3 Florestas mais econômicas e a diversidade de caminhos

Uma forma simples de se criar uma floresta de distribuição, dados uma rede, um conjunto de réplicas e um conjunto de *sites* clientes com suas respectivas demandas, é ligar cada cliente até a réplica mais próxima através do caminho mais curto. Com o uso de compartilhamento de fluxos de rede, no entanto, esse esquema pode não ser ótimo. Almeida [2] mostra que pode existir uma penalidade (em termos de banda de rede média necessária) associada à utilização de um esquema tão simples, dependendo das características da rede existente entre as réplicas e os clientes. Mostra ainda que outros protocolos ligeiramente mais sofisticados permitem obter um roteamento, em muitas situações, próximo do ótimo. Assim, um dos problemas na área de roteamento de mídia contínua, foco deste trabalho, é determinar qual é o melhor protocolo para se construirem florestas de distribuição na Internet atual. Alguns desses protocolos alternativos são apresentados na seção 3.3.2. Já o capítulo 6 estuda os ganhos que poderiam ser obtidos, na Internet atual, com o uso de tais protocolos.

A possibilidade de criação de florestas de distribuição alternativas à do caminho mais curto depende, naturalmente, da **topologia de rede** existente entre os *sites* participantes. Quando se diz topologia de rede, está-se referindo aos nós de uma rede e quais nós estão ligados a quais outros nós. Assim, se existem vários caminhos possíveis entre os pares de *sites* clientes ou réplicas que participarão de dada floresta de distribuição, então há várias florestas de distribuição diferentes que podem ser criadas, abrindo possibilidades para a existência de florestas de menor custo que a dos caminhos mais curtos. O número de caminhos existentes entre os pares de *sites* são uma característica topológica que denominamos **diversidade de caminhos**. O capítulo 5 estuda essa característica em topologias reais da Internet.

Assim, para estudar roteamento de mídia contínua na Internet atual, é necessário usar mapas topológicos correspondentes à Internet real, tanto para medir a diversidade de caminhos presente nessas topologias quanto para simular a criação de florestas de distribuição utilizando protocolos alternativos ao caminho mais curto.

2.3 Mapas topológicos

A Internet, conforme explicamos na seção 2.1, pode ser vista como um conjunto de sistemas autônomos interconectados. Uma tabela BGP contém diversos caminhos dados em termos de identificadores de sistemas autônomos. Assim, a partir de tal tabela, é possível construir um grafo no qual os vértices correspondem a sistemas autônomos e as arestas correspondem a *links* entre roteadores de borda de sistemas autônomos adjacentes. Chamamos esse grafo de mapa topológico da Internet **ao nível de sistemas autônomos**.

Quando se fala em roteamento, pode-se dizer que a *internet* de cada sistema autônomo é essencialmente composta de roteadores que interconectam as diversas redes físicas desse sistema autônomo e/ou conectam redes desses sistemas a redes de sistemas vizinhos. Portanto, outra forma de enxergar a Internet é como um grafo em que vértices correspondem a roteadores e arestas correspondem a *links* entre esses roteadores. Esse grafo é bem mais detalhado que o mapa ao nível de sistemas autônomos, uma vez que cada sistema autônomo possui, em geral, diversos roteadores. Chamamos esse grafo de mapa topológico da Internet **ao nível de roteadores**.

Ao contrário do que ocorre com o caso do mapa topológico ao nível de sistemas autônomos, não existe uma tabela de roteamento que liste as conexões entre todos os roteadores da Internet. Assim, mapear a Internet ao nível de roteadores envolve outras técnicas, sendo a principal delas o uso de ***traceroute***.

O *traceroute* é um programa que tenta obter os endereços IP de todos os roteadores existentes no caminho entre dois nós da Internet. Ele funciona combinando dois artifícios: o campo *time-to-live* (TTL) dos pacotes IP e a existência de um protocolo notificação de erros denominado *Internet Control Messages Protocol* (ICMP). Esse protocolo é implementado pela grande maioria dos roteadores da Internet. O campo TTL é um campo no pacote IP que é decrementado por cada roteador no caminho de um pacote. Caso o valor desse campo atinja o zero, o roteador descarta o pacote e envia uma notificação de erro ICMP ao nó origem,

indicando que o pacote percorreu um caminho muito longo e não conseguiu chegar ao destino (isso pode ocorrer em casos de falhas ou mudanças no roteamento). Assim, se o programa *traceroute* sendo executado em uma máquina A quer descobrir os endereços IP de todos os roteadores existentes entre A e B, ele inicialmente envia um pacote com destino a B mas com TTL igual a 1. O primeiro roteador R_1 no caminho decrementará o campo TTL e notificará A, com um pacote ICMP, que o pacote não conseguiu chegar até o destino B. Esse pacote ICMP conterá, no seu campo origem, o endereço[2] IP do roteador R_1. Em seguida A enviará um pacote com destino a B e com TTL igual a 2. O segundo roteador R_2 no caminho entre A e B responderá com um pacote ICMP. Daí poderemos concluir que existe um *link* entre R_1 e R_2, se assumirmos que o segundo pacote tomou o mesmo caminho que o primeiro[3]. O programa rodando em A segue enviando pacotes com TTL incrementais para descobrir todos os roteadores no caminho entre A e B.

A seção 3.4.2 detalha como os dados de *traceroute* podem ser usados para criar mapas topológicos ao nível de roteadores.

2.3.1 Topologias usadas neste trabalho

Diversos trabalhos que estudam protocolos de rede realizam simulações em grafos correspondentes a topologias reais da Internet. Parte deles utiliza topologias ao nível de sistemas autônomos, que podem ser obtidas, por exemplo, através de tabelas BGP [37] extraídas de roteadores de borda. Outra parte desses trabalhos utiliza topologias mais detalhadas, ao nível de roteadores, isto é, ao nível de conectividade IP. Tais topologias podem ser obtidas através de medidas ativas entre diversos pontos espalhados pela Internet. O tipo de medida ativa mais comum é a gerada pelo comando *traceroute*. Em casos em que se procura analisar a banda média de rede gasta por determinado protocolo de distribuição de dados, ou mais precisamente, o ganho potencial máximo que se pode obter com tal protocolo, em geral precisa-se de um grau maior de precisão na topologia e assim preferem-se topologias no nível de roteadores. Isso porque em topologias ao nível de sistemas autônomos não é possível determinar o gasto com banda de rede no interior dos sistemas autônomos, uma vez que não se sabe por quantos *links* se passa ao atravessar um sistema autônomo e essa quantidade de *links* atravessados pode variar bastante de um sistema autônomo para outro.

Considere por exemplo a figura 2.1, que mostra em nível de sistemas autônomos e em nível de roteadores a topologia existente entre um servidor S e dois *sites* clientes A e B. Os quatro sistemas autônomos estão delimitados por elipses rotuladas A_i. Vimos, na seção 1.2, que é possível criar uma árvore alternativa à dos caminhos mais curtos nessa topologia e obter uma economia em termos de banda de rede consumida. Se, no entanto, analisarmos a mesma

[2] Mais precisamente, um dos endereços IP do roteador.

[3] Infelizmente não é tão incomum que a rota mude. Isso reduz a confiabilidade do uso de *traceroutes* para inferir a topologia ao nível de roteadores, por isso é necessário que se execute o comando várias vezes para tentar filtrar estatisticamente os falsos caminhos decorrentes de mudança no roteamento, como veremos no capítulo 4.

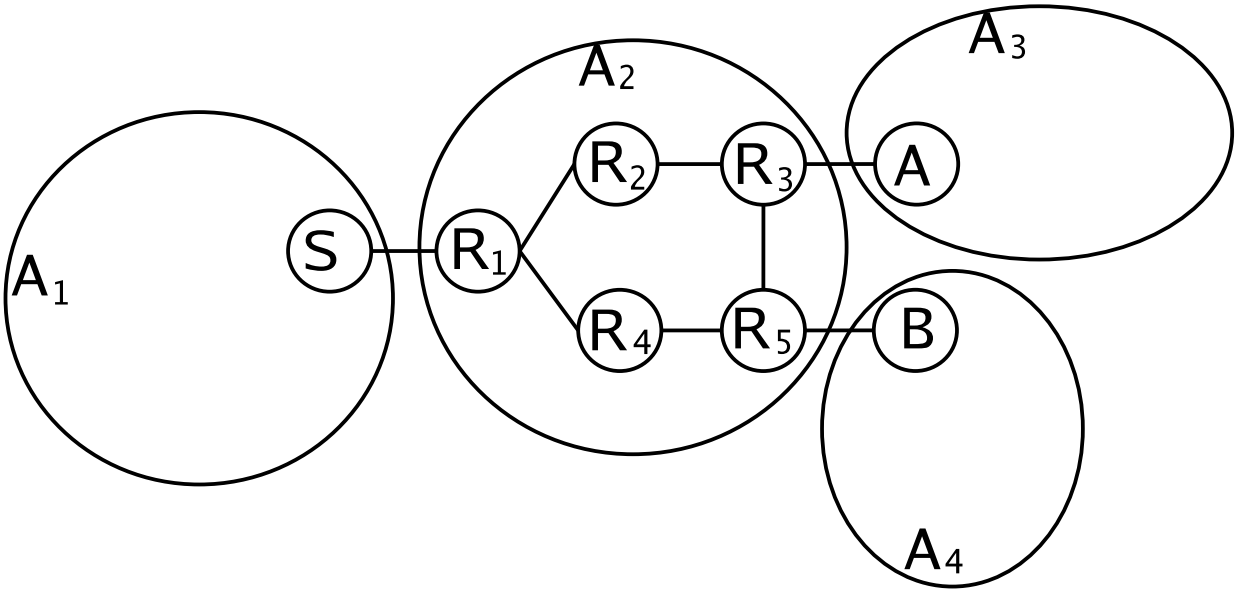


Figura 2.1: Uma topologia simples aos níveis de sistemas autônomos e de roteadores.

topologia ao nível de sistemas autônomos, vemos que existe apenas uma árvore possível: o *site* A estaria ligado ao servidor S através do caminho SA_2A e o *site* B pelo caminho SA_2B.

Assim, o menor detalhamento da topologia ao nível de sistemas autônomos pode esconder ganhos potenciais obteníveis com o uso de florestas de distribuição alternativas à dos caminhos mais curtos. Por esse motivo, neste trabalho focamos em topologias ao nível de roteadores. O capítulo 4 mostra como essas topologias foram obtidas.

2.3.2 Topologias sintéticas

A obtenção de topologias reais, em nível de roteadores, para a análise dos protocolos de criação de florestas de distribuição não é uma tarefa fácil. Por esse motivo, este trabalho utiliza-se também de topologias sintéticas ao nível de roteadores. O GT-ITM [12] é um programa capaz de sintetizar topologias ao nível de roteadores. Tentando tornar as topologias mais realistas, o GT-ITM inicialmente cria uma topologia em nível de sistemas autônomos, de acordo com parâmetros que controlam o número de nós e a sua conectividade. Esses sistemas autônomos são classificados em de trânsito e *stubs*. Os sistemas autônomos de trânsito correspondem, na Internet real, a sistemas autônomos de provedores de tráfego, que formam o *backbone* da Internet. Sistemas autônomos *stub* são aqueles que não provêem tráfego para outros sistemas autônomos, mas sim apenas aceitam tráfego destinado a eles mesmos e enviam tráfego originado neles mesmos. O GT-ITM procura criar topologias mais realistas ao refletir a estrutura hierárquica existente entre provedores de tráfego e sistemas *stub*.

Após gerar essa topologia em nível de sistemas autônomos, o GT-ITM utiliza outro conjunto de parâmetros similar ao primeiro, fornecido pelo usuário, para transformar cada nó

correspondente a um sistema autônomo em um outro grafo, este correspondente à *internet* (isto é, a rede ao nível de roteadores) desse sistema autônomo.

Neste trabalho utilizamos, além de topologias reais, topologias ao nível de roteadores geradas pelo GT-ITM para analisar protocolos para criação de árvores de distribuição, como veremos no capítulo 6.

2.4 Símbolos usados ao longo dessa dissertação

A tabela 2.1 relaciona os símbolos usados com maior freqüência ao longo desta dissertação.

D e $d_{X,Y}$	Distância média, respectivamente entre um conjunto de *sites* e entre dois *sites* X e Y.
C e $c_{X,Y}$	Número de caminhos diferentes, respectivamente entre um conjunto de *sites* e entre dois *sites* X e Y.
G e $g_{X,Y}$	Grau de diferença médio, respectivamente entre um conjunto de *sites* e entre dois *sites* X e Y.
A e $a_{X,Y}$	Grau de assimetria médio, respectivamente entre um conjunto de *sites* e entre dois *sites* X e Y.
V	conjuntos de nós em uma rede (*sites* e roteadores)
i	identificador de um nó da rede
N_i	demanda do *site* cliente i
$Clientes$	conjunto de *sites* clientes
m	número de réplicas em uma floresta de distribuição
n	número de *sites*, incluindo réplicas e clientes
$Sites1$ e $Sites2$	Partição do conjunto de *sites* da floresta. $Sites1$ é o conjunto de menor diâmetro, $Sites2$ os *sites* restantes.

Tabela 2.1: Símbolos utilizados nesta dissertação

2.5 Conclusão

Esta dissertação foca no roteamento de mídia contínua em topologias ao nível de roteadores da Internet atual[4], coletadas segundo a metodologia descrita no capítulo 4. Isso é feito através da análise da diversidade de caminhos (capítulo 5) que permita a criação de árvores de distribuição alternativas, bem como da análise de protocolos (capítulo 6) para a criação de tais árvores. Para este último caso, assume-se que é possível criar árvores arbitrárias nas topologias reais, ainda que isso não seja uma prática na Internet atual como um todo. Está fora do escopo deste trabalho determinar como o roteamento com compartilhamento de fluxos se daria na estrutura atual da Internet, descentralizada, composta por sistemas autônomos distintos, cada um deles realizando o roteamento interno como bem entende e deliberadamente

[4] Ainda que os resultados obtidos possam muitas vezes ser estendidos ao uso de roteamento em redes *overlay*.

aceitando ou rejeitando tráfego de sistemas autônomos vizinhos. Isso é objeto de estudo para trabalhos futuros.

Antes de mostrarmos como foi o nosso trabalho de coleta, análise da diversidade de caminhos e dos protocolos de roteamento, apresentamos, no capítulo 3, os trabalhos relacionados importantes para o entendimento da nossa metodologia.

Capítulo 3

Trabalhos Relacionados

Este capítulo discute trabalhos anteriores direta ou indiretamente relacionados com esta dissertação. A seção 3.1 apresenta trabalhos feitos no sentido de caracterizar a diversidade de caminhos, bem como trabalhos que mostram como a diversidade de caminhos pode ser aproveitada para melhorar a qualidade de serviço. A seção 3.2 apresenta os principais protocolos propostos para entrega com compartilhamento de fluxos, que focam em reduzir a banda de servidor e conseqüentemente diminuem também a banda de rede. Protocolos para roteamento e localização de réplicas são apresentados na seção 3.3. Por fim, na seção 3.4, apresentamos trabalhos relacionados que dizem respeito à coleta de dados para mapeamento de topologias reais.

3.1 Diversidade de caminhos

A existência de múltiplos caminhos, mesmo que com trechos compartilhados, entre pares de *sites* da Internet pode ser usada para melhorar a qualidade de serviço em aplicações de transmissão de mídia contínua. Diversos trabalhos [6, 5, 24, 58] estudam a questão do envio de dados através de múltiplos caminhos ou de caminhos alternativos ao padrão, ou caracterizam a diversidade de caminhos. Esta seção descreve alguns deles.

Apostolopoulos [6] estuda uma alternativa à codificação tradicional de vídeo, propondo a codificação em múltiplas camadas de um mesmo vídeo. Os pacotes referentes a cada camada poderiam ser enviados através de caminhos distintos. Essas camadas são recombinadas no cliente. Para cada quadro de vídeo, quanto mais camadas correspondentes àquele quadro tiverem sido recebidas, maior será a qualidade de exibição daquele quadro. Se o cliente receber pelo menos uma camada de dado quadro, já é possível exibir aquele quadro, ainda que com qualidade baixa. O vídeo resultante, combinando todas as camadas, possui uma taxa de bits apenas um pouco maior que o vídeo original de alta qualidade.

Andersen et al [5] aproveitam a existência de diversidade de caminhos criando redes *overlay* para prover alternativas ao roteamento tradicional da Internet e enviando dados por caminhos menos congestionados.

Golubchik et al [24] estudam a melhoria na qualidade de serviço de uma aplicação que

utilize múltiplos caminhos simultaneamente para enviar vídeo através da Internet, em relação à tradicional abordagem de envio através de um único caminho. Através de modelos analíticos, é analisado o impacto da utilização de múltiplos caminhos considerando-se as seguintes métricas: taxa de perdas de dados, distribuição dos comprimentos de rajadas de perdas e autocorrelação entre os pacotes perdidos. Os autores argumentam que essas métricas podem ser utilizadas para refletir o grau de qualidade de serviço percebido pelo usuário de uma aplicação de vídeo sob demanda. A conclusão é que, em geral, a utilização de múltiplos caminhos melhora as características de perda em relação à utilização de um único caminho. Isso incentiva a caracterização da diversidade de caminhos existente em redes reais.

Apesar de motivar o projeto dos protocolos acima mencionados, a diversidade de caminhos em topologias reais da Internet foi analisada, até agora, de forma limitada. Teixeira et al [58] tentam caracterizar a diversidade de caminhos de ISPs norte-americanos e europeus. Os mapas topológicos desses ISPs foram obtidos pelo projeto Rocketfuel [57] e um deles foi fornecido pelo próprio ISP. Ao comparar o mapa real desse ISP com o mapa correspondente inferido pelo Rocketfuel, os autores constatam que os mapas do Rocketfuel não são exatos o suficiente para se realizar uma caracterização em termos de diversidade de caminhos. Por isso, caracteriza-se apenas a diversidade de caminhos a partir do mapa real, mostrando que, para tal ISP, existem pelo menos dois caminhos totalmente disjuntos de *links* entre 90% dos pares de pontos-de-presença (PoP). Assim, ao menos em tal provedor, revela-se uma alta diversidade de caminhos. No entanto, caracterizar a diversidade de caminhos em termos do número de caminhos completamente disjuntos é desnecessariamente restritivo, já que o importante é apenas que os caminhos sejam disjuntos nos *links* gargalo [52] [1]. Por esse motivo, utilizamos outras métricas, descritas no capítulo 5, para caracterizar a diversidade de caminhos da topologia que coletamos.

3.2 Compartilhamento de fluxos

Diversos protocolos foram propostos para diminuir a banda de servidor em aplicações de mídia contínua, através do uso de compartilhamento de fluxos. Apesar de focarem na redução da banda do servidor, como efeito colateral ocorre também uma redução na banda de rede [2], já que fluxos de rede também estão sendo compartilhados por corresponderem diretamente aos fluxos saindo do servidor. Esses protocolos, denominados protocolos para transmissão com compartilhamento de fluxos, assumem que seja possível dois ou mais clientes escutarem um

[1] Ao se dividir entre dois caminhos um conjunto de dados a serem transmitidos, contornando assim o gargalo que existiria caso se utilizasse um único caminho, naturalmente poderá haver *links* gargalo em cada um desses dois caminhos. Mesmo assim, essa abordagem permite aumentar a banda passante total entre a origem e o destino, bem como melhorar a qualidade de serviço ao aliviar a carga total sobre cada caminho, conforme mostram Golubchik et al [24].

[2] Essa redução na banda de rede será observada no mínimo em 1 *link*: aquele existente entre o servidor e o *gateway* com o qual ele se conecta à Internet. No caso de uma topologia em estrela, esse será o único *link* em que se observará ganho em termos de banda de rede consumida.

mesmo fluxo emitido pelo servidor e determinam quais clientes devem escutar quais fluxos e quando um novo fluxo deve ser criado.

Batching [15] é um protocolo que não provê serviço imediato, isto é, quando um cliente requisita um objeto, é provável que ele tenha que esperar algum tempo antes de começar a recebê-lo. Esse protocolo possui um funcionamento muito simples. A cada t unidades de tempo, um novo fluxo compartilhado é criado pelo servidor, desde que pelo menos um cliente tenha requisitado o objeto em questão nas últimas t unidades de tempo. Chamamos t de "janela de *batching*", que é o tempo máximo que um cliente espera antes de começar a receber o objeto. Cada novo fluxo criado é compartilhado por todos os clientes que requisitaram o objeto nas últimas t unidades de tempo, daí a economia em banda.

Skyscraper Broadcast [18] estende o *Batching* dividindo a entrega de um mesmo objeto em K canais. O objeto é dividido em K partes e a janela de *batching* de um canal possui o mesmo tamanho da parte sendo exibida no canal, isto é, um novo fluxo para aquele canal é criado toda vez que o fluxo anterior terminar (desde que haja clientes requisitando-o). A vantagem dessa abordagem é que o objeto pode ser dividido em partes de tamanhos diferentes, de forma que as partes iniciais sejam pequenas. Isso faz com que se possa prover serviço quase imediato, já que as janelas de *batching* do princípio da mídia são menores. No entanto, pode ser necessário que um cliente escute mais de um canal simultaneamente e isso exige mais banda de rede do cliente, bem como espaço em *buffer* local.

Patching [28] é um protocolo que provê serviço imediato, às custas de exigir que o cliente escute dois fluxos simultaneamente e tenha espaço em *buffer*. Existem variações do protocolo para os casos em que o espaço em *buffer* é limitado, mas o funcionamento básico é descrito a seguir. O protocolo opera com dois tipos distintos de fluxos: fluxos *unicast* (denominados *patches*) e fluxos compartilhados. Quando um cliente requisita um objeto, ele imediatamente começa a receber um fluxo *patch* correspondente ao início do objeto e, ao mesmo tempo, escuta o fluxo compartilhado que tenha começado mais recentemente, desde que não tenha começado há mais que m minutos, onde m é uma janela de tempo cujo valor ótimo é uma função da demanda pelo objeto em questão. Caso não exista tal fluxo compartilhado, em vez de receber dois fluxos, um novo fluxo compartilhado será criado e o cliente escutará somente esse fluxo. No caso em que o cliente recebe dois fluxos, o cliente escuta e exibe o fluxo *unicast*, enquanto armazena em *buffer* os dados do fluxo compartilhado. Quando a mídia atinge o momento de exibição correspondente ao início dos dados em *buffer*, o fluxo *patch* é encerrado e o cliente passa a exibir os dados do *buffer*, enquanto continua armazenando os dados recebidos pelo fluxo compartilhado.

Bandwidth Skimming(BS) [19] é uma extensão do protocolo *Patching* na qual o correspondente ao fluxo *patch* é um fluxo compartilhado, ao invés de *unicast*. Cada requisição inicia, portanto, um novo fluxo compartilhado, f. O cliente escuta e exibe o fluxo f ao mesmo tempo em que escuta um ou mais dos outros fluxos compartilhados já disponíveis, armazenando esses dados em *buffer*. Quando o fluxo f chega ao ponto em que começam os dados em *buffer*, o cliente deixa de escutar o fluxo f e passa a exibir os dados em *buffer*, da mesma forma que ocorre no protocolo *Patching*. O fato de todos os fluxos serem compartilháveis permite que a

união de fluxos ocorra formando uma estrutura hierárquica. Conforme vimos na seção 2.2.2, Eager et al [20] mostram que o BS consegue atingir um desempenho muito próximo do limite inferior teórico e por isso assumimos neste trabalho a utilização de tal protocolo, aproximando seu custo pela própria função de cálculo do limite inferior.

3.3 Localização e roteamento para fluxos compartilhados

Os protocolos descritos acima focam na redução de banda através do compartilhamento dos fluxos que saem do servidor. Esta seção apresenta trabalhos no sentido de otimizar o roteamento dos fluxos compartilhados de forma a reduzir ainda mais a banda de rede. Note que a questão da localização dos servidores também influencia no custo de roteamento. Esse problema, em geral, é considerado separadamente do de roteamento, uma vez que determinar localização e roteamento ótimos simultaneamente é um problema NP-completo. Assim, descrevem-se aqui protocolos de localização de réplicas e de roteamento dos fluxos. Esses protocolos são utilizados na criação de florestas de distribuição, conceito definido no capítulo 2.

3.3.1 Protocolos localização de réplicas

Chamamos de localização de réplicas o processo de determinar em quais nós da rede deverão ficar as réplicas.

Diversos protocolos foram propostos com foco em transmissão *unicast*, para a qual o roteamento pelo caminho mais curto minimiza a banda média de rede. Em Li et al [36] e Krishnan et al [34] determina-se a localização ótima para servidores *proxy* para *web*, localização esta obtida com o uso de um modelo de otimização. Krishnan et al [34] propõem também uma solução gulosa na qual coloca-se uma réplica de cada vez, mantendo-se fixas as réplicas já colocadas. A réplica é colocada no *site* que ocasionar um menor valor na função que estabelece o custo total. Outros protocolos bastante utilizados são o *Hotspot* [46], que coloca réplicas nos *sites* clientes de maior demanda e o *Fanout* [47, 31], que coloca réplicas nos *sites* com maior grau de saída. Em todos esses estudos utiliza-se também a localização aleatória, para efeito de comparação com os protocolos propostos.

Com relação a estudos voltados para o compartilhamento de fluxos, diversos trabalhos [16, 17, 48, 64, 66] exploram a localização de servidores em CDNs que utilizam fluxos compartilhados. Zhao et al [70] estudaram protocolos de roteamento para fluxos compartilhados e foi utilizado o protocolo *Fanout* para determinar a localização do servidor, ou seja, o servidor foi colocado em um dos *sites* com maior grau de saída. Já Almeida et al [2, 4] introduzem outros três protocolos para escolher a localização das réplicas, sendo duas heurísticas, denominadas *Maximum Savings* e *Min-cost TSP*, e o protocolo ótimo, que obtém a solução através de um modelo de otimização. Os dois primeiros protocolos, que tratam apenas o problema de localização das réplicas, funcionam de forma gulosa, como em Krishnan et al [34]. Já o protocolo ótimo obtém ao mesmo tempo a localização e o roteamento que, combinados, minimizam a banda média de rede. Os três protocolos recebem como parâmetro o valor m, o número de réplicas a serem escolhidos.

```
seja Candidatos o conjunto de possiveis replicas
seja m o numero de replicas a serem escolhidas, m ≤ |Candidatos|
Replicas ← ∅
para i ← 1 ate m
        melhorCusto ← ∞
        melhorCandidato ← NULO
        para cada c ∈ Candidatos
                Replicas ← Replicas ∪ {c}

                // a funcao reconstroiTSP liga cada cliente a replica
                // mais proxima e retorna o custo dessa floresta
                custo ← reconstroiTSP( )

                Replicas ← Replicas − {c}
                se custo < melhorCusto
                        melhorCusto ← custo
                        melhorReplica ← c
                fimse
        fimpara
        Candidatos ← Candidatos − {melhorCandidato}
        Replicas ← Replicas ∪ {melhorCandidato}
fimpara
```

Figura 3.1: Protocolo Min-cost TSP para localização de réplicas

Por ter se mostrado mais promissor dentre as heurísticas apresentadas por Almeida et al [2, 4], o protocolo *Min-cost TSP* foi escolhido para este trabalho.

Min-cost TSP A figura 3.1 mostra o pseudo-código para esse protocolo. Para escolher a i-ésima réplica, calcula-se o custo associado a cada nó candidato. Esse custo corresponde à banda média total da floresta dos caminhos mais curtos, usando como réplicas o candidato e as réplicas já alocadas. Essa floresta é criada ligando cada cliente à réplica mais próxima. A banda média total é dada pela soma das bandas médias em cada *link* dessa floresta. A i-ésima réplica corresponderá ao candidato de menor custo. Esse processo se repete até que m réplicas sejam alocadas. A complexidade desse algoritmo é $O(|S| \times |V|)$, onde S é o conjunto de possíveis réplicas e V é o conjunto de nós da rede.

3.3.2 Protocolos de roteamento

Os protocolos de roteamento recebem como entrada o conjunto de réplicas obtido por algum protocolo de localização e determinam a que réplica deve ser conectado cada cliente e através de que caminho se dá essa conexão, ou seja, criam a floresta de distribuição.

Para transmissão *unicast* existe um protocolo ótimo (e simples) para roteamento, o **caminho mais curto (SP)**. Esse protocolo é o *default* para entrega *unicast* na Internet [37]. Cada cliente é conectado pelo caminho mais curto até o servidor mais próximo. A definição de proximidade é que pode variar. Em muitos casos utiliza-se o número de saltos ou o tempo de ida-e-volta entre o servidor e o cliente.

O caminho mais curto também é proposto como padrão da Internet para entrega usando compartilhamento de fluxos, seja dentro de um mesmo sistema autônomo [41], seja entre

sistemas autônomos distintos [21].

Entretanto, quando se usa compartilhamento de fluxos, o protocolo SP pode não ser ótimo. Por esse motivo, outros protocolos de roteamento são propostos. Fei et al [22] propõem modelos de otimização e heurísticas para roteamento assumindo localização fixa e um único fluxo compartilhado, como ocorre no caso de conteúdo ao vivo ou de difusão (*broadcast*). Nesse caso a banda de rede independe do número de clientes recebendo o fluxo em cada *site* cliente.

Para o caso de vídeo sob demanda foram propostos protocolos de roteamento para compartilhamento de fluxos em dois trabalhos [70, 4]. Zhao et al [70] analisam dois protocolos para roteamento com compartilhamento de fluxos, denominados *Greedy Link* e *Greedy Cost*, para o caso de um único servidor. *Greedy Link* constrói a árvore de distribuição de forma gulosa, ligando, a cada passo, o cliente que puder ser inserido ocasionando o menor aumento no número total de *links* da árvore. *Greedy Cost* funciona da mesma forma, mas em vez de tentar minimizar o número de *links* da árvore, o foco é na minimização da banda de rede total da árvore. Esse custo é definido como a banda média de rede necessária para distribuir um objeto usando BS, considerando as demandas dos *sites* clientes. Os protocolos propostos são comparados com o SP, mas apresentaram ganhos limitados, inferiores a 16%. O melhor deles foi o *Greedy Cost*. O *Greedy Link*, para demandas baixas, chegou até mesmo a apresentar perdas significativas em relação ao SP.

Apesar de o *Greedy Link* ser mais simples por não necessitar saber as demandas dos *sites* clientes, ele não se mostrou promissor. Por isso, Almeida et al [4] propõem uma extensão do *Greedy Cost* para o caso em que há mais de um servidor, denominada Menor Custo Incremental (MCI). São propostos, ainda, um protocolo de complexidade computacional inferior ao MCI, denominado Menor Custo Ordenado (MCO) e o protocolo que obtém a solução ótima para localização e roteamento utilizando um modelo de otimização. Por terem sido usados nas nossas análises, detalhamos abaixo o funcionamento do MCI e do MCO.

Mínimo Custo Incremental (MCI) A figura 3.2 mostra o pseudo-código para o protocolo MCI. Inicialmente, o protocolo MCI constrói uma floresta constituída de m árvores. Cada árvore é formada por um único nó, que corresponde a um dos servidores escolhidos pelo algoritmo de localização de réplicas. Para escolher o i-ésimo cliente a ser inserido, o protocolo MCI conecta cada um dos clientes ainda não inseridos a cada um dos nós na floresta através do caminho mais curto. A opção que ocasionar o menor aumento de custo na floresta será a escolhida. A complexidade desse protocolo é $O(|Clientes|^2 \times |V|)$, onde $Clientes$ é o conjunto de *sites* clientes e V é o conjunto de nós da rede. Portanto, para o caso em que todos os nós são clientes, esse algoritmo possui complexidade $O(|V|^3)$.

Mínimo Custo Ordenado (MCO) A figura 3.3 mostra o pseudo-código para o protocolo MCO. Este protocolo funciona de forma semelhante ao MCI, porém possui menor complexidade, já que insere os clientes em uma ordem pré-determinada. Primeiro são inseridos os clientes com maior demanda e, em caso de empate, primeiro o cliente mais próximo de algum

```
seja Replicas o conjunto de servidores escolhido pelo algoritmo de localizacao
seja Clientes o conjunto de sites clientes

//Ao longo do algoritmo, Vertices sempre corresponde a todos os vertices das
//florestas de distribuicao parcialmente criadas.
Vertices ← Replicas

ClientesNaoInseridos ← Clientes
para i ← 1 ate |ClientesNaoInseridos|
        para cand ∈ ClientesNaoInseridos
                melhorCand ← NULO
                melhorVertice ← NULO
                menorCustoInc ← ∞
                para v ∈ Vertices

                        //testaConectar(c,v) retorna o custo incremental
                        //associado a conexao do cliente c ao vertice v
                        //pelo menor caminho.
                        custoInc ← testaConectar( cand, v)

                        se custoInc < menorCustoInc
                                menorCustoInc ← custoInc
                                melhorCand ← cand
                                melhorVerticeleftarrowv
                        fimse
                fimpara
        fimpara
        ClientesNaoInseridos ← ClientesNaoInseridos − {melhorCand}
        NovosVertices ← conectar( melhorCand, melhorVertice)
        Vertices ← Vertices ∪ NovosVertices
fimpara
```

Figura 3.2: Mínimo Custo Incremental (MCI)

servidor. A inserção, como no caso do Mínimo Custo Incremental, é feita através de uma conexão pelo caminho mais curto com qualquer nó de qualquer das árvores já parcialmente criadas, nó para o qual o aumento no custo da floresta seja o menor possível. Esse protocolo possui complexidade $O(|Clientes| \times |V|)$, onde $Clientes$ é o conjunto de *sites* clientes e V é o conjunto de nós da rede. Portanto, para o caso em que todos os nós são clientes, esse algoritmo possui complexidade $O(|V|^2)$.

3.4 Coleta e mapeamento de topologias

O mapeamento de topologias reais envolve a escolha do nível em que se realizará o mapeamento, geralmente escolhendo-se entre mapear em nível de roteadores ou em nível de sistemas autônomos.

3.4.1 Topologias em nível de sistemas autônomos

Zhang et al [69] montam uma topologia em nível de sistemas autônomos a partir de várias bases de dados correspondentes a tabelas de roteamento BGP [37] de roteadores de borda, tais como Routeviews [51], RIPE RIS [50], servidores de rota, servidores *looking glass* e bases

```
seja Replicas o conjunto de servidores escolhido pelo algoritmo de localizacao
seja Clientes uma sequencia de sites clientes ordenada de acordo com
                a funcao compara(Clientes_i, Clientes_j)

//Ao longo do algoritmo, Vertices sempre corresponde a todos os vertices das
//florestas de distribuicao parcialmente criadas.
Vertices ← Replicas

para i ← 1 ate |Clientes|
        cliente ← Clientes_i
        melhorVertice ← NULO
        menorCustoInc ← ∞
        para v ∈ Vertices
                custoInc ← testaConectar( cliente, v)
                se custoInc < menorCustoInc
                        menorCustoInc ← custoInc
                        melhorVertice ← v
                fimse
        fimpara
        NovosVertices ← conectar( cliente, melhorVertice)
        Vertices ← Vertices ∪ NovosVertices
fimpara

//retorna o cliente prioritario de acordo com o criterio de comparacao do MCO
funcao compara(cliente1, cliente2)
        se demanda(cliente1) > demanda(cliente2)
                retorna cliente1
        senao se demanda(cliente2) > demanda(cliente1)
                retorna cliente2
        senao se distanciaReplicaMaisProxima(cliente1) ≥ distanciaReplicaMaisProxima(cliente2)
                retorna cliente1
        senao
                retorna cliente2
fimfuncao
```

Figura 3.3: Mínimo Custo Ordenado (MCO)

de registro de rota (IRR). Segundo os autores, é a base de dados de sistemas autônomos mais completa já criada e publicada. Cada *link* entre sistemas autônomos é marcado com a data e hora em que foi observado, dando ao usuário da base a possibilidade de escolher entre uma base mais completa, escolhendo todos os *links*, ou uma base mais recente e/ou precisa, escolhendo apenas *links* observados em dado intervalo de tempo. Além disso, a base é mantida atualizada através de um processo automatizado e publicada na *Web* para o público interessado.

Outros trabalhos [38] procuram mapear seqüências de interfaces IP em seqüências de sistemas autônomos, com o objetivo de criar uma ferramenta de *traceroute* em nível de sistemas autônomos. Em geral isso é feito através da comparação de caminhos nos níveis de roteadores e de sistemas autônomos, coletados a partir dos mesmos pontos de coleta, usando, respectivamente, *traceroute* e tabelas BGP.

3.4.2 Topologias em nível de roteadores

O nosso interesse, no entanto, é no mapeamento da Internet no nível de roteadores, devido à necessidade de maior precisão dos mapas topológicos, conforme detalhado no capítulo 2. Existem projetos que coletam rotas usando *traceroute* entre diversos pontos de coleta espalhados pelo mundo, tais como o projeto do NLANR [43] e o projeto Skitter do CAIDA [11]. A mera coleta de rotas, no entanto, não permite a montagem de topologias precisas em nível de roteadores. Isto porque que uma rota consiste numa seqüência de interfaces em vez de uma seqüência de roteadores, porém um mesmo roteador pode possuir diversas interfaces.

O projeto Skitter procura resolver esse problema através de um procedimento que denominaremos "Teste IP", para identificação de pares de interfaces pertencentes a um mesmo roteador. Entretanto, esse teste não resolve completamente o problema. A metodologia do Rocketfuel [57] estende a solução usada pelo projeto Skitter, através do uso de um tipo de teste adicional, que denominaremos "Teste IPID". Por serem importantes para este trabalho, esses dois tipos de teste estão descritos em detalhes na seção 3.4.2.1.

O projeto Rocketfuel [57, 56] mapeia topologias em nível de roteadores de diversos provedores de acesso (ISP) através da coleta de rotas a partir de diversos servidores de *traceroute* públicos. Os destinos de cada rota são endereços IP de prefixo pertencente ao ISP sendo mapeado. O objetivo é selecionar IPs de forma a cobrir toda a rede do ISP e ao mesmo tempo limitar o número de *traceroutes* executados, para não sobrecarregar os servidores públicos de *traceroute*. As interfaces descobertas passam por um processo de resolução de interfaces sinônimas no qual se procura determinar quais interfaces pertencem a um mesmo roteador, realizando os testes descritos na seção 3.4.2.1. Para determinar quais pares de interface devem ser testados, Rocketfuel se baseia principalmente em informações de DNS. Usam-se informações de DNS também para determinar quais roteadores pertencem ao ISP sendo mapeado.

Teixeira et al [58] mostram que as topologias geradas pelo projeto Rocketfuel são imprecisas. Isso é feito comparando-se uma topologia inferida pelo Rocketfuel com a topologia real do mesmo ISP, que forneceu um mapa de sua topologia aos autores sob um acordo de sigilo. Os autores atribuem a imprecisão e incompletude do mapa inferido pelo Rocketfuel às seguintes causas: falta de múltiplos pontos de coleta; rotas incompletas; mudança no roteamento durante a coleta de uma rota; DNS incorreto; falsos positivos na resolução de interfaces sinônimas; erros ao usar DNS para concluir que duas interfaces são sinônimas; adição de *links* reversos. Um outro problema que pode ocorrer, segundo constatamos, é o fato de não se testarem pares de interfaces que deveriam ser testados, por pertencerem a um mesmo roteador. Isso ocorre porque a metodologia do Rocketfuel não testa todos os pares de interfaces encontrados, mas apenas aqueles que apresentam nome DNS parecidos ou estão a uma mesma distância, em saltos, dos pontos de coleta. Na seção 4.5, discutimos como resolvemos cada um desses problemas.

O projeto Rocketfuel [57, 56] usa dados coletados a partir de servidores de *traceroute*, como os listados em [33]. Essa abordagem também usada por Almeida [2] na coleta rotas entre *sites* norte-americanos e europeus, é também a abordagem utilizada neste trabalho,

conforme veremos no capítulo 4.

3.4.2.1 Resolução de interfaces sinônimas

Esta seção descreve os testes utilizados pelo projeto Rocketfuel [57] para determinar se duas interfaces IP são sinônimas, isto é, se pertencem a um mesmo roteador. Esses testes também são utilizados em nossa metodologia, como veremos no capítulo 4.

O projeto Mercator [27] introduziu uma técnica para determinar se duas interfaces são sinônimas, baseada no endereço IP de origem de pacotes ICMP do tipo "Port Not Reachable" retornados por essas interfaces. Sejam A e B dois endereços IP de interfaces que se deseja determinar se são sinônimas. Envia-se para A um pacote UDP, destinado a alguma porta alta (por exemplo, 33434), que provavelmente estará fechada. Estando a porta fechada, espera-se receber de A um pacote ICMP do tipo "Port not reachable", (contendo o pacote original UDP encapsulado). Chamemos esse pacote ICMP de A_{ICMP}. Repete-se o processo para a interface B, obtendo-se um pacote B_{ICMP}. Caso A e B estejam realmente em um mesmo roteador, é possível que ambas as respostas A_{ICMP} e B_{ICMP} tenham saído de uma mesma interface, em geral a interface pela qual o roteador enviaria um pacote destinado à máquina que está realizando o teste. Essa interface de saída pode ser A, B, ou alguma outra interface do roteador em questão. Caso o campo SRC (origem) do pacote A_{ICMP} seja igual a B, ou o campo SRC do pacote B_{ICMP} seja igual a A, é fato que A e B pertecem a um mesmo roteador. Caso ambos sejam diferentes de A e de B, mas sejam iguais entre si (por exemplo, ambos iguais a C), isso indica que não só A e B são sinônimas entre si, mas que existe uma terceira interface C que também é sinônima de A e B. Além disso, sempre que A_{ICMP} tiver o campo SRC diferente de A (sendo igual a D), poderemos dizer que A é sinônima de D. O mesmo vale para B. Chamaremos o teste descrito acima de **Teste IP**.

Sinônimos descobertos por essa técnica são garantidamente sinônimos (não há falsos positivos). No entanto não existe a garantia de que, caso A, A_{ICMP}, B e B_{ICMP} não se enquadrem em nenhuma das situações descritas acima, as interfaces não sejam sinônimas. Em outras palavras, é possível que A e B sejam sinônimos e esse fato não seja descoberto por essa técnica.

O projeto Rocketfuel [57, 56] aprimorou a técnica, utilizando, além do Teste IP, um outro teste, baseado no valor do campo IPID (um identificador de pacote, geralmente gerado pelo sistema operacional do roteador usando um contador que cresce seqüencialmente) desses pacotes. Em termos gerais, a técnica verifica se, nos pacotes ICMP retornados pelas duas interfaces, os campos IPID possuem valores próximos entre si, dentro de certos limites. Em caso positivo supõe-se que os pacotes podem ter sido gerados por um mesmo contador em um mesmo computador. Chamaremos esse tipo de teste de **Teste IPID**.

A técnica do Rocketfuel, que combina os dois tipos de teste, é hoje a mais precisa com o fim de determinar se duas interfaces são sinônimas. Utilizamos essa técnica, portanto, para cada par de interfaces descobertas pela coleta de rotas, estendendo-a para lidar com um número bem maior de interfaces, conforme se descreve no capítulo 4.

Capítulo 4

Coleta

Este capítulo descreve a coleta dos dados necessários ao mapeamento de topologias reais da Internet. Esses "mapas" foram utilizados, neste trabalho, com dois fins. Primeiro, para caracterizar a Internet sob diversos aspectos relevantes ao roteamento de mídia contínua, tais como o aspecto dos caminhos existentes e sua diversidade entre vários pontos da Internet, o comprimento dos caminhos e a assimetria entre caminhos de ida e volta entre dois pontos. Segundo, para se avaliar os ganhos da aplicação de protocolos de roteamento de mídia contínua otimizados para compartilhamento de fluxos. Tais protocolos tiram proveito dessa diversidade e de técnicas como compartilhamento de fluxos para obter economias na banda de rede necessária à entrega da mídia contínua. Esse mapeamento também pode ser utilizado por outros trabalhos que se interessem pela realização de quaisquer simulações que tenham como entrada uma topologia real.

Pela própria dificuldade de se mapear topologias ao nível de roteadores, existe uma carência grande de tais mapas. Essa carência, por si só, já justifica quaisquer esforços de mapeamento.

O processo de mapeamento se divide nas seguintes fases:

- Escolha dos pontos de coleta
- Coleta
- Resolução de interfaces sinônimas
- Padronização e filtragem dos dados

As seções seguintes descrevem cada uma dessas fases.

4.1 Escolha dos pontos de coleta

Os pontos de coleta são servidores públicos de *traceroute*. Esses servidores foram escolhidos a partir de uma listagem disponível em [33]. Alternativas ao uso desses servidores públicos seriam ou a utilização do Planet Lab [44] para coletar rotas usando *traceroute*, ou usar os dados

do CAIDA [11]. O Planet Lab apresenta restrições ao uso, necessidade de autenticação, compartilhamento dos nós com outros grupos de pesquisa e instabilidade de disponibilidade dos nós. Os dados do CAIDA estão em formatos que precisariam de programas para interpretá-los e extrair o tipo de informação que desejávamos, assim achamos mais simples nós mesmos coletarmos as rotas a partir dos servidores públicos.

A escolha dos servidores foi guiada basicamente por dois critérios. O primeiro deles é o grau de penetração da Internet em cada localização geográfica, com base na intuição de que, geralmente, em países mais desenvolvidos o grau de penetração é mais alto. Com base nesse critério, procurou-se escolher um grande número de servidores nos EUA, na Europa e no Japão.

O segundo critério diz respeito ao espalhamento geográfico desses servidores. Assim, em locais de grande extensão geográfica, como EUA e Europa, procurou-se espalhar da melhor forma possível os servidores ao longo da área, tentando não deixar grandes regiões sem cobertura. Esse critério foi utilizado visando cobrir grande parte Internet.

Esses dois critérios foram adotados dentro das limitações impostas pela disponibilidade de servidores *traceroute* públicos em cada região do mundo, conforme a listagem de servidores de que dispúnhamos.

4.1.1 Problemas com servidores

Idealmente, cada um dos 62 servidores escolhidos teria executado o *traceroute* até cada outro servidor uma vez a cada 4 horas durante os 4 meses de coleta e, além disso, a rota não teria problemas como laços, roteadores defeituosos ou que não respondessem.

No entanto, ao longo dos 4 meses, alguns problemas surgiram. Servidores passaram a bloquear nossas requisições, ou não atendiam a uma parte das requisições. Alguns ficaram bom tempo fora do ar. Outros mudaram a URL da interface *web* e demorou um tempo até que percebêssemos a mudança, tempo durante o qual não foram coletados dados a partir de tais servidores.

Com relação aos servidores que passaram a bloquear as requisições vindas do IP correspondente ao nosso sistema de coletas, não houve solução senão removê-los da nossa lista de pontos de coleta. Quanto a servidores que mudaram a URL da interface *web*, ajustamos o sistema de coletas assim que essa mudança foi percebida. Já com os servidores que não atendiam a todas as nossas requisições, ou que geravam rotas incompletas ou apresentavam saídas defeituosas, não houve o que fazer, a não ser filtrar os dados ao final do período de 4 meses.

Foram mantidos apenas os 53 servidores que coletaram rotas no mínimo 300 vezes até cada outro servidor.

Os dados geográficos dos servidores escolhidos são mostrados nas tabelas 4.1 e 4.2 e na figura 4.1. Inicialmente foram escolhidos 62 servidores, mas 9 deles, marcados com um "X" na última coluna das tabelas e desenhados como círculos vermelhos na figura, foram removidos, por terem gerado poucos dados.

Continente (total de servidores após filtragem)	**País (total de servidores após filtragem)**	**Cidade e/ou Estado**	**Servidor**	**Removidos após filtragem**
América do Norte (13)	Estados Unidos (12)	AZ	www.telcom.arizona.edu	
		CA	www.sdsc.edu	
		CA	www.usc.edu	
		CA	www.net.berkeley.edu	
		CA	www.slac.stanford.edu	
		(CO)	www.undergroundpalace.com	
		MD	noc.net.umd.edu	
		ME	home.acadia.net	
		NJ	www.net.princeton.edu	X
		OR	darkwing.uoregon.edu	
		PA	www.net.cmu.edu	
		WA	www.washington.edu	
		WI	cgi.cs.wisc.edu	
	México (1)	?	www.netsolutions.com.mx	
América do Sul (8)	Argentina (1)	Pinamar	tools.telpin.com.ar	
	Bolívia (2)	La Paz	www.scbbs.net	
		Santa Cruz	trace.megalink.com	
	Brasil (4)	Florianópolis	200.146.123.10	
		Florianópolis	nic-2.matrix.com.br	
		Rio de Janeiro	guanabara.rederio.br	
		São Paulo	registro.br	
	Chile (1)	Valparaiso	www.dcsc.utfsm.cl	
	Equador (0)	?	205.247.193.10	X
Oceania (2)	Austrália (1)	Perth	www.autons.net.au	X
		(Sidney)	www.zip.com.au	
	Nova Zelândia (1)	Auckland	www.kcbbs.gen.nz	

Tabela 4.1: Pontos de coleta na América e Oceania

Continente (total de servidores após filtragem)	**País (total de servidores após filtragem)**	**Cidade e/ou Estado**	**Servidor**	**Removidos após filtragem**
África (2)	África do Sul (1)	(Joanesburgo)	services.truteq.com	
	Togo (1)	Lome	labojfl.esiba.edu	
Ásia (8)	Hong Kong (2)		traceroute.hgc.com.hk	
			traceroute.pacific.net.hk	
	Índia (1)	(Nova Delhi)	202.71.136.244	
	Indonésia (1)	?	speedtest.indo.net.id	
	Japão (3)	?	traceroute.hinet.net	
		?	www.harenet.ad.jp	
		?	www.kawaijibika.gr.jp	
		(Hiroshima)	www.tumori.nu	X
	Taiwan (1)	?	140.111.1.22	
Europa (20)	Alemanha (2)	Duesseldorf	lg.inetbone.net	
		Karlsruhe	www-zorn.ira.uka.de	X
		Karlsruhe	sites.inka.de	
		Unterschleissheim	www.tnib.de	X
	Espanha (1)	Madrid	ipet.as12769.net	
	França (3)	Coulommiers	www.netultra.net	
		Montpellier	lg.hostingfrance.com	
		(Paris)	www.azuria.net	
		Saint Etienne	www.univ-st-etienne.fr	X
	Grécia (1)	Atenas	www.ntua.gr	
	Islândia (1)	Reykjavik	www.rhnet.is	
	Itália (3)	Bologna	www.cnaf.infn.it	
		Roma	www.mclink.it	
		Torino	carmen.cselt.it	
	Polônia (2)	Warszawa	cgi.ipartners.pl	
		Warszawa	www.atcom.net.pl	
	Portugal (1)	Lisboa	glass.cprm.net	
	Reino Unido (3)	Aylesbury	www.nildram.net	
		Glasgow	ppewww.ph.gla.ac.uk	
		Londres	www.hotlinks.co.uk	X
		Londres	www.mailbox.net.uk	
	Rússia (2)	Moscou	lg.transtk.ru	
		Moscou	ulda.inasan.rssi.ru	
	Turquia (0)	Ankara	appsrv.ttnet.net.tr	X
	Ucrânia (1)	Donets	nic.dn.ua	

Tabela 4.2: Pontos de coleta na África, Ásia e Europa

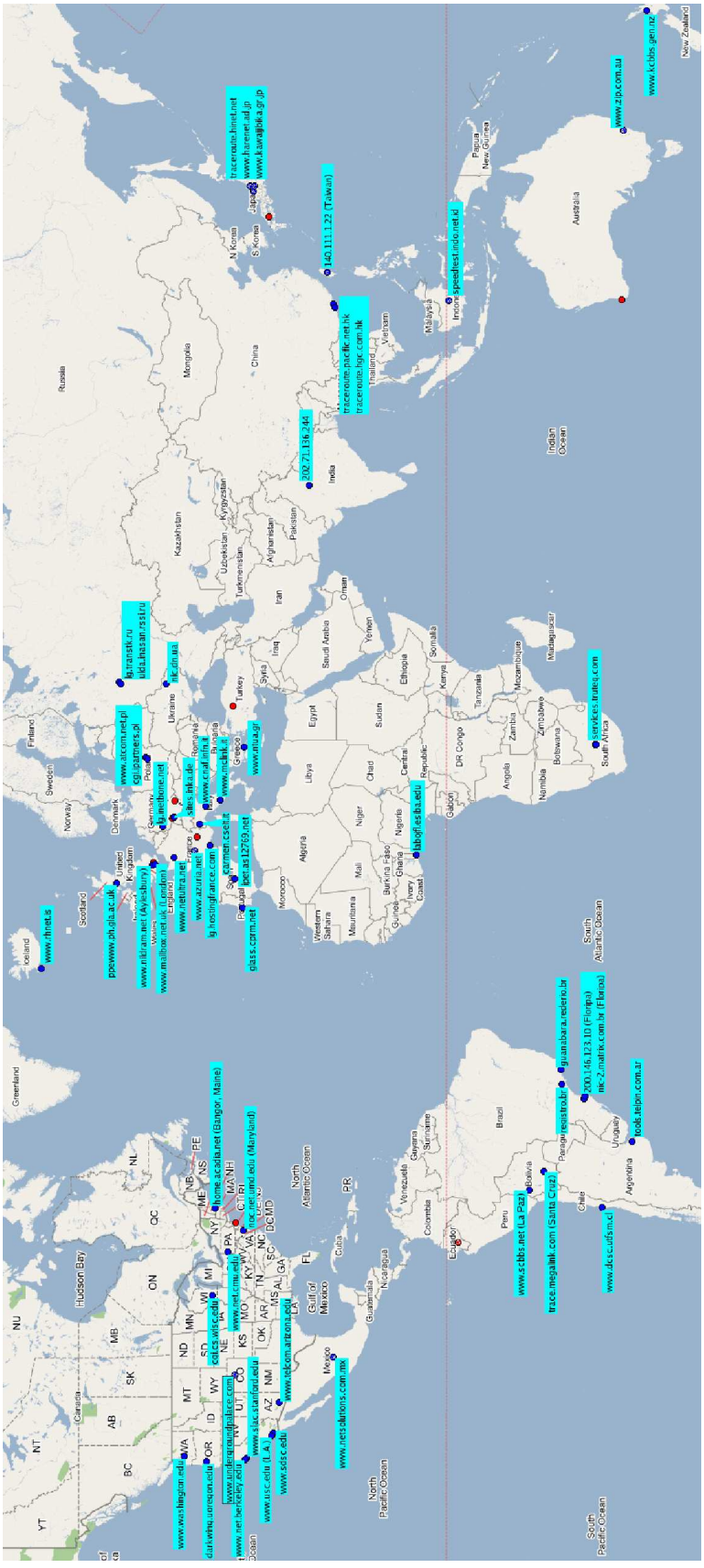


Figura 4.1: Distribuição geográfica dos pontos de coleta de *traceroute*

```
[eric@tornado data]$ more 140.111.1.22_2005.04.21.11.06.08.raw
traceroute from www.net.berkeley.edu (128.32.206.221) to 140.111.1.22 (140.111.1.22)
 1  vlan206.inr-203-eva.Berkeley.EDU (128.32.206.1)  0.884 ms  0.709 ms  0.878 ms
 2  vlan210.inr-202-doecev.Berkeley.EDU (128.32.255.9)  0.410 ms  0.451 ms  1.377 ms
 3  ge-1-3-0.inr-002-reccev.Berkeley.EDU (128.32.0.38)  58.657 ms  0.524 ms  0.452 ms
 4  hpr-oak-hpr--ucb-ge.cenic.net (137.164.27.129)  0.773 ms  5.747 ms  0.671 ms
 5  sac-hpr--oak-hpr-10ge.cenic.net (137.164.25.17)  19.292 ms  6.165 ms  2.482 ms
 6  lax-hpr--sac-hpr-10ge.cenic.net (137.164.25.10)  12.021 ms  11.921 ms  12.155 ms
 7  twaren-local.lsanca.pacificwave.net (207.231.240.133)  11.992 ms  11.716 ms  11.870 ms
 8  S4-GE-CHTI-RI1.HCC-LA.twaren.net (211.79.48.209)  172.373 ms  171.994 ms  171.976 ms
 9  S16-POS-EBT-R1-TPC-HCC.twaren.net (211.79.59.182)  173.506 ms  173.579 ms  173.362 ms
10  1G-GE-R1.TPB-TPC.twaren.net (211.79.59.70)  183.171 ms  183.166 ms  183.170 ms
11  210.200.33.10 (210.200.33.10)  157.170 ms  156.636 ms  156.702 ms
12  140.111.230.249 (140.111.230.249)  157.152 ms  157.368 ms  157.259 ms
13  rs.edu.tw (140.111.1.22)  157.169 ms *  157.134 ms
[eric@tornado data]$
```

Figura 4.2: Exemplo de saída do *traceroute*

4.2 Coleta de rotas

Foi implementado um sistema de coletas para automatizar a coleta de rotas entre cada par de servidores. O sistema de coletas tira proveito do fato de que todos esses servidores públicos são acessíveis através de interface *web*. Essas interfaces têm o funcionamento muito parecido em todos os servidores. Um formulário HTML recebe do usuário o nome ou IP do servidor destino. Assim, usamos o programa *wget* [42] para requisitar essas páginas, passando por GET ou POST (dependendo de como funciona a interface do servidor em questão) o parâmetro "servidor destino". Usamos o *cron* [65] para automatizar a execução do conjunto de comandos *wget* a cada 4 horas, durante 4 meses, no período de maio a agosto de 2005.

A página HTML recebida é então filtrada de forma a deixar apenas a saída do programa *traceroute* executado pelo servidor. Essa saída é então armazenada, juntamente com a data e hora em que foi realizada a coleta. Na maioria dos casos a saída tem um formato como o exemplo da figura 4.2. Note que o nome do arquivo contém a data e a hora da coleta e a primeira linha do arquivo contém os servidores origem e destino. Em outros casos a saída é ligeiramente diferente. Para esses casos criaram-se *scripts* para converter a saída para esse formato, que adotamos como padrão.

Os dados acima contêm informação de atraso de ida-e-volta dos pacotes, bem como hora e data da coleta. A análise dos atrasos foi deixada para trabalhos futuros e assim criou-se um terceiro formato de arquivo em que todas as rotas coletadas entre dado par de servidores são armazenados, uma rota por linha do arquivo, cada linha contendo os endereços IP dos roteadores no caminho entre os servidores. Cada arquivo contém, portanto, todas as **seqüências de interfaces** correspondentes aos *traceroute* executados de dado servidor até outro.

4.3 Resolvendo interfaces sinônimas

Esta seção descreve como as seqüências de interfaces foram transformadas em **seqüências de roteadores**.

Roteadores possuem pelo menos duas interfaces IP. Portanto, as 13335 interfaces que apareceram nas rotas coletadas podem corresponder a um número muito menor de roteadores.

Chamaremos de **interfaces sinônimas** duas ou mais interfaces que pertençam a um mesmo roteador.

É importante saber quais interfaces pertencem a cada roteador, do contrário enxergaríamos uma topologia com muito mais nós que a real e com uma diversidade de caminhos disjuntos de roteadores também superior. A seguir descrevemos o processo usado para determinar as interfaces sinônimas.

4.3.1 Testando todos os pares de interface

O projeto Rocketfuel [57] testava apenas pares de interfaces que davam algum indicativo de provavelmente pertencerem a um mesmo roteador, por exemplo: pares de interfaces com nome parecido no DNS, ou pares de interfaces cujos IPs são parecidos. Isso pode deixar de fora diversos pares que não têm essas características em comum e ainda assim pertencem a um mesmo roteador. A vantagem da nossa metodologia de resolução de interfaces sinônimas sobre a metodologia adotada no projeto Rocketfuel é o fato de testarmos todos os pares de interfaces encontrados nas coletas de *traceroute*, usando para cada par, o "testes IP" e o "teste IPID" descritos na seção 3.4.2.1.

O maior problema com a nossa metodologia é a quantidade de pares a serem testados. Havendo n interfaces, são $(n \times (n-1))/2$ pares a serem testados[1]. No nosso caso, encontramos 13335 interfaces, das quais 8897 respondiam ao teste de resolução de interfaces sinônimas, o que corresponde a quase 40 milhões de pares. Dependendo da velocidade com que essas interfaces respondem (se responderem), um teste de um par de interfaces pode durar até 5 segundos. Mesmo assumindo que cada teste durasse em média apenas 1 segundo, testar 40 milhões de pares seqüencialmente é inviável, pois demoraria mais de 1 ano.

Existe a possibilidade de paralelização desses testes. Sendo 8900 interfaces, mesmo que não queiramos envolver uma mesma interface em mais de 1 teste por um período de 5 segundos, é possível, em tese, testar 4450 pares simultaneamente a cada 5 segundos. Criou-se então um sistema de resolução de interfaces sinônimas, com os seguintes objetivos:

- gerar os pares a serem testados, não testando interfaces que não estejam respondendo;
- paralelizar a realização dos testes;
- priorizar pares de interfaces que tiverem maior chance de serem sinônimas entre si;
- testar pelo menos 3 vezes, temporalmente espaçadas, pares cujo teste IPID (seção 3.4.2.1) acusasse sinonimidade, mas o teste IP (seção 3.4.2.1) não conseguisse determinar isso.

[1] Uma otimização poderia ser feita para diminuir o número inicial de pares, tirando proveito do fato de a realização do Teste IP não requerer que ambas as interfaces de um par passem pelo teste simultaneamente. Lembrando que o Teste IP consiste no envio de um pacote a cada interface do par e a comparação entre os campos SRC dos pacotes ICMP retornados por elas, seria possível enviar somente 1 pacote a cada uma de todas as interfaces encontradas e utilizar os resultados para formar uma base inicial de interfaces sinônimas. Com isso o número total inicial de interfaces, que passariam pelo Teste IPID, seria reduzido, já que a relação de sinonimidade é transitiva (se A é sinônimo de B e B é sinônimo de C, então A é sinônimo de C).

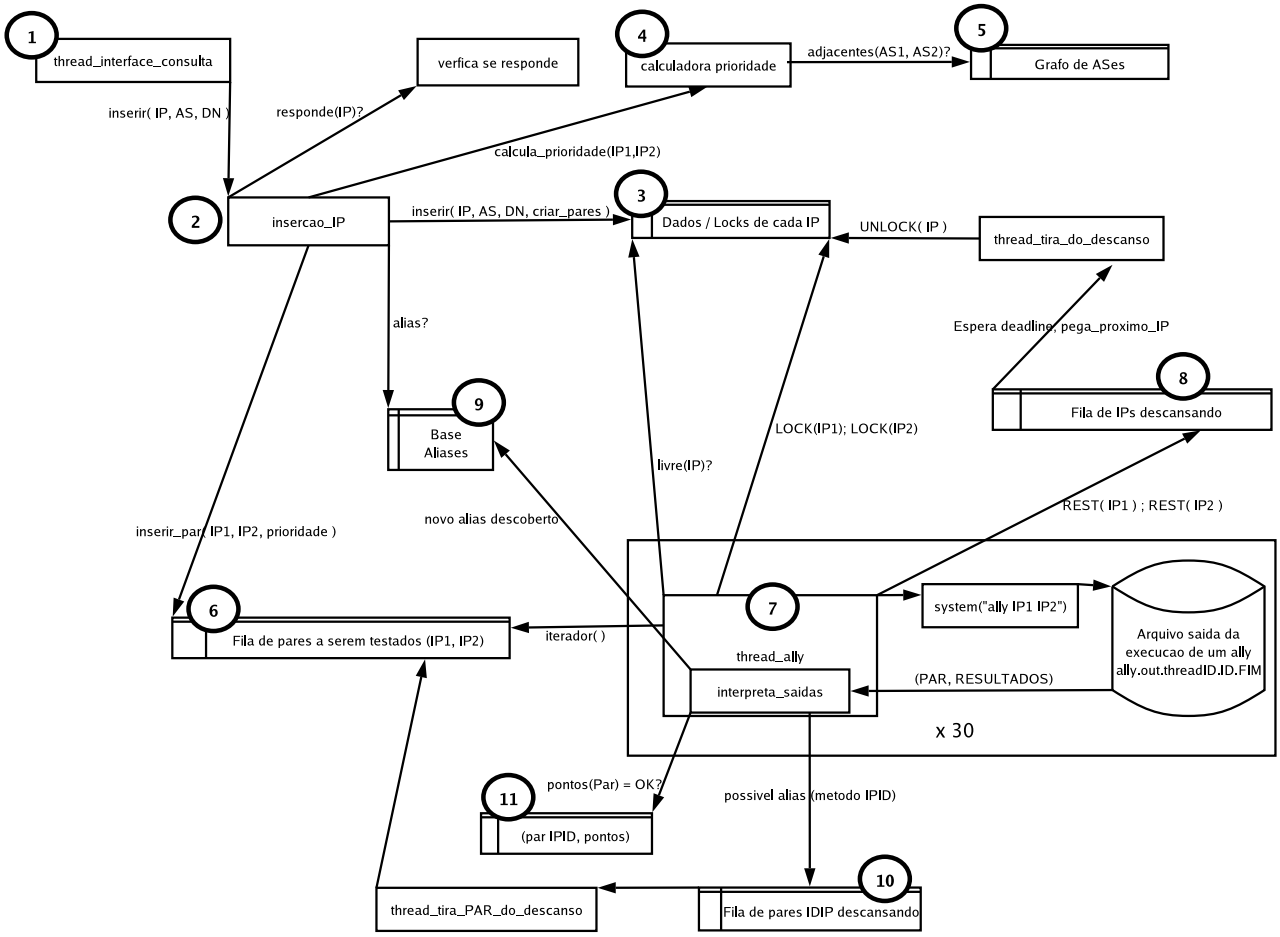


Figura 4.3: Arquitetura do sistema de resolução de interfaces sinônimas

```
[root@murupi bin]# ./ally 150.164.3.39 201.78.62.81
0: from 150.164.3.39: id 9400, ttl 60/64
1: from 201.78.62.81: id 40987, ttl 46/51
ip1=150.164.3.39 ip2=201.78.62.81 !ipid !return_ttl !out_ttl !same_ip !b_returns_a !a_returns_b !cisco
[root@murupi bin]#
[root@murupi bin]#
[root@murupi bin]# time ./ally 150.164.3.39 201.78.62.81
0: from 150.164.3.39: id 9402, ttl 60/64
1: from 201.78.62.81: id 41019, ttl 46/51
ip1=150.164.3.39 ip2=201.78.62.81 !ipid !return_ttl !out_ttl !same_ip !b_returns_a !a_returns_b !cisco

real    0m0.062s
user    0m0.002s
sys     0m0.003s
```

Figura 4.4: Exemplo de execução da ferramenta *ally*

Isso porque o teste IPID pode gerar falsos positivos decorrentes de coincidências no valor do campo IPID.

4.3.1.1 Arquitetura e funcionamento do sistema de resolução de interfaces sinônimas

Nosso sistema de resolução de interfaces sinônimas foi construído utilizando como base a ferramenta *ally* do projeto Rocketfuel [57]. O programa *ally* recebe como parâmetros duas interfaces IP, realiza o Teste IP e, caso o resultado não seja positivo, realiza também o Teste IPID.

A figura 4.4 mostra um exemplo de execução do *ally*, para duas interfaces que não são sinônimas. A linha rotulada "0:" mostra os dados do pacote ICMP recebido do roteador que contém a primeira interface, 150.164.3.39. Neste caso o pacote ICMP veio dessa mesma interface, segundo mostra o campo "from" dessa linha, mas ele poderia ter vindo de outra, revelando assim um sinônimo para 150.164.3.39. Essa linha mostra também o valor do campo IPID desse pacote, 9400, bem como o TTL do pacote de requisição quando esse pacote chegou à interface e o TTL do pacote ICMP recebido. A linha rotulada "1:" mostra o mesmo tipo de informação para o pacote recebido da segunda interface. Poderia haver linhas "2:" e "3:" correspondente a segundos pacotes enviados às interfaces. O envio do segundo pacote ocorre, por exemplo, quando os primeiros pacotes indicam que as interfaces podem ser sinônimas pelo teste IPID, o que não é o caso, já que os campos IPID dos primeiros pacotes são bem diferentes entre si.

Na mesma figura, na segunda execução, que foi cronometrada com o comando *time*, note que o tempo de processamento é de 5 milisegundos, enquanto o tempo de resposta é de 62 milisegundos. Como o teste da figura foi realizado em uma máquina que não estava sobrecarregada, isso indica que o processo *ally* ficou muito tempo parado, simplesmente esperando os pacotes ICMP de resposta retornarem. Daí a possibilidade de paralelização dos testes, a qual nosso sistema explora.

Nosso sistema é responsável por criar e gerenciar os pares de interface, determinando quando cada par deve ser submetido ao teste do *ally*, interpretando a saída do *ally* e assim criando uma base de sinônimos. A seguir detalhamos o funcionamento do sistema.

A figura 4.3 mostra a arquitetura do nosso sistema. Os rótulos numéricos dessa figura (dentro de círculos) serão utilizados a seguir, entre parênteses, para facilitar a descrição passo-a-passo do funcionamento do sistema.

O sistema recebe, através de um arquivo fornecido pela interface com o usuário **(1)**, uma lista de interfaces (endereços IPv4), cada uma delas opcionalmente associada a um nome DNS e a um identificador de sistema autônomo.

O sistema primeiro testa **(2)** se a interface está respondendo ao tipo de teste realizado pelo *ally*. Caso dada interface não esteja respondendo, ela é armazenada a parte e o sistema realiza um novo teste com ela a cada 6 horas para verificar se ela passou a responder e já pode ser inserida.

Caso esteja respondendo, a interface será então inserida para testes. Criam-se os objetos correspondentes aos pares dessa interface com todas as outras interfaces já inseridas **(3)**, atribuindo-se uma prioridade de 1 a 5 a cada novo par criado. Essa prioridade é calculada **(4)**

com base na semelhança entre os endereços IP das interfaces, entre os nomes DNS e entre os identificadores de sistemas autônomos. Considera-se que identificadores de sistemas autônomos são semelhantes se eles são iguais ou se correspondem a sistemas autônomos adjacentes de acordo com o grafo de sistemas autônomos **(5)** obtido em [13, 10]. Os pares são inseridos na fila de pares **(6)** para testes, em uma posição que depende da prioridade do par, pares mais prioritários ficando no princípio da fila.

Existem n fluxos de execução paralelos **(7)** responsáveis por testar os pares da fila (no nosso caso usamos $n = 30$, mas esse valor pode ser configurado de acordo com a capacidade de processamento da máquina onde o sistema estiver rodando). Cada fluxo de execução varre a fila a partir do princípio (maior prioridade), buscando por um par tal que nenhuma das interfaces esteja travada, isto é, envolvida em um teste com alguma outra interface. Ao encontrar um par assim, esse par é retirado da fila e as interfaces que o compõem são travadas **(3)**.

O fluxo de execução realiza a chamada *fork*, para executar o *ally*, que realiza o teste com as duas interfaces e armazena sua saída em um arquivo.

Após terminado o *ally*, o par é descartado e as interfaces correspondentes ficam em uma fila de "descanso" **(8)** por 5 segundos antes de serem destravadas **(6)**. Isso impede que as interfaces sejam testadas muito freqüentemente. Um dos problemas com testes demasiadamente freqüentes é que existem roteadores que limitam a quantidade de pacotes ICMP gerados por unidade de tempo. Portanto, testar várias vezes seguidas uma interface de tal roteador pode levar a vários desses testes serem inconclusivos.

Também após a execução do *ally*, o mesmo fluxo **(7)** que deu origem ao processo *ally* é encarregado de ler o arquivo gerado (a saída do *ally*) e interpretá-lo, para determinar se foram descobertos novos sinônimos e assim atualizar a base de interfaces sinônimas **(9)**.

Caso o par tenha passado no teste IP, eles são garantidamente sinônimos, assim essa informação é armazenada na base de sinônimos **(9)** e o objeto correspondente ao par é descartado. Caso contrário, se apenas o teste IPID tiver sido positivo, existe a chance, mas não a garantia, de as interfaces serem sinônimas. Neste caso, um novo par é criado com essas interfaces. Esse novo par descansa por 1 hora **(10)** para dar tempo de os contadores dos dois roteadores se dessincronizarem caso o resultado positivo tenha sido apenas uma coincidência. Após esse tempo, o novo par volta para a fila de testes **(6)**, a fim de ser testado novamente. Ao ter passado por 3 vezes pelo teste IPID (essa informação é mantida em **(11)**), as interfaces do par são finalmente consideradas sinônimas e armazenadas na base de sinônimos **(9)**. Se em alguma dessas 3 vezes o teste IPID falhar, considera-se que havia sido apenas uma coincidência e o par é removido do sistema sem que as interfaces que o compõem sejam consideradas sinônimas.

4.3.2 Resultados da resolução de interfaces sinônimas

O sistema de resolução de interfaces sinônimas, rodando paralelamente em 10 máquinas, foi alimentado com as 13335 interfaces encontradas, juntamente com os nomes DNS (na maioria dos casos obtidos a partir das próprias saídas do *traceroute*) e identificadores de sistemas

autônomos (obtidos a partir de informações do *whois*). Com isso todos os pares puderam ser testados em um período de cerca de 10 dias.

Cerca de 1/3 das interfaces (4438) não responderam aos testes de resolução de interfaces sinônimas e acabaram sendo consideradas roteadores de 1 interface [2]. Cada uma dessas interfaces foi marcada com um 'N' na nossa base de rotas. Nosso objetivo original com a marcação dessas interfaces era que, no momento de se montarem as topologias para as simulações de roteamento, pudéssemos não levar em conta essas interfaces. Isso nos daria uma rede precisa (já que todos os nós teriam passado pelos testes de resolução de sinônimos), ainda que não tão completa. No entanto, durante a fase de montagem de topologias, constatou-se que, ao se removerem as interfaces 'N', a topologia obtida era desconectada (na verdade, completamente fragmentada). Tivemos então que optar por não remover essas interfaces, obtendo assim uma topologia potencialmente menos precisa, porém mais completa. A marca 'N' foi mantida para o caso de ser útil a outros grupos de pesquisa que queiram aproveitar esses dados.

Foram testados todos os pares possíveis entre as restantes 8897 interfaces (quase 40 milhões de pares). Durante o processo de resolução de interfaces sinônimas, foram descobertas outras 790 interfaces (fato que pode ocorrer conforme a última situação descrita no segundo parágrafo da seção 3.4.2.1). O gráfico da figura 4.5 mostra o resultado final da resolução de interfaces, para as 8897 interfaces originais que responderam aos testes mais as 790 novas descobertas. Note que a grande maioria (92%) dos roteadores possui de 1 a 7 interfaces e que existe um roteador com 25 interfaces.

Notou-se ainda que, em cerca de 77.000 dentre os últimos pares de interfaces testados pelo sistema de resolução de interfaces sinônimas, apenas 3% dos pares que foram considerados sinônimos pelo menos uma vez pelo Teste IPID conseguiram esse resultado em 3 realizações desse teste. Os outros 97% dos resultados "positivos" eram, na realidade, falsos-positivos. Essa alta taxa de falsos-positivos mostra que foi acertada a escolha de realizar 3 vezes o Teste IPID antes de considerar que um par de interfaces são sinônimas. Devido a uma perda massiva dos *logs* gerados pelo sistema de resolução de interfaces sinônimas, somente foi possível analisar os dados dessa pequena amostra, relativa aos últimos 77.000 pares de interfaces testados (o total de pares testados chega a quase 40 milhões). Além disso, por esses pares serem menos prioritários, eles acabaram dando origem a poucos pares (pouco mais de 1000) de interfaces sinônimas. Assim, infelizmente não dispomos de estatísticas a respeito de quantos pares passaram no Teste IP, e quantos passaram no Teste IPID, e os dados a respeito de falsos-positivos referem-se apenas a essa amostra.

Para cada roteador foi escolhida arbitrariamente uma de suas interfaces para ser o nome padrão desse roteador. Dessa forma foi possível traduzir os nossos dados de seqüências de interfaces para seqüências de roteadores.

[2] Um roteador de 1 interface é, na verdade, um roteador do qual *conhecemos o endereço IP* de apenas uma das interfaces. Estritamente falando não existem roteadores com uma interface, já que um roteador possui, por definição, no mínimo duas interfaces, dessa forma servindo à sua função de interconectar duas ou mais redes físicas

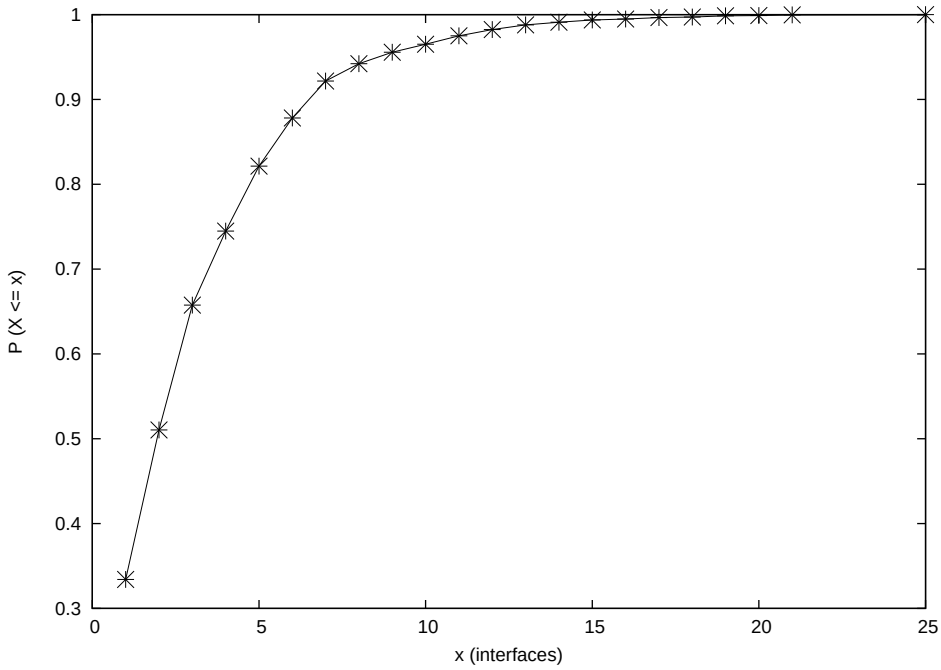


Figura 4.5: Número X de interfaces em um roteador

4.4 Padronização e Filtragem

4.4.1 Aglomeração e padronização das pontas

As seqüências de roteadores passaram por um processo de padronização e aglomeração das "pontas". Com padronização queremos dizer o seguinte. Algumas vezes, os endereços IP de alguns dos servidores de *traceroute* mudaram. Isso ocorreu freqüentemente em 2 dos servidores, o que revela um aparente uso de *DNS round-robin*. O processo de padronização escolheu um desses IPs para ser o padrão e converteu as demais seqüências de rotedores para esse padrão.

Já com relação à aglomeração, foi feito o seguinte: nas pontas (início e fim) de cada seqüência de roteadores, foram aglomerados em um único nó todos os roteadores que estavam no domínio (DNS) do servidor daquela ponta. Isso teve o objetivo de simplificar as topologias, já que o alvo de nossas análises é a região mais central da Internet e portanto a topologia que se encontra dentro dos domínios dos servidores *traceroute* é menos relevante. Além disso, as nossas análises, seja a caracterização da diversidade de caminhos (capítulo 5), seja a avaliação de protocolos de roteamento usando fluxos compartilhados (capítulo 6), assumem que são abundantes a diversidade de caminhos e a banda de rede disponíveis dentro dos domínios (redes locais) dos 53 servidores focados e, por isso, o custo de rede nesses domínios é desprezível em relação ao custo de rede fora dos domínios.

4.4.2 Filtragem

Um problema fácil de ser detectado que aparece em rotas coletadas por *traceroute* é a existência de laços. Um mesmo roteador aparece, por exemplo, a uma distância de 4 saltos e mais tarde, na mesma coleta, a uma distância de 7 saltos. Esse tipo de problema pode indicar mudança no roteamento ou ainda *bug* no sistema operacional de algum roteador no caminho. Seqüências de roteadores contendo laços foram removidas.

Outro problema que ocorre em coletas de rotas usando *traceroute* é a existência de roteadores que estão configurados para não responder com os pacotes ICMP do tipo "time exceeded" com que se espera que respondam. Esses roteadores aparecem como um asterisco na saída do *traceroute*. Além disso, alguns sistemas operacionais contêm *bugs* que fazem com que o campo TTL do pacote ICMP que eles enviam como resposta seja insuficiente para que o pacote chegue de volta até o servidor *traceroute*. Neste caso, em geral ocorre um número grande de linhas contendo asteriscos consecutivas na saída do *traceroute*. Esses problemas estão descritos em detalhes em [30]. Por garantia, foram eliminadas todas as seqüências de roteadores que continham asteriscos.

Além disso, eliminaram-se seqüências que apareceram somente uma vez ao longo de todo o período de coletas, pois elas podem indicar que houve uma mudança no roteamento durante a coleta da rota. O princípio do mapeamento por *traceroute* é que todos os pacotes IP enviados durante um determinado período de tempo seguirão uma mesma rota até o servidor de destino. Como esses pacotes têm TTL incrementais, começando de 1, os pacotes não chegam até o destino mas sim retornam, encapsulados em pacotes ICMP, a partir dos vários roteadores existentes no caminho. Assim, assume-se que existe um caminho passando por todos esses roteadores. No entanto, caso haja alguma mudança no roteamento entre a fonte e o destino durante a execução do *traceroute*, pode aparecer um caminho que na verdade é a concatenação de dois ou mais caminhos. Acreditamos que a remoção de caminhos que tenham aparecido somente uma vez durante as coletas, bem como os caminhos contendo laços[3] é capaz de eliminar esse tipo de problema.

4.4.3 Resultados

Como resultado do processo filtragem, constatou-se que a remoção de laços diminuiu em 10% o número total de seqüências de roteadores. Outros 20% das seqüências restantes foram removidos por conterem asteriscos. Por fim, 1,1% das seqüências restantes foram removidos por terem aparecido uma única vez.

A figura 4.6 mostra a distribuição acumulada do número de seqüências de roteadores entre cada par origem-destino, após cada filtragem realizada. Veja que no início do processo temos pelo menos 300 coletas de rota entre quase todos (>97,5%) os pares origem-destino. Ao final das filtragens, por volta de 15% dos pares origem-destino não possuem nenhuma seqüência de

[3]Caminhos contendo laços são outro sintoma comum da mudança no roteamento durante a coleta (*route flapping*).

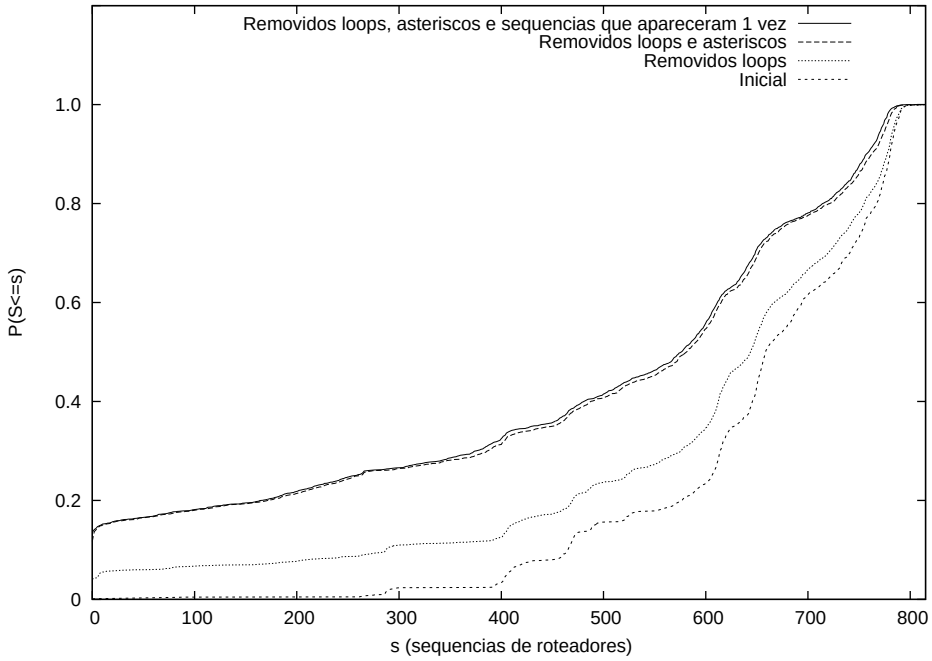


Figura 4.6: Número S de seqüências de roteadores após cada passo da filtragem

roteadores entre eles. No entanto verificou-se que, em todos esses casos, pelo menos 1 caminho no sentido reverso existe na base, portanto esses servidores foram mantidos.

4.5 Resolvendo problemas apontados

Teixeira et al [58] mostram diversas fontes de imprecisão no mapeamento de topologias utilizado pelo projeto Rocketfuel [56, 57]. Nesta seção, mostramos como cada uma desses problemas foi tratado no nosso processo de mapeamento.

O primeiro problema apontado é a falta de pontos de observação. Isso leva as topologias de ISPs mapeadas pelo Rocketfuel a serem incompletas. No nosso caso, isso não chega a ser um problema. Não temos a intenção de mapear toda a Internet. Nosso objetivo é que o subconjunto mapeado seja preciso[4].

O segundo problema também diz respeito à incompletude da topologia, causada pela existência de rotas incompletas. No nosso caso essas rotas foram eliminadas. E, como foram realizadas diversas coletas de rota rota entre cada par de *sites*, as rotas incompletas, na maioria dos casos, não fizeram falta[5].

A terceira fonte de imprecisões é a mudança no roteamento durante a coleta de um de uma rota. Isso levaria à inserção de *links* que na verdade não existem. Acreditamos que

[4] Já a **representatividade** desse conjunto é algo bem mais difícil de se medir, conforme notamos na seção 5.1.2.1

[5] Exceto nos 15% dos pares de pontos de coleta, mencionados na seção 4.4.3, que ficaram sem caminhos em um dos dois sentidos (ida ou volta). Mas caminhos alternativos foram encontrados para todos esses casos, ao se montar o mapa da topologia global usando todas as rotas, conforme veremos no capítulo 6.

removendo os laços e os caminhos que apareceram uma única vez, esse problema tenha sido tratado.

A quarta fonte de imprecisão diz respeito à determinação de quais interfaces pertencem ao ISP sendo mapeado. No nosso caso isso não é um problema, pois não estamos interessados em mapear um ISP específico.

Com relação à resolução de interfaces sinônimas, os autores apontam que podem haver falsos positivos. No entanto falsos positivos ocorreriam apenas no caso do teste IPID, quando, por coincidência, os pacotes vindos de duas interfaces possuissem valores próximos entre si no campo IPID. Para tratar isso, se um par de interfaces passa apenas no teste IPID, ele é submetido a esse teste mais 2 vezes, antes de ser considerado um par de interfaces sinônimas.

Os autores de [58] interpretam de maneira errônea a metodologia do Rocketfuel, ao dizerem que o uso de DNS para resolver interfaces sinônimas também pode introduzir falsos positivos. O Rocketfuel apenas usa informações de DNS para determinar quais pares de interfaces devem passar pelos testes IP e IPID e nunca para agrupar duas interfaces com base simplesmente no nome. Portanto, ao contrário do afirmado em [58], o uso de DNS não introduz falsos positivos na fase de resolução de interfaces sinônimas.

No entanto, consideramos que o mero uso de DNS para determinar quais pares de interfaces devem ser testados leva à falta de teste para alguns pares de interfaces que pertencem ao mesmo roteador mas não possuem nomes parecidos. Por esse motivo, nossa metodologia testa todos os pares de interfaces. De fato, verificamos que 7% das interfaces para as quais encontramos pelo menos um sinônimo não possuem nem endereço IP e nem nome DNS semelhantes aos de nenhum de seus sinônimos[6].

O último problema apontado por [58] é a adição de *links* reversos, assumindo que todo *link* AB é bidirecional. Essa adição de links reversos não ocorre na nossa metodologia.

4.6 Conclusões

O processo de coleta deu origem a uma base de dados de seqüências de roteadores correspondentes a caminhos existentes e efetivamente utilizados no roteamento, entre 53 servidores espalhados por todo o mundo. Esse foi o resultado após a coleta de rotas entre os 62 servidores, a resolução de interfaces sinônimas, as remoções dos servidores pouco ativos e a filtragem dos dados.

Essa base de dados pode ser utilizada para diversos fins. Um deles é tentar caracterizar a Internet em termos de diversas métricas interessantes do ponto de vista de roteamento de mídia contínua. Outra finalidade dessa base é a montagem de mapas topológicos, grafos correspondentes a um subconjunto real da topologia da Internet. Tais grafos podem ser utilizados como entrada em quaisquer simulações de protocolos de rede. De fato, essas foram as duas finalidades que demos a essa base, conforme veremos nos capítulos 5 e 6.

[6] Considerando que dois endereços IP são semelhantes se estão na mesma sub-rede /16 (por exemplo, 192.168.10.4 e 192.168.0.1 são semelhantes), e que dois nomes DNS são semelhantes se diferem apenas no primeiro campo (roteador1.ufmg.br e roteador2.ufmg.br são semelhantes).

Capítulo 5

Caracterização

Este capítulo analisa as seqüências de roteadores obtidas conforme descreve o capítulo 4, de forma a caracterizar a Internet com relação a parâmetros importantes para o roteamento de mídia contínua. Essas seqüências de roteadores correspondem a caminhos efetivamente tomados por pacotes IP na Internet atual entre os pontos de coleta, portanto o uso dessas seqüências torna a caracterização mais realista e precisa. Daremos preferência, neste capítulo, aos termos "caminho" e "*site*" em vez dos respectivos sinônimos "seqüência de roteadores" e "ponto de coleta".

Dois aspectos importantes da rede e do roteamento feito sobre ela são a diversidade de caminhos e a assimetria. Esses aspectos têm importância para diversas aplicações, em especial para prover um roteamento de mídia contínua mais eficiente e/ou com melhor qualidade, com a aplicação de técnicas como codificação em múltiplas camadas [7], roteamento por múltiplos caminhos [24], compartilhamento de fluxos de rede [4] e redes *overlay* adaptáveis (*resilient overlay networks, RON*) [5]. Por esse motivo esses dois aspectos são o foco desta caracterização.

A diversidade de caminhos é dada em termos de dois parâmetros, o número de caminhos diferentes existentes entre dois pontos e o grau de diferença entre esses caminhos. Um terceiro parâmetro que pode influenciar na diversidade de caminhos é a distância entre esses dois *sites*. Neste trabalho nos concentramos nesses três parâmetros, número de caminhos, grau de diferença e distância, a serem melhor definidos na seção 5.1.1, para quantificar a diversidade de caminhos existente na Internet.

Já a assimetria diz respeito à diferença entre o caminho tomado por um pacote que vai de A até B e um caminho que vai de B até A. A forma como essa diferença é medida depende do tipo de estudo sendo feito. Pode se referir à diferença no atraso de propagação, ou à diferença entre o número de *links* ou roteadores, ou mesmo à diferença entre a seqüência específica de roteadores tomados no caminho de ida e no caminho de volta. Neste trabalho definimos a assimetria dessa última forma, conforme veremos na seção 5.1.1.

O restante do capítulo se organiza da seguinte forma. A seção 5.1 descreve cada um dos parâmetros usados na caracterização. Descreve também a forma geral como cada um desses parâmetros foi analisado, isto é, quais os conjuntos de *sites* utilizados, bem como o significado

dos gráficos e tabelas criados. A seção 5.2 apresenta gráficos e tabelas e interpreta esses dados no contexto de roteamento de mídia contínua. Por fim, a seção 5.3 sumariza e comenta os principais resultados obtidos.

5.1 Metodologia

5.1.1 Parâmetros

A seguir descrevem-se os parâmetros utilizados na caracterização. Cada um desses parâmetros é medido para cada par ordenado de *sites* (origem, destino).

5.1.1.1 Distância média (D)

A distância média $d_{A,B}$ entre dois *sites* A e B é a média entre os comprimentos de todos os caminhos diferentes observados entre esses dois *sites*, partindo do *site* A e chegando ao *site* B. Esse comprimento corresponde ao número de roteadores que compõem o caminho.

Esse parâmetro está relacionado tanto com a diversidade de caminhos quanto com a taxa de perda de pacotes, dois aspectos importantes no roteamento de mídia contínua. Por um lado, é intuitivo que, quanto maior a distância entre dois *sites*, maior é a possibilidade de existirem diversos caminhos entre eles, caminhos que poderiam ser aproveitados em formas alternativas de roteamento. Por outro lado, quando o caminho entre o servidor e um cliente é muito longo, é mais provável que haja mais perda de pacotes.

Esse parâmetro possui também outras aplicações, como, por exemplo, estimar o diâmetro da Internet. Essa estimativa é usada para determinar o valor inicial do campo TTL (*time-to-live*) de um pacote IP a ser enviado pela Internet, de forma a garantir que esse pacote consiga chegar a qualquer outro nó da rede, mas também que o pacote seja descartado após um tempo, para não sobrecarregar a rede, caso se perca devido a alguma falha de roteamento. Outra análise interessante que pode ser feita com base nesse parâmetro é verificar se existe uma relação entre a distância "topológica" definida acima e a distância geográfica entre os *sites*.

5.1.1.2 Número de caminhos diferentes (C)

O número de caminhos diferentes $c_{A,B}$ existentes entre dois *sites* A e B é a quantidade de caminhos observados[1] que diferem em pelo menos um roteador.

A importância desse parâmetro está no fato de ele estar diretamente relacionado com a diversidade de caminhos. No entanto, deve-se notar que esse parâmetro, por si só, diz pouco a respeito da diversidade de caminhos existente entre dois *sites*, uma vez que diversidade de caminhos tem a ver também com o grau de disjunção entre os diferentes caminhos. Não se

[1] É importante frisar que todos os parâmetros da caracterização se referem a caminhos **observados**, isto é, correspondentes aos dados de *traceroute* coletados, mas diversos outros caminhos podem existir caso se considere a topologia montada a partir do conjunto de todas as rotas coletadas.

deve, portanto, considerar este parâmetro isoladamente, mas sim combiná-lo com o grau de diferença médio G, definido na seção 5.1.1.3.

Outros trabalhos, como o de Teixeira et al [59], tentam quantificar a diversidade de caminhos em termos de caminhos completamente disjuntos. Apesar de ser uma abordagem válida, é desnecessariamente restritiva. Em geral é raro encontrar sequer dois caminhos totalmente disjuntos entre dois *sites* na Internet. O importante para uma aplicação que tire proveito da diversidade de caminhos não é que dois ou mais caminhos entre o servidor e o cliente sejam completamente disjuntos, mas sim que haja disjunção nos *links* gargalo.

5.1.1.3 Grau de diferença médio (G)

Definimos o grau de diferença entre um par de caminhos da seguinte forma. Cada roteador desses dois caminhos é associado a um caractere. Um caminho, ou seqüência de roteadores, se traduz, portanto, em uma *string*. Determina-se a distância mínima de edição u, entre as *strings* correspondentes aos dois caminhos, isto é, o menor número de operações de troca, inserção ou remoção de caracteres necessárias para transformar uma *string* na outra. Esse valor consiste na distância de Levenshtein [35], um ou simplesmente "distância de edição". Essa distância, também usada por He et al [26, 25] para medir assimetria, pode ser calculada por um algoritmo de programação dinâmica *bottom-up* que é comprovadamente correto. Divide-se u por $max\{m, n\}$, onde m e n são os comprimentos de cada uma das *strings*. Isso porque o maior valor possível para u, dadas duas *strings* de quaisquer de comprimentos m e n, é justamente o comprimento da maior *string*. Fazendo-se essa divisão, temos um valor normalizado no intervalo [0, 1]. Se esse valor é igual a 0, os dois caminhos são completamente iguais. Se é igual a 1, são completamente diferentes (não possuem nenhum roteador em comum).

O grau de diferença **médio** $g_{A,B}$ de um par de *sites* A e B é a média dos graus de diferença entre cada par de caminhos diferentes existentes entre esse par de *sites*.

No contexto de roteamento de mídia contínua, esse parâmetro é importante pois, juntamente com o número de caminhos diferentes (seção 5.1.1.2), caracteriza a diversidade de caminhos na rede.

Uma forma mais ingênua de medir o grau de diferença entre caminhos consiste em contar o número de roteadores comuns aos dois caminhos e dividir pelo comprimento do maior caminho. A vantagem da nossa definição sobre esta é que a nossa leva em conta a ordem em que os roteadores aparecem.

5.1.1.4 Grau de assimetria médio (A)

O grau de assimetria entre um caminho de ida e um caminho de volta entre dois *sites* é calculado da mesma forma que o grau de diferença entre dois caminhos de ida (seção 5.1.1.3), porém com o caminho de volta invertido. Assim, se o grau de assimetria é igual a 1, isto significa que o caminho de volta é completamente diferente (não possui nenhum roteador em comum) do caminho de ida. Quando é igual a 0, o caminho é o mesmo, porém no sentido contrário. Essa forma de quantificar a assimetria é a mesma adotada por He et

al [26, 25]. Note que faz mais sentido comparar esses dois caminhos apenas se eles são observados aproximadamente no mesmo momento, já que é nessa situação que a assimetria pode ser um problema.

Para calcular o grau de assimetria **médio** $a_{A,B}$ para um par de *sites* A e B, comparamos cada caminho (A, B) observado em um momento t, com o caminho (B, A) cujo momento de coleta t' seja o mais próximo possível de t, até um limite máximo de 4 horas. Não sendo encontrado tal caminho (B, A) dentro desse limite, o caminho (A, B) em questão é desconsiderado. O grau de assimetria médio para o par (A, B) é dado pela média dos valores resultantes dessas comparações.

O fenômeno da assimetria de roteamento é importante pois "pode influenciar a maneira com a qual modelamos e simulamos a Internet" [25]. Balakrishnan et al [9] estudam os efeitos desse fenômeno sobre o desempenho do TCP. De maneira geral o desempenho desse importante protocolo é degradado pela existência de assimetria. Apesar de ser uma prática geral o uso de UDP para transmissão de mídia contínua, o canal de controle entre o cliente e o servidor é muitas vezes implementado sobre TCP. Mesmo que não seja, o mero fato de os fluxos de dados e de controle passarem por caminhos diferentes, com diferentes características de atraso, largura de banda e, em especial, de taxa de perdas, pode impactar no desempenho desse tipo de aplicação.

5.1.2 Metodologia geral para análise de cada parâmetro

A análise de cada um dos parâmetros descritos na seção 5.1.1 foi feita através de tabelas e funções de distribuição acumuladas plotadas para diversos conjuntos de *sites*. Os conjuntos de *sites* escolhidos são:

- mundo, ou seja, todos os 53 *sites* das tabelas 4.1 e 4.2;
- cada um dos continentes com pelo menos 3 *sites*: América do Norte, América do Sul, Ásia e Europa;
- e cada um dos países com pelo menos 3 *sites*: Brasil, EUA, França, Japão e Reino Unido.

Para os conjuntos correspondentes a países, os dados foram apresentados somente sob a forma de tabelas. Para os demais, foram apresentadas distribuições acumuladas, sumarizadas também nas tabelas. Não foram incluídos na análise pares de *sites* que, após as filtragens descritas na seção 4.4.2, apresentaram uma contagem de coletas de rota válidas inferior a 30.

Cada tabela apresenta dados relativos a um dos parâmetros, sendo cada linha da tabela correspondente a um dos conjuntos de *sites*. A primeira coluna apresenta o nome do conjunto de *sites* e, entre parênteses, o número total de *sites* naquele conjunto. A segunda coluna apresenta o número total de pares ordenados que entraram na análise, por apresentarem contagem de coletas de rota válidas igual ou superior a 30. A terceira coluna apresenta o valor médio da distribuição. A quarta e a quinta coluna apresentam, respectivamente, o

coeficiente de variação (CV) e o intervalo de confiança (IC) para o valor médio. Para todos os casos foi utilizado um grau de confiança igual a 95%.

O CV é dado por $\frac{\sigma}{\mu}$, onde σ é o desvio-padrão e μ é a média para dada distribuição. Esse coeficiente permite comparar distribuições estatísticas diferentes quanto à sua variabilidade. Além disso, um CV próximo de zero indica uma distribuição com variância baixa e, portanto, indica que a média é uma valor bastante significativo para a distribuição. Em outras palavras, faz sentido comparar duas distribuições simplesmente com base em seus valores médios, caso apresentem CV próximos de zero.

Note que o número de pares ordenados em uma região com n *sites* pode ser inferior a $n \times (n-1)$ devido à filtragem mencionada acima. Para os parâmetros "grau de assimetria médio" e "grau de diferença", esse número pode ser ainda menor. No primeiro caso, porque é possível que não sejam encontrados caminhos reversos correspondentes a um caminho de ida em todos os casos, devido à restrição que requer que as coletas tenham sido realizada com menos de 4 horas de diferença (veja seção 5.1.1.4). No segundo caso, porque para alguns pares ordenados de *sites* foi observado somente 1 caminho, e portanto não há "pares de caminhos diferentes" a serem comparados entre si.

5.1.2.1 Representatividade dos sites escolhidos

Apesar de nossa análise ter sido dividida com base em regiões geográficas, falando-se em termos de "diversidade de caminho na América do Sul" ou "distância média no Japão", deve-se estar atento ao fato de que não foi realizado um estudo que determine a representatividade do conjunto de *sites* escolhidos, isto é, se os *sites* escolhidos constituem uma amostra que reflete bem as características do todo. Assim, ao se dizer "diversidade de caminhos na América do Sul", deve-se automaticamente subentender "diversidade de caminhos **entre os *sites* escolhidos** na América do Sul". A frase é abreviada por uma simples questão de concisão mas não de generalização. Medir a representatividade estatística dos *sites* escolhidos em cada país e cada continente é uma tarefa nada trivial, deixada para trabalhos futuros. Conforme explicamos na seção 4.1, a escolha desses *sites* foi limitada pela disponibilidade de servidores públicos de *traceroute* em cada região.

5.2 Resultados

5.2.1 Distância média

Antes de se analisar a distância média, deve-se ter em mente que essa distância corresponde ao número de roteadores entre os **domínios** DNS dos servidores *traceroute* usados como pontos de coleta, devido à aglomeração de pontas explicada na seção 4.4.1. Assim, não estão sendo contados os saltos dentro desses domínios e a distância real entre os servidores é maior que a apresentada aqui. Verificou-se que a distância entre cada par de pontos de coleta é subestimada, em média, em 12% e no máximo em 47%. Em 90% dos pares de pontos de coleta a distância média é subestimada em menos de 25%.

Em ambos os gráficos da figura 5.1 pode-se notar que a distância média entre *sites* localizados em um mesmo continente é menor que a distância média entre todos os *sites* do mundo. Isso revela o já esperado fato de que a maioria dos caminhos mais longos ocorrem entre *sites* de continentes diferentes. Curiosamente, as distribuições das distâncias médias entre os *sites* dos continentes mostrados na mesma figura são parecidas, bem como a distância média, que está entre 11,1 e 12,7, conforme se vê na tabela 5.1, bem abaixo da média mundial, de 14,6.

Note, pela tabela 5.1, coeficiente de variação é baixo, entre 0,2 e 0,3 para a maioria dos conjuntos de *sites*, com exceção do Reino Unido. O fato de esse coeficiente ser baixo faz com que uma análise dos valores médios seja significativa.

Os EUA possuem um valor médio muito próximo da América do Norte, pelo fato de 12 dos 13 *sites* norte-americanos estarem localizados nesse país.

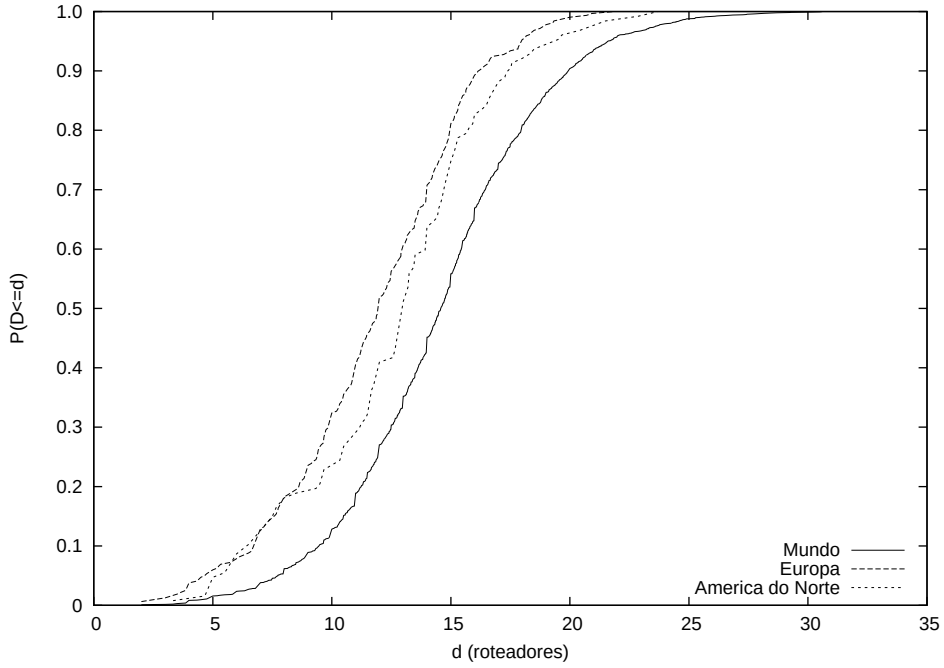
Vê-se ainda que, para as regiões menores, ou seja, os países exceto EUA, a distância média é ainda menor que para os continentes. Como exemplo, veja que a distância média entre os *sites* brasileiros é por volta de 8 saltos, enquanto a distância média para a América do Sul está por volta de 12. Verificou-se que isso se deve principalmente à existência de caminhos mais longos, por volta de 14 saltos (CV=0,17), entre os *sites* brasileiros e os *sites* não-brasileiros, assim como entre os *sites* não-brasileiros, por volta de 11 saltos (CV=0,32). Esse é um dos exemplos em que a dispersão geográfica está relacionada com a distância topológica, porém isso não é regra geral.

Foi calculado o coeficiente de correlação entre a distância geográfica e o parâmetro distância média que definimos. O coeficiente de correlação entre duas variáveis aleatórias X e Y (com valores médios iguais a μ_X e μ_Y e desvios-padrão iguais a σ_X e σ_Y) mede a intensidade e direção de uma relação de linearidade entre essas duas variáveis e é definido como $\rho_{X,Y} = \frac{E((X-\mu_X)(Y-\mu_Y))}{\sigma_X \sigma_Y}$. Esse coeficiente varia no intervalo $[-1, 1]$. Quanto mais próximo das extremidades do intervalo, maior é a intensidade dessa relação de linearidade. Quanto mais próximo de zero, menos correlacionadas são as variáveis.

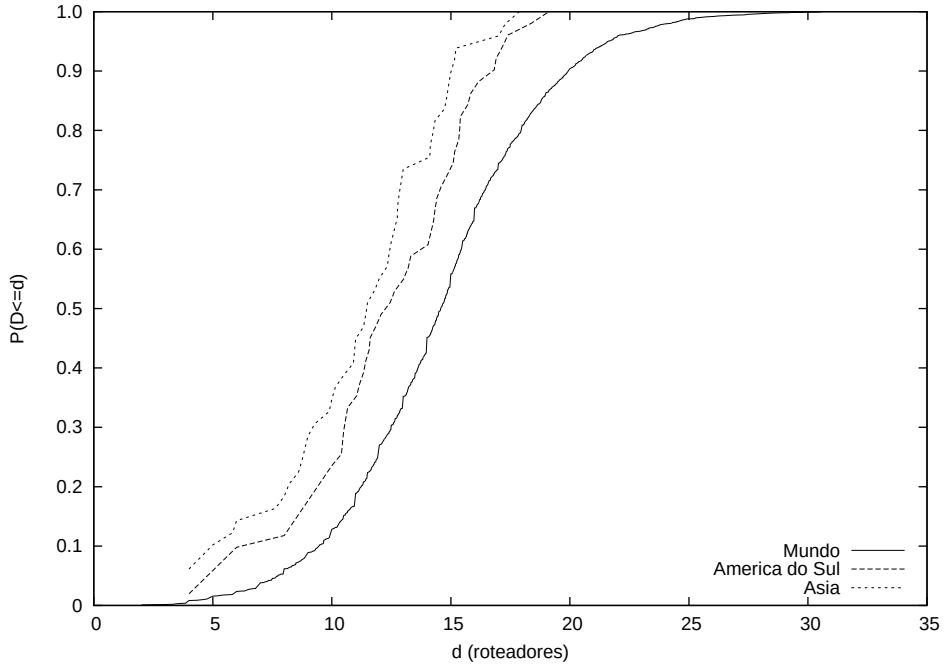
O grau de correlação ficou em $0,35$, um valor considerado médio-baixo. Isso pode se dever a *links* intercontinentais ou transcontinentais, que "aproximam", em 1 ou poucos saltos, regiões geograficamente distantes.

5.2.2 Número de caminhos diferentes

A figura 5.2 apresenta as funções de distribuição acumuladas do número de caminhos diferentes. Observando-se a curva "Mundo", percebe-se a existência de pares de *sites* com um número muito alto de caminhos diferentes, sendo que 3% dos pares possuem mais de 80 caminhos diferentes entre cada par de *sites*. Em 4 casos há mais que 100 caminhos. O maior valor (não mostrado na figura) é de 123 caminhos e ocorre entre o servidor neo-zelandês e o servidor do Rio de Janeiro. Verificou-se que a distância média entre esses dois *sites* é relativamente alta, 15,1 saltos, e o grau de diferença médio é baixo, cerca de 0,37. Isso pode indicar que mudanças freqüentes no roteamento podem ter acontecido apenas numa região restrita, próxima ao servidor do Rio de Janeiro, já que os outros três casos com mais de 100 caminhos também envolveram esse servidor.



(a) Distância Média D entre pares de *sites* - Mundo, Europa, América do Norte



(b) Distância Média D entre pares de *sites* - Mundo, América do Sul, Ásia

Figura 5.1: Distância média D entre pares de *sites*

Conjunto de *sites*	**Pares selecionados**	**Média**	**CV**	**IC da média**
Mundo (53)	2410	14.62	0.29	14.45 , 14.79
América do Norte (13)	127	12.74	0.32	12.02 , 13.46
América do Sul (8)	51	12.35	0.29	11.34 , 13.36
EUA (12)	105	12.00	0.32	11.25 , 12.76
Europa (20)	318	11.80	0.32	11.37 , 12.22
Ásia (8)	49	11.16	0.32	10.14 , 12.18
Japão (3)	5	9.72	0.14	7.98 , 11.47
Reino Unido (3)	6	9.44	0.59	3.61 , 15.28
Brasil (4)	11	8.32	0.25	6.91 , 9.72
França (3)	6	6.44	0.18	5.19 , 7.69

Tabela 5.1: Distância média

Como se vê pelas outras curvas, o número máximo de caminhos entre cada par não chega a ser tão alto para pares de servidores em um mesmo continente. O maior valor é 63 caminhos, ocorrido na América do Sul. A hipótese mencionada acima, da ocorrência freqüente de mudanças no roteamento próximo ao Rio de Janeiro, também ajudaria a explicar esse grande número de caminhos na América do Sul.

O continente que apresenta menor número de caminhos entre pares de *sites* é a América do Norte, fato provavelmente devido a uma maior estabilidade da rede entre esses *sites*, que são em sua maioria instituições de pesquisa e ensino superior, interconectadas por um *backbone* dedicado. Mas note que, mesmo nesse caso, mais de 70% dos pares de *sites* apresentam pelo menos dois caminhos diferentes entre cada par. No mundo inteiro, isso acontece com 82% dos pares de *sites*.

Assim como para a métrica "distância média", os maiores valores para "número de caminhos diferentes" ocorrem, em geral, em pares de *sites* que estão em continentes diferentes, como se vê pelo fato de a curva mundial estar mais deslocada para a direita (a exceção é a América do Sul, pelo motivo exposto acima). Podemos relacionar os valores altos para essas duas métricas ocorrendo entre pares intercontinentais da seguinte forma. A maioria dos caminhos longos ocorre entre continentes. Em caminhos longos há mais roteadores. Trocando-se um desses roteadores, cria-se um novo caminho diferente. Portanto, quanto maior a distância, maior é o potencial para aparecimento de novos caminhos diferentes, ainda que esses caminhos possam apresentar baixo grau de diferença médio entre si.

Pela tabela 5.2, percebe-se essa tendência se estendendo para conjuntos de *sites* referentes a países, ou seja, distância média mais baixa parece refletir em um menor número de caminhos diferentes.

Por fim, observa-se um coeficiente de variação relativamente alto para essa métrica, por volta de 1 para os conjuntos de *sites* correspondentes a continentes e ao mundo. Isso indica que enquanto alguns pares possuem muitos caminhos diferentes entre si, outros apresentam poucos. Analisou-se, como exemplo, a América do Sul (CV=1,00). Verificou-se a existência

de muitos caminhos entre *sites* brasileiros e *sites* não-brasileiros, sendo que todos os 25% pares de *sites* com maior número caminhos correspondem a essa situação. O fato de esses caminhos serem em geral mais longos, como se viu na seção 5.2.1, contribui para isso.

Assim, julgando por esse parâmetro isoladamente e considerando que a distância média entre *sites* é praticamente a mesma em todos os continentes, a diversidade de caminhos na América do Sul aparenta ser maior que nos outros continentes, em especial numa região próxima ao *site* do Rio de Janeiro. A menor diversidade seria encontrada na América do Norte. Ásia e Europa ficam mais ou menos empatados. É importante lembrar, no entanto, que este parâmetro, por si só, é insuficiente para caracterizar a diversidade de caminhos. Por isso, a seguir analisamos os resultados para o parâmetro "grau de diferença médio".

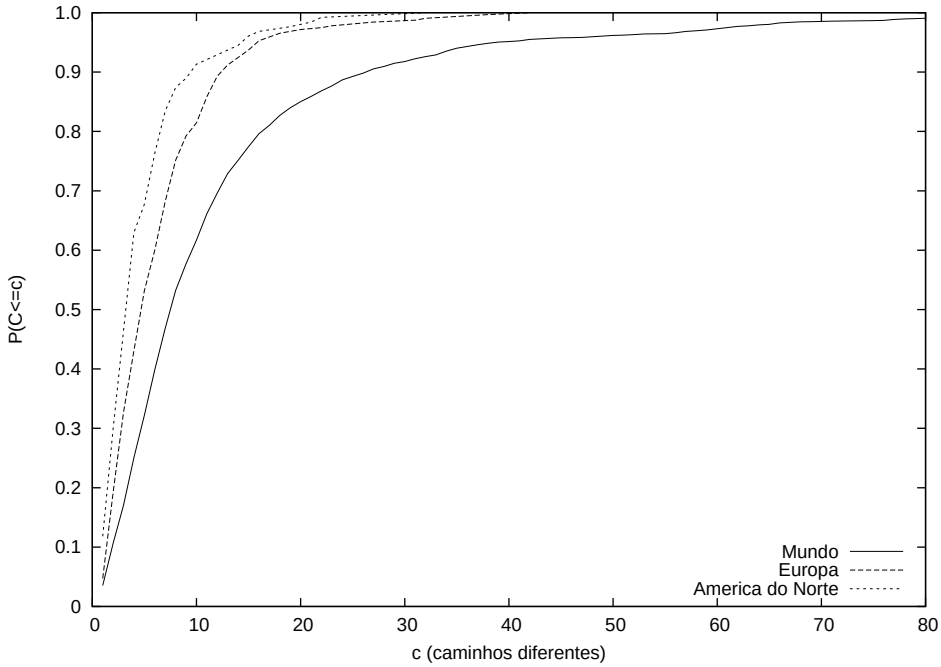
5.2.3 Grau de difereça médio

A figura 5.3 mostra as distribuições acumuladas do grau de diferença médio para o mundo e para os continentes com maior número de *sites*. O primeiro fato a se notar nessas curvas é que não existe grau de diferença médio igual a zero. Isso ocorre porque para se determinar o grau de diferença médio são comparados apenas pares de caminhos que apresentam alguma diferença entre si, portanto o valor do grau de diferença não pode ser zero.

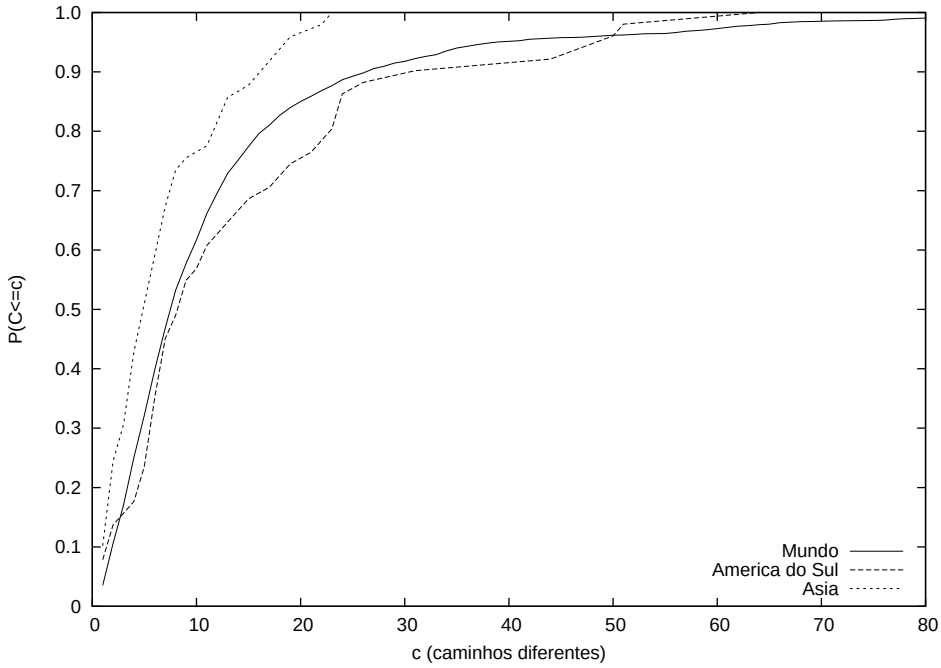
Por outro lado, existem pares de *sites* que possuem caminhos completamente disjuntos. Um limite inferior para o número de *sites* que apresentam pelo menos dois caminhos completamente disjuntos é dado pelos pares cujo valor do grau de diferença médio é 1. Esses pares de *sites* podem não corresponder a todos os pares que possuem pelo menos dois caminhos completamente disjuntos, porque o valor mostrado no gráfico é a média entre os graus de diferença entre cada par de caminhos. Assim, se um par de *sites* AB possui 3 caminhos entre eles, sendo que 2 desses caminhos são completamente disjuntos e o terceiro possui algum trecho compartilhado com o segundo, o valor do grau de diferença médio para esse par será inferior a 1, apesar de haver dois caminhos completamente disjuntos.

Vemos que nas curvas do mundo, da América do Norte e da Europa aparece o valor 1. Ocorre para 0,4% dos pares do mundo, incluindo os 0,9% dos pares norte-americanos e os 1,6% dos pares europeus. Essas porcentagens correspondem respectivamente, a 11, 1 e 5 pares. Portanto há pelo menos 5 pares de *sites* intercontinentais que apresentam caminhos completamente disjuntos. Um exemplo é o par *(www.washington.edu, lg.transtk.ru)*, que contém um servidor estadunidense e outro russo. Entre esse par de servidores aparecem dois caminhos, sendo um de comprimento 6 e outro de comprimento 16. O caminho de comprimento 6 apareceu muito mais freqüentemente nas nossas coletas, indicando que o outro é provavelmente uma rota alternativa que apareceu devido a algum problema transiente na rota *default*.

Se escolhermos arbitrariamente um valor mínimo maior que 0,8 para determinar quais pares de *sites* possuem alto grau de diferença médio, então não há pares na América do Sul nem na Ásia com alto grau e há apenas 12 pares europeus e 3 pares norte-americanos com alto grau de diferença médio, totalizando portanto 15 pares intracontinentais. Como no mundo inteiro há 40 pares de *sites* com grau de diferença médio maior que 0,8, então 25 desses pares



(a) Número C de caminhos diferentes entre pares de *sites* - Mundo, Europa, América do Norte



(b) Número C de caminhos diferentes entre pares de *sites* - Mundo, América do Sul, Ásia

Figura 5.2: Número C de caminhos diferentes entre pares de *sites*

Conjunto de *sites*	Pares selecionados	Média	CV	IC da média
América do Sul (8)	51	14.53	1.01	10.41 , 18.65
Mundo (53)	2410	12.66	1.16	12.07 , 13.25
Ásia (8)	49	7.10	0.82	5.44 , 8.77
Europa (20)	318	6.76	0.86	6.12 , 7.40
França (3)	6	5.5	0.43	3.04 , 7.96
América do Norte (13)	127	5.09	0.92	4.26 , 5.91
Brasil (4)	11	4.82	0.67	2.65 , 6.98
EUA (12)	105	4.81	0.99	3.88 , 5.73
Japão (3)	5	3.6	0.58	1.02 , 6.17
Reino Unido (3)	6	2.83	0.68	0.80 , 4.87

Tabela 5.2: Número de caminhos diferentes

são intercontinentais. Observa-se novamente que, em geral, a diversidade de caminhos é maior quando se consideram regiões geográficas mais dispersas. Isso pode ser observado também na tabela 5.3, onde se vê que os países possuem um grau de diferença médio inferior aos continentes.

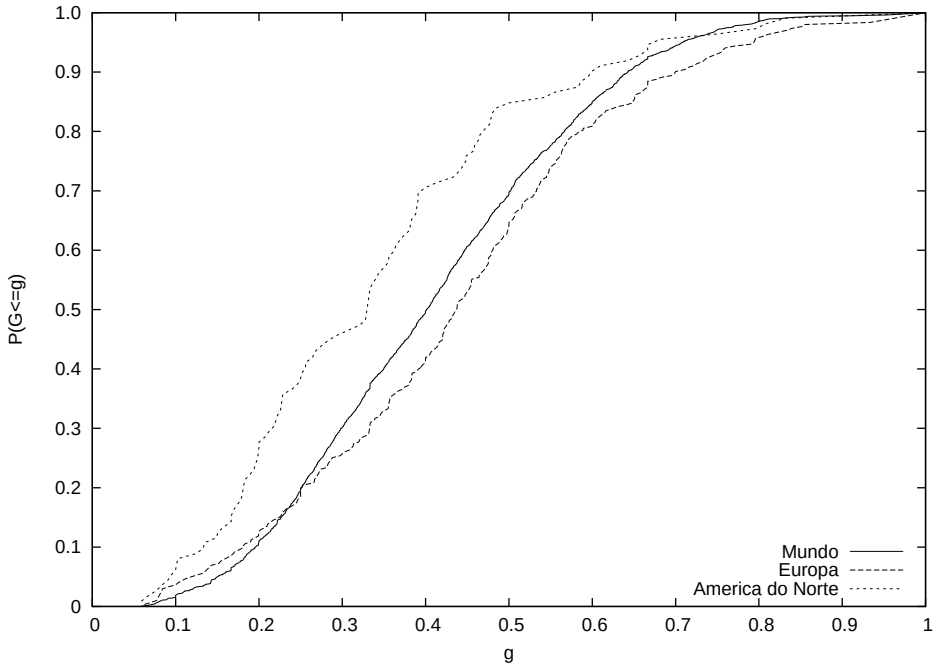
A situação é diferente quando se analisam valores intermediários de grau de diferença. Nesse caso, a América do Sul apresenta diversidade de caminhos acima da média mundial, seguida da Europa, também acima da média. A Ásia está em torno da média mundial, enquanto a América do Norte possui um grau de diferença bem mais baixo. Veja na tabela 5.3 os valores médios específicos de cada continente. Observa-se também que mais de 45% dos pares de *sites* da América do Sul possuem grau de diferença médio superior a 0,5, indicando a existência de caminhos razoavelmente disjuntos.

5.2.4 Grau de assimetria médio

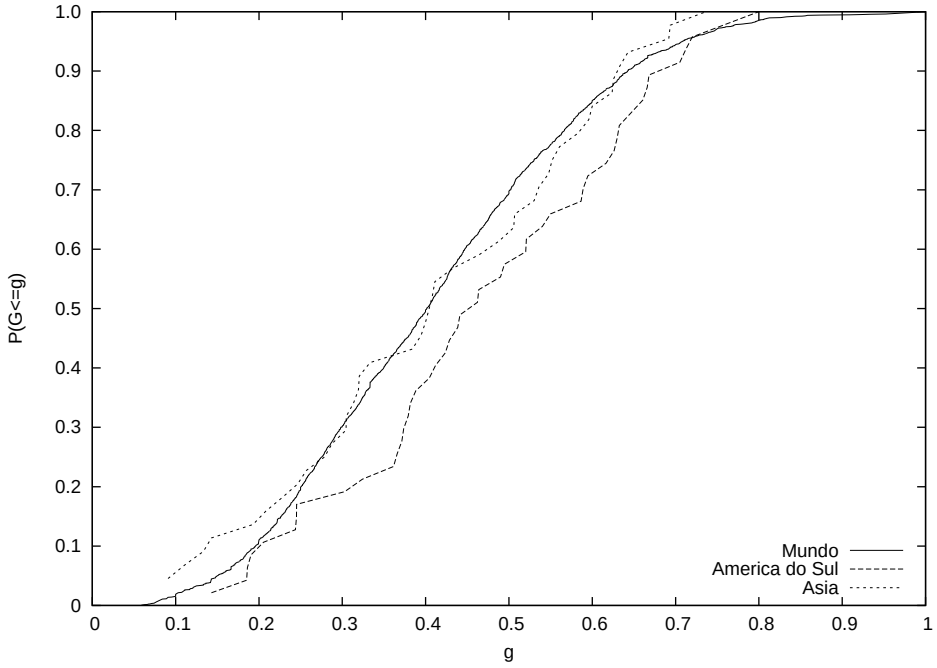
A figura 5.4 apresenta as distribuições acumuladas para o grau de assimetria médio nos continentes com maior número de *sites*.

Não foram encontrados pares de *sites* com caminhos de ida completamente simétricos com os caminhos de volta durante todo o tempo de coleta. Isso corresponderia a valores de assimetria média iguais a zero nas curvas acima. No entanto há alguns valores muito próximos de zero, revelando a existência de pares de *sites* entre os quais os caminhos de ida e volta foram quase completamente simétricos em todas as coletas realizadas. Cerca de 2% dos pares de *sites* da América do Sul e da Ásia possuem caminhos de ida e volta quase completamente simétricos, possuindo grau de assimetria médio inferior a 0,004. Essa porcentagem corresponde a 1 par de *sites* em cada um desses continentes. Isso ocorre em 0,2% dos pares de *sites* do mundo, o que corresponde a apenas 4 pares. Portanto, há apenas outros 2 pares de *sites* que são quase completamente simétricos e são, obviamente, intercontinentais.

Por outro lado, há um número bem maior de pares de *sites* apresentando caminhos completamente assimétricos. As porcentagens para América do Norte, América do Sul, Ásia,



(a) Grau G de diferença médio entre pares de caminhos diferentes entre pares de *sites* - Mundo, Europa, América do Norte



(b) Grau G de diferença médio entre pares de caminhos diferentes entre pares de *sites* - Mundo, América do Sul, Ásia

Figura 5.3: Grau G de diferença médio entre pares de caminhos diferentes entre pares de *sites*

Conjunto de *sites*	Pares selecionados	Média	CV	IC da média
França (3)	6	0.61	0.31	0.41 , 0.81
América do Sul (8)	47	0.47	0.37	0.42 , 0.52
Europa (20)	303	0.44	0.45	0.42 , 0.46
Ásia (8)	44	0.41	0.43	0.36 , 0.47
Mundo (53)	2325	0.41	0.42	0.40 , 0.42
Reino Unido (3)	4	0.39	0.76	0.0 , 0.87
Brasil (4)	8	0.38	0.43	0.24 , 0.51
Japão (3)	5	0.38	0.43	0.17 , 0.59
América do Norte (13)	112	0.34	0.54	0.31 , 0.38
EUA (12)	91	0.33	0.57	0.29 , 0.37

Tabela 5.3: Grau de diferença médio

Europa e Mundo são, respectivamente, 5%, 8%, 12%, 6% e 6%, correspondendo a 6, 4, 5, 16 e 132 pares de *sites*. Dos 132 pares de *sites* do mundo, há, portanto, 101 pares intercontinentais. Assim como para os outros 3 parâmetros, os valores maiores tendem a se concentrar em pares de *sites* intercontinentais.

Pelas distribuições acumuladas nota-se claramente que a assimetria é muito maior na América do Sul que nos outros continentes. Observa-se pela tabela 5.4 que o grau de assimetria médio para a América do Sul é 0,8, um valor muito alto. A média mundial é de 0,71 e a média dos demais continentes é inferior à mundial. Enquanto na América do Sul quase 70% dos pares de *sites* possuem grau de assimetria médio superior a 0,8, apenas 25% dos pares europeus e 20% dos norte-americanos possuem assimetria tão alta. Por outro lado, a Europa possui grau de assimetria menor que 0,3 em 12% dos pares de *sites*, enquanto isso acontece em apenas 6% dos pares sul-americanos.

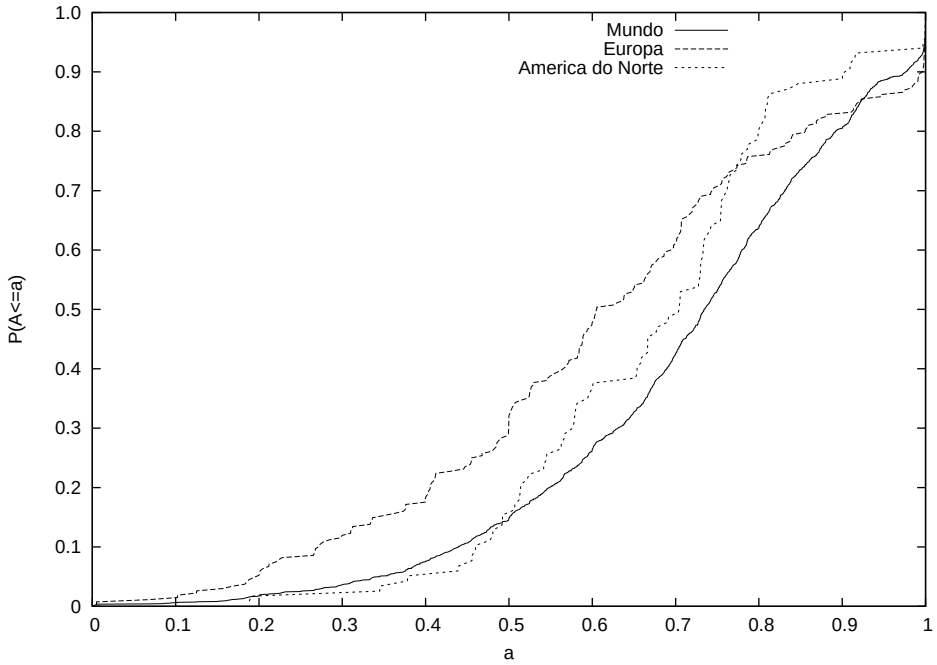
Com relação ao coeficiente de variação, ele é sempre baixo, exceto na Europa e na Ásia. Verificou-se que isso ocorre na Ásia principalmente por causa dos *sites* em Hong Kong e do *site* da Indonésia. Quase todos os 45% pares mais assimétricos envolvem um ou ambos os países.

Assim, a Europa e a América do Norte possuem menor grau de assimetria, e a América do Sul possui um grau de assimetria muito alto. No entanto, apesar dessa diferença, de maneira geral a assimetria é alta em todas as regiões, mesmo em países pequenos.

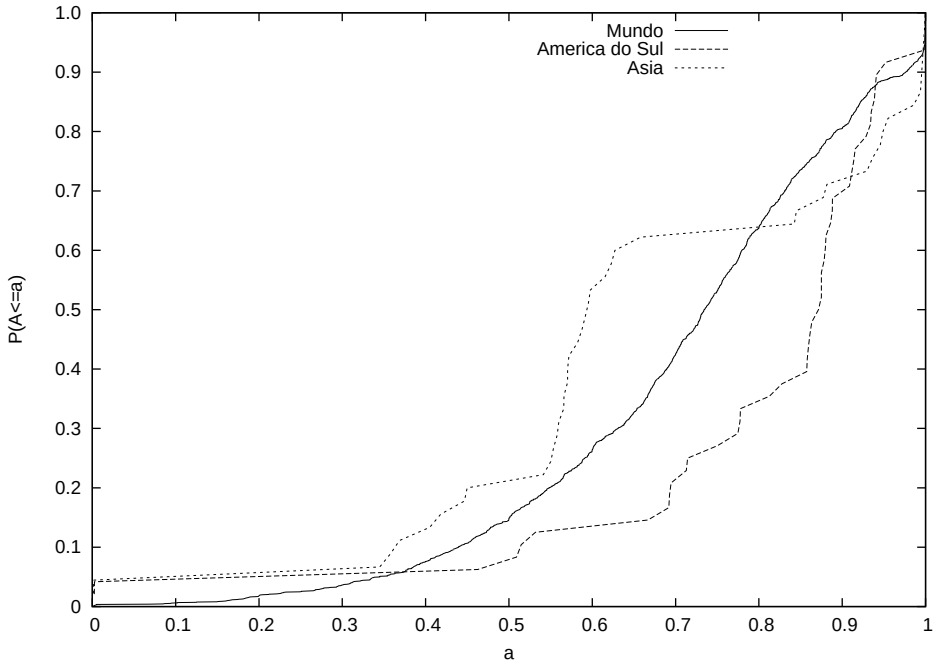
5.2.5 Correlação entre parâmetros relacionados à diversidade de caminhos

Foi sugerido que pudesse haver alguma correlação[2] entre os dois parâmetros principais relativos à diversidade de caminhos, "grau de diferença médio" e "número de caminhos diferentes". Mediu-se então a correlação entre esses dois parâmetros, usando os dados relativos a todos os *sites*. Tivemos como resultado uma correlação praticamente nula, igual a 0,0147. A figura 5.5

[2]Coeficiente definido na seção 5.2.1.



(a) Grau A de assimetria médio entre caminhos de ida e volta observados no mesmo momento (+- 4h) entre pares de *sites* - Mundo, Europa, América do Norte



(b) Grau A de assimetria médio entre caminhos de ida e volta observados no mesmo momento (+- 4h) entre pares de *sites* - Mundo, América do Sul, Ásia

Figura 5.4: Grau A de assimetria médio entre caminhos de ida e volta observados no mesmo momento (+- 4h) entre pares de *sites*

Conjunto de *sites*	**Pares selecionados**	**Média**	**CV**	**IC da média**
França (3)	6	0.81	0.19	0.65 , 0.97
América do Sul (8)	48	0.80	0.27	0.73 , 0.86
Brasil (4)	10	0.75	0.22	0.63 , 0.87
Mundo (53)	2201	0.71	0.28	0.70 , 0.72
EUA (12)	95	0.70	0.23	0.67 , 0.73
América do Norte (13)	117	0.67	0.25	0.64 , 0.71
Ásia (8)	45	0.67	0.39	0.59 , 0.74
Europa (20)	268	0.62	0.39	0.59 , 0.65
Reino Unido (3)	6	0.58	0.57	0.23 , 0.92
Japão (3)	5	0.35	0.91	0.0 , 0.75

Tabela 5.4: Grau de assimetria médio

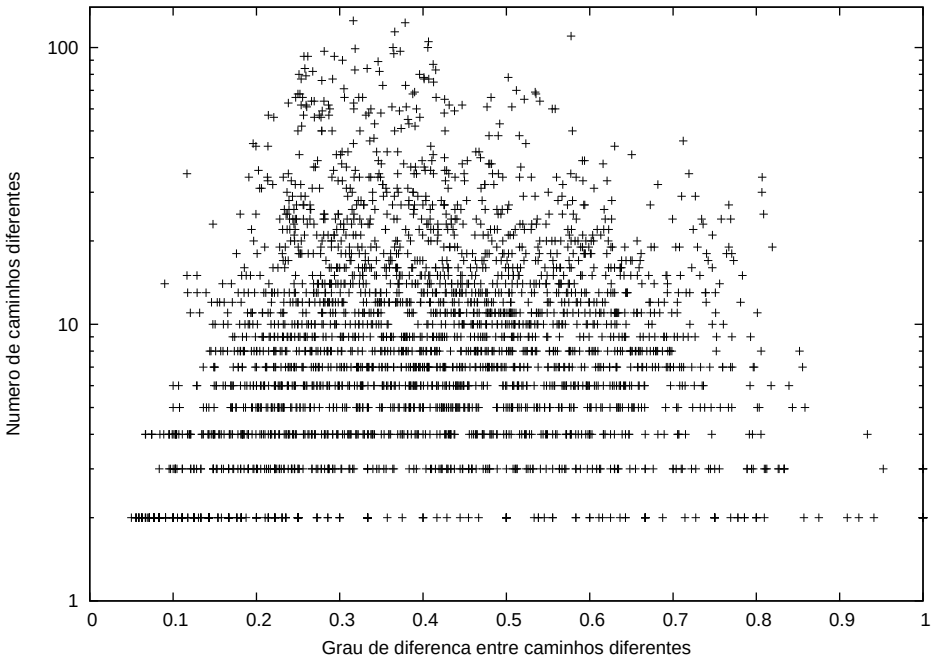


Figura 5.5: Grau de diferença médio *versus* Número de caminhos diferentes

plota os dois parâmetros, refletindo a baixa correlação. O "número de caminhos diferentes" foi plotado numa escala logarítmica por questões de clareza. Esse resultado indica que os parâmetros não estão correlacionados e sim medem grandezas ortogonais. Um trabalho futuro interessante seria relacionar esses dois parâmetros de forma a se obter um único parâmetro, função desses dois, que medisse o aspecto da diversidade de caminhos.

5.3 Conclusões

Com relação à assimetria, observamos que é alta, mesmo entre *sites* próximos geograficamente. Esse resultado está de acordo com os resultados de He [25], que também analisou a assimetria de roteamento em topologias globais para dois conjuntos de dados diferentes, sendo um deles mais concentrado em instituições de ensino superior e pesquisa da América do Norte e o outro mais espalhado pelo mundo, em *sites* comerciais. Apesar da utilização de metodologias diferentes, os resultados foram parecidos, ambos apontando para um grau de assimetria alto.

Quanto à diversidade de caminhos, analisando os parâmetros "número de caminhos diferentes" e "grau de diferença médio", percebemos que de maneira geral existe uma diversidade de caminhos alta no mundo, quando se consideram regiões geográficas mais abrangentes, como continentes. Percebe-se também que a diversidade é mais baixa quando se consideram pequenos conjuntos de *sites* geograficamente próximos. A diversidade de caminhos é especialmente alta entre os *sites* da América do Sul e é mais baixa na América do Norte, quando se comparam os quatro continentes com maior número de pontos de coleta.

À primeira vista é um resultado surpreendente, pois esperava-se que na América do Norte a diversidade de caminhos fosse maior. Uma possível explicação para essa menor diversidade na América do Norte é o fato de que a maioria dos *sites* daquele continente correspondem a instituições de ensino superior ou centros de pesquisa, ao contrário do que acontece nos outros continentes (veja a seção 5.1.2.1). Essas instituições são provavelmente interconectadas por um ou poucos *backbones* não-comerciais, de uso exclusivo para pesquisa. Portanto, a rede entre essas instituições é mais estável, há menos mudança no roteamento e isso faz com que a diversidade de caminhos aparente não ser tão grande. Já nas outras regiões do mundo, os *sites* provavelmente estão ligados a diversos provedores de acesso comerciais diferentes. Os caminhos envolvem vários sistemas autônomos distintos e a relação de provedor/cliente entre os sistemas autônomos também pode mudar bastante ao longo dos 4 meses de coleta.

É importante notar ainda que, além de ser alta a diversidade de caminhos de uma maneira geral, ela foi medida somente com relação a caminhos **observados** em *traceroutes*. Se fossem considerados outros caminhos, como por exemplo aqueles que se podem obter através de redes *overlay* entre os *sites* analisados, ou, mais ainda, todos os outros caminhos que surgem quando se monta um mapa topológico usando como entrada todos os dados de *traceroute* coletados, a diversidade de caminhos é, com certeza, ainda maior. Em outras palavras, a nossa medida é um limite inferior na real diversidade de caminhos existente e mesmo assim é um valor alto.

Essa alta diversidade de caminhos incentiva o uso de técnicas que tirem proveito disso, seja para melhorar a qualidade de serviço, seja para diminuir o custo de rede. Neste sentido, analisaremos, no capítulo 6, as potencialidades de aplicação de protocolos de roteamento para múltiplos clientes, protocolos estes que usam compartilhamento de fluxos para diminuir a banda de rede. A alta diversidade de caminhos existente permite a criação de várias florestas de distribuição alternativas à floresta dos menores caminhos. Como veremos no capítulo 6, essas florestas de distribuição alternativas possuem muitas vezes custo menor que a floresta dos menores caminhos. É essa a importância da diversidade de caminhos no roteamento de

mídia contínua.

Já a análise detalhada das potencialidades da aplicação de técnicas para melhorar a qualidade de serviço e/ou para aumentar a banda entre dois pontos da rede é deixada para trabalhos futuros. No entanto, sabe-se de antemão que existe essa possibilidade de melhoria na qualidade de serviço e no aumento de banda agregada com a utilização de múltiplos caminhos simultâneos para rotear fluxos de mídia contínua entre dois pontos em uma rede, conforme foi mostrado por Golubchik et al [24]. Assim, futuramente deve-se apenas quantificar essa melhoria.

Além disso, foi medida a correlação geográfica entre o parâmetro "distância média" e a distância geográfica, considerando-se todos os 53 *sites*. Observamos que a correlação é baixa. Isso significa que, ao analisarmos as potencialidades para aplicação de protocolos de roteamento para múltiplos clientes, devemos levar em conta também os aspectos topológicos ao invés de simplesmente os geográficos.

Como trabalhos futuros dessa caracterização, pretendemos descobrir uma maneira mais sistemática de relacionar os parâmetros "número de caminhos" e "grau de diferença", de forma a obter uma única métrica que permita comparar melhor a diversidade de caminhos existente nas diversas regiões. Além disso, pretendemos utilizar a metodologia de He [25] em nossos dados, a fim de comparar melhor as duas formas de se medir a assimetria. O interessante dessa metodologia é o fato de ela dispensar o processo de resolução de interfaces sinônimas descrito na seção 4.3.1. Mas note que esse processo ainda é necessário para se obter uma topologia precisa ao se combinarem todas as rotas coletadas entre os diversos *sites*.

Capítulo 6

Roteamento para múltiplos clientes

Este capítulo analisa protocolos de roteamento para múltiplos clientes baseados em compartilhamento de fluxos, em um grande número de configurações reais e sintéticas.

A análise dos protocolos de roteamento é feita comparando-se o custo de roteamento, em termos de banda média de rede, entre as florestas de distribuição geradas por esses protocolos. Utiliza-se, como referência, o proposto padrão [21, 41] para se construir essas florestas de distribuição. Diversos parâmetros podem influenciar no ganho obtido por um protocolo, dentre eles o número de réplicas, o número de *sites* participantes, a heterogeneidade de demanda, a concentração relativa entre réplicas e *sites* clientes e o grau de dispersão da topologia. Testamos protocolos de roteamento em um grande número de configurações a fim de tentar entender como cada parâmetro influencia no custo relativo de rede entre os protocolos. Além disso, analisamos individualmente diversas florestas de distribuição criadas, a fim de entender por que alguns dos protocolos propostos apresentam ganhos consideráveis em relação ao protocolo convencional.

6.1 Metodologia

6.1.1 Protocolos avaliados

Os protocolos de roteamento criam a floresta de distribuição de mídia contínua. Eles recebem como entrada um grafo correspondente à rede, um conjunto de nós que são servidores e um conjunto de nós correspondentes a *sites* clientes, juntamente com a demanda destes. Em seguida, escolhem em qual servidor cada cliente se conectará e qual o caminho pelo qual se dará essa conexão. O conjunto de nós servidores é determinado por protocolos de localização de réplicas. O foco deste trabalho é nos protocolos de roteamento, portanto será assumido um único protocolo de localização de réplicas.

Para localização de réplicas será usado o protocolo *Min-cost TSP*, definido na seção 3.3.1. No caso do protocolo *default* da Internet, isto é, localização *Min-cost TSP* e roteamento SP [37, 21, 41], utilizaremos o custo para *unicast* como função objetivo do protocolo de localização. Para os demais protocolos, utilizaremos como função objetivo para localização o

Função objetivo da localização	Protocolo de roteamento	Nome identificador
Unicast	SP	"protocolo convencional"
BS	SP	SP ("protocolo escalável")
BS	MCO	MCO ("protocolo escalável")
BS	MCI	MCI ("protocolo escalável")

Tabela 6.1: Protocolos de roteamento avaliados

custo para compartilhamento de fluxos usando BS [19] como protocolo de entrega. O cálculo do custo para ambos os casos está definido na seção 2.2.2.

Os protocolos de roteamento avaliados são o MCI, o MCO e o SP, definidos na seção 3.3.2. O protocolo de roteamento utilizado como referência, com o qual os outros serão comparados, é o SP, com função objetivo de localização de réplicas otimizada para *unicast*. Para evitar confusão deste protocolo com o protocolo de roteamento SP com função objetivo de localização de réplicas otimizada para compartilhamento de fluxos, chamaremos, daqui em diante, de **protocolo convencional** o protocolo com localização otimizada para *unicast*. Sempre que nos referirmos ao protocolo SP, estaremos falando daquele cuja localização é otimizada para compartilhamento de fluxos. Ao nos referirmos de uma maneira geral aos protocolos MCI, MCO e SP, usaremos o termo **protocolos escaláveis**. A tabela 6.1 resume essas definições.

6.1.2 Métricas

A métrica de principal interesse para se compararem protocolos cujo objetivo é otimizar o roteamento é o consumo de banda média de rede nas florestas de distribuição criadas por esses protocolos. Além disso, não é do nosso interesse avaliar árvores de distribuição que utilizem *unicast*, uma vez que o protocolo ótimo para *unicast* já é bem conhecido e aplicado na prática. Nosso interesse, portanto, é no roteamento utilizando compartilhamento de fluxos. Assim, a banda média de rede é calculada para cada *link* da floresta utilizando a fórmula para compartilhamento de fluxos apresentada na seção 2.2.2. O custo de uma floresta é a soma das bandas médias em cada um de seus *links*, uma vez que assumimos que o custo de roteamento é o mesmo em qualquer dos *links*.

Todos os protocolos de roteamento serão comparados com o protocolo convencional. Portanto, ao invés de mostrarmos a banda média de rede para as florestas criadas por cada um dos protocolos, na maioria das vezes utilizaremos como métrica o **ganho** percentual em banda média de rede dos protocolos avaliados, MCI, MCO e SP, em relação ao protocolo convencional. Essa normalização permite comparar entre si situações de diferentes ordens de grandeza da banda de rede média (por exemplo, quando a demanda total é diferente).

6.1.3 Simulação dos protocolos de roteamento

Para avaliar os protocolos de roteamento, implementou-se um simulador desses protocolos. O simulador recebe como entrada:

- um grafo, correspondente à rede IP existente entre os *sites*, sendo os vértices correspondentes aos nós da rede e as arestas correspondentes a *links*;

- um conjunto *Clientes* de vértices desse grafo que funcionarão como *sites* clientes, e a demanda N_i de cada cliente i desse conjunto ;

- um conjunto de vértices S desse grafo que **podem** funcionar como servidores, ou seja, os *sites* nos quais podem ser alocadas réplicas;

- um inteiro m, maior que zero e menor ou igual a $|S|$, que é o máximo de réplicas a serem alocadas;

- o nome da função objetivo do protocolo de localização *Min-cost TSP*, isto é, "unicast" ou "bs" (Bandwidth Skimming);

- os nomes dos protocolos de roteamento a serem simulados: "SP", "MCO" ou "MCI".

A saída do simulador corresponde às florestas de distribuição geradas por cada protocolo, juntamente com o custo dessas florestas.

Considera-se que todos os nós (roteadores ou *sites*) são capazes de bifurcar um fluxo de entrada em dois ou mais fluxos de saída, isto é, permitem o compartilhamento de fluxos de rede.

O simulador executa então o protocolo de localização *Min-cost TSP* com a função objetivo selecionada, determinando assim a localização da primeira, segunda, ..., m-ésima réplicas, dentre as possíveis localizações fornecidas para o simulador. Para cada nova réplica escolhida, são executados a partir do zero cada um dos protocolos de roteamento selecionados[1]. As florestas de distribuição geradas e seus respectivos custos são impressas em um arquivo de saída.

A execução de todos esses protocolos envolve a determinação do caminho mais curto entre vários pares de pontos na rede fornecida como entrada. Esse caminho mais curto é determinado com o uso do algoritmo de Dijkstra, não correspondendo necessariamente ao caminho mais curto observado nos *traceroutes* coletados. Além disso, os protocolos MCO e MCI precisam determinar o custo de árvores de distribuição, para assim poderem fazer suas escolhas. Esse custo é calculado como o custo de rede para compartilhamento de fluxos descrito na seção 2.2.2.

Para o protocolo convencional, foram passados para o simulador os parâmetros "unicast" e "SP". Para os protocolos escaláveis, utilizamos os parâmetros "bs", "SP", "MCO" e "MCI".

[1] A execução do protocolo de localização *Min-cost TSP* já envolve a construção de florestas SP, e portanto este protocolo de roteamento não precisa ser realmente executado, bastando imprimir as florestas geradas pelo protocolo de localização.

6.1.4 Parâmetros variados

Para entender em que situações os protocolos escaláveis superam o convencional, diversos parâmetros foram variados. Esta seção descreve esses parâmetros e justifica a necessidade de variação de cada um deles.

Número de réplicas (m) Um dos parâmetros que deve ser variado é a quantidade de réplicas. Simulações realizadas por Zhao et al [70] revelam que o ganho do protocolo escalável sobre o convencional, quando se utiliza apenas um servidor, geralmente é inferior a 10%. No entanto, Almeida et al [4] verificam que, em se aumentando o número de réplicas, os ganhos podem ser maiores. A quantidade ótima de réplicas a ser utilizada é aquela que minimiza os custos totais de distribuição, ou seja, o custo de rede mais o custo de servidores. Esse custo total é uma função do custo por unidade de banda de rede e do custo por unidade de banda de servidor. Uma vez que a relação entre custo de rede e custo de servidor não é clara, acreditamos que é interessante estudar os gastos com banda de rede variando-se a quantidade de servidores réplicas, para todas as configurações simuladas.

Heterogeneidade de demanda (N) Os protocolos escaláveis de criação de florestas de distribuição, MCI e MCO, levam em conta a demanda de cada *site* cliente na hora de se criar a floresta de distribuição. Por esse motivo, achamos interessante variar a heterogeneidade de demanda entre os diferentes *sites* clientes. Essa variação de demanda é algo que pode acontecer naturalmente na distribuição de mídia contínua. Determinado conteúdo pode ser mais interessante para certa "comunidade" de clientes do que para outra e, além disso, essas comunidades podem ter tamanhos diferentes, o que obviamente influencia suas demandas.

Concentração relativa entre servidores e clientes É natural que ocorra uma concentração de servidores em determinadas regiões, em especial ao se analisar a disposição geográfica, ao invés de topológica, dos *sites* participantes. Por exemplo, supondo-se que determinada empresa alemã esteja transmitindo ao vivo a copa do mundo da Alemanha através da Internet, essa empresa pode colocar vários servidores espalhados pelo território alemão para servir seus clientes. Isso não impede, porém, que clientes em outras partes do mundo se conectem a esses servidores alemães. Neste caso, teríamos os servidores concentrados na Alemanha e os clientes espalhados pelo resto do mundo. Quando acontece essa concentração de servidores, em muitas situações, como a mostrada na figura 6.1, os protocolos MCI e MCO podem ser melhores que o SP, em termos de economia de banda média de rede. Veremos que isso ocorre porque, enquanto a banda para compartilhamento de fluxos cresce linearmente com a distância, ela cresce apenas logaritimicamente com a demanda, o que quer dizer que é melhor dar preferência ao compartilhamento, como fazem o MCI e o MCO, do que à distância, como faz o SP.

Número de *sites* participantes (n) O número de *sites* participantes também pode influenciar no ganho. Se na figura 6.1 houvesse um terceiro *site* cliente no Brasil, o ganho do

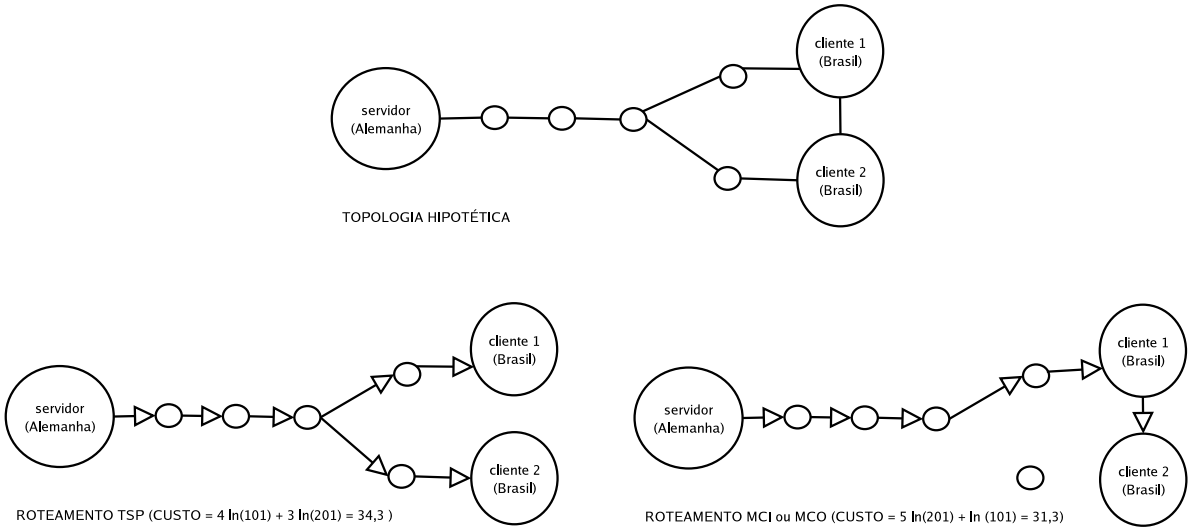


Figura 6.1: Servidores concentrados na Alemanha, clientes no Brasil com demanda $N_b = 100$ (Situação hipotética)

protocolo MCO sobre o SP poderia ser ainda maior (dependendo da configuração). Por outro lado, numa situação em que haja somente 1 servidor e 1 cliente, em geral não há ganho do MCO sobre o SP.

Dispersão A dispersão, definida pela distância[2] média entre os *sites* participantes, é um outro parâmetro que pode influenciar no ganho. Se na figura 6.1 houvesse 3 (em vez de 2) *links* entre a bifurcação e cada um dos *sites* clientes e a distância entre os clientes permanecesse a mesma, o ganho seria maior. O custo de entrega usando o roteamento SP passaria a ser de 43,6, enquanto o custo do MCO seria 36,4, ou seja, um ganho de 16%. Na situação mostrada na figura os custos são de 34,3 e 31,3, portanto o ganho é de apenas 8%.

6.1.4.1 Metodologia para variação dos parâmetros

A variação dos parâmetros "Número de *sites*" e "Dispersão" ocorre naturalmente ao se usarem as várias **configurações** obtidas, descritas na seção 6.1.5. Já os parâmetros "Concentração relativa entre servidores e clientes" e "Heterogeneidade de demanda" tiveram que ser variados em cada uma das configurações. Cada configuração consiste em um conjunto de *sites* participantes. Portanto, precisou-se escolher as demandas de cada *site*, além de fixar o papel (cliente, servidor, ambos) de cada *site*. Isso foi feito através da utilização de diversos **tipos de teste** para cada configuração. Esses tipos de teste estão descritos na seção 6.1.6.

[2] A unidade de distância pode ser geográfica (km), topológica observada (saltos nas de rotas coletadas com *traceroute*) ou topológica global (saltos pelo caminho mais curto usando a rede gerada a partir de todos os dados obtidos), dependendo do tipo da análise sendo feita e do processo correspondente que levou à criação da configuração (conjunto de *sites*) em questão. Isso ficará mais claro na seção 6.1.5.1.

6.1.5 Configurações testadas

Chamamos de "configuração" um conjunto de *sites*, clientes ou servidores, mais a rede em si. Esses *sites*, no caso das configurações reais (não sintéticas), correspondem um subconjunto dos pontos de coleta de *traceroutes* apresentados nas tabelas 4.1 e 4.2. Com o objetivo de analisar os protocolos na maior variedade de situações possível, automatizou-se o processo de geração das configurações a serem testadas, conforme se descreve nas subseções a seguir.

6.1.5.1 Sites clientes e servidores

Os *sites* de uma determinada configuração são simplesmente um subconjunto do conjunto de *sites* original, isto é, dos pontos de coleta de *traceroutes*.

Uma forma de se escolherem tais subconjuntos é através de um processo de aglomeração. Assim, por exemplo, se aglomerarmos os *sites* tomando como base a distância geográfica, poderemos ter 6 subconjuntos, cada um correspondendo a um continente. De fato, esse foi o primeiro tipo de aglomeração utilizado: os 53 *sites* existentes foram aglomerados em 6 subconjuntos, sendo cada subconjunto criado correspondente aos *sites* de um dos continentes (África, América do Norte, América do Sul, Ásia, Europa e Oceania).

Além disso, como gostaríamos de testar configurações de variados graus de dispersão e diferentes quantidades de *sites*, utilizamos o processo de aglomeração não apenas para dividir os *sites* em 6 subconjuntos como se mencionou acima, mas para dividi-los em 2, 3, 4, até 12 subconjuntos de *sites*. Esse processo foi automatizado com a utilização do programa *cluto* [32], que serve para aglomerar ("clusterizar") dados. A figura 6.2 ilustra esse processo.

O programa *cluto* oferece diversos métodos de aglomeração, sendo que, dependendo do conjunto de dados, alguns métodos são melhores que outros, pois geram "melhores" agrupamentos. Utilizamos como teste um conjunto de dados que corresponde às distâncias, em quilômetros, entre os 53 *sites*. Ao aglomerarmos esses dados em 6 subconjuntos de *sites*, cada subconjunto correspondeu exatamente aos *sites* de um dos seis continentes. Por esse motivo, acreditamos que o método de aglomeração escolhido é capaz de gerar bons agrupamentos para nosso conjunto de dados.

Como foi observado na seção 5.3, apesar de ser interessante analisar os dados geograficamente, não parece haver uma relação forte entre a geografia e a topologia de rede. Por esse motivo, realizamos outros dois tipos de aglomeração que julgamos interessantes, por levarem em conta a topologia da rede, em vez de dados geográficos. Assim, foram realizados os seguintes tipos de aglomeração:

Distância geográfica realizada com base na distância geográfica entre os *sites*, medida em quilômetros com o auxílio dos *websites* [23] e [29].

Distância observada com base na distância (em número de saltos) média observada nas rotas coletadas entre os *sites*.

Distância global com base na distância pelo menor caminho entre os *sites* na rede global (veja a definição de rede global na seção 6.1.5.2)

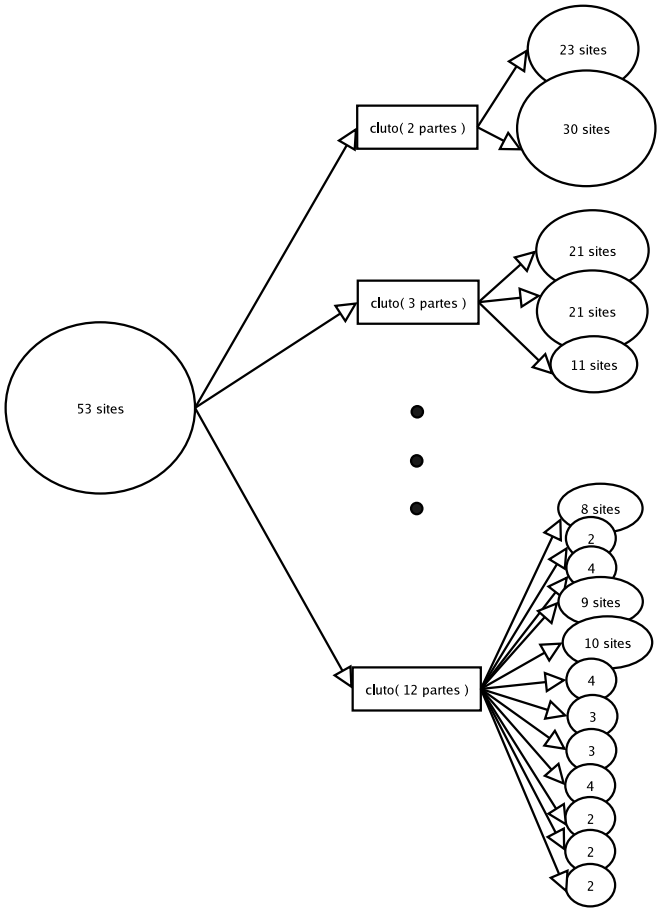


Figura 6.2: Aglomeração dos *sites* usando distância geográfica.

Para as redes sintéticas, foi realizado somente o correspondente à aglomeração por distância global, já que não existem dados geográficos ou de *traceroute* em uma topologia sintética.

6.1.5.2 Redes

Em grande parte dos testes com topologias reais usou-se, como rede IP, o que chamamos de **rede global**, a qual foi obtida a partir de todas as seqüências de roteadores resultantes da fase de coleta. A rede global consiste em um grafo direcionado, onde cada vértice representa um roteador (ou *site*) e cada aresta corresponde a um *link*. Assumiu-se que existe um *link* entre dois roteadores que aparecem adjacentes em pelo menos uma das seqüências de roteadores obtidas no processo de coleta e filtragem.

Em outras configurações reais usamos a **rede local**. Usamos esse termo num sentido bem diferente do tradicional. Uma rede local é gerada da mesma forma que a rede global, porém somente se utilizam as seqüências de roteadores coletadas **entre** os *sites* (clientes ou servidores) participantes da configuração em questão. Isso garante que, ao se analisar um subconjunto dos *sites*, não sejam tomadas rotas passando por *links* que não tenham sido observados nos *traceroutes* entre esses *sites*. Esse tipo de rede foi criada para configurações originadas de

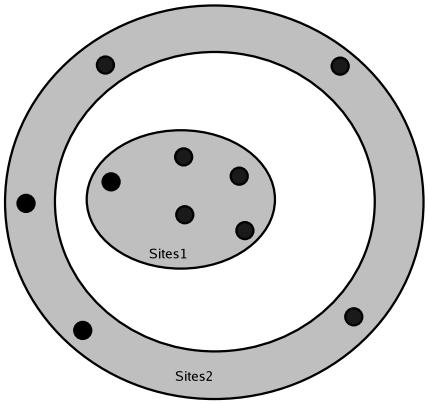


Figura 6.3: Típica aglomeração em 2 partes, Sites1 e Sites2

aglomeração com base na distância geográfica e na distância observada (seção 6.1.5.1).

Vários testes também foram realizados usando redes sintéticas, geradas usando o gerador *GT-ITM* [12]. Foram geradas 10 redes *transit-stub* [68], em nível de roteadores, de 600 nós cada. Em cada uma dessas redes, foram escolhidos aleatoriamente 60 nós para fazerem o papel de *sites*.

6.1.6 Tipos de teste para dada configuração

Com o objetivo de variar os parâmetros "Heterogeneidade de demanda" e "Concentração relativa entre servidores e clientes", foram criados 13 tipos de testes a serem realizados em cada uma das configurações. Para tanto, o conjunto de *sites* de cada configuração foi dividido em 2 subconjuntos, Sites1 e Sites2, utilizando o mesmo método de aglomeração utilizado para se escolher os *sites* dessa configuração. Convencionou-se adotar o nome Sites1 para o subconjunto que tivesse menor diâmetro (máxima distância entre 2 dos *sites*) e o nome Sites2 para os *sites* restantes. Na maioria dos casos de aglomeração baseada em topologia ao invés de geografia, o processo de aglomeração em 2 subconjuntos levou à criação de um subconjunto mais coeso (realmente um aglomerado) e outro subconjunto composto pelo restante dos *sites*, conforme mostra a figura 6.3.

Os tipos de teste criados estão descritos na tabela 6.2. Por exemplo, o teste 6 utiliza como réplicas *sites* no grupo Sites2 e os clientes são todos os *sites* do grupo Sites1, cada cliente i tendo demanda $N_i = 1000$.

Para entender como a variação da concentração relativa entre servidores e clientes atua sobre o ganho do protocolo escalável sobre o convencional, usamos as seguintes seqüências de testes:

testes 1, 2 e 3 Servidores são os Sites1 e a seqüência representa o afastamento dos clientes em relação ao servidores.

testes 4, 5 e 6 Servidores são os Sites2 e a seqüência representa um afastamento dos clientes em relação ao servidores.

	Servidores		*Sites* clientes/demandas	
Identificador	Sites1	Sites2	N_{Sites1}	N_{Sites2}
1	X		1000	-
2	X		1000	1000
3	X		-	1000
4		X	-	1000
5		X	1000	1000
6		X	1000	-
7	X	X	1000	1000
8	X		100	1000
9	X		1000	100
10		X	100	1000
11		X	1000	100
12	X	X	100	1000
13	X	X	1000	100

Tabela 6.2: Tipos de teste realizados em cada uma das configurações

teste 7 Representa a situação em que os servidores e os clientes estão espalhados de forma homogênea.

Para entender como a variação da heterogeneidade de demanda atua sobre o ganho do protocolo escalável sobre o convencional, usamos as seguintes seqüências de testes:

testes 8, 2 e 9 Servidores são os Sites1 e a seqüência representa a demanda se aproximando dos servidores.

testes 11, 5 e 10 Servidores são os Sites2 e a seqüência representa a demanda se aproximando dos servidores.

testes 12, 7 e 13 Todos são servidores e a seqüência representa a demanda se aproximando da região mais central (Sites1).

6.1.7 Dados das configurações testadas

Foram testados os seguintes tipos de configuração:

- distância geográfica, rede global
- distância geográfica, rede local
- distância observada, rede global
- distância observada, rede local
- distância global, rede global
- distância global sobre as redes sintéticas

Tipo de configuração	Total de configurações	Variação do número de *sites*	Variação do diâmetro
distância geográfica, rede global	30	2-30	250-6300 (km)
distância geográfica, redes locais	30	2-30	250-6300 (km)
distância observada, rede global	49	2-32	1-14 (saltos)
distância observada, redes locais	49	2-32	1-14 (saltos)
distância global, rede global	49	2-40	1-9 (saltos)
redes sintéticas	327	2-37	1-7 (saltos)

Tabela 6.3: Dados das configurações usadas

A tabela 6.3 apresenta alguns dados a respeito das configurações geradas pelo processo de aglomeração descrito na seção 6.1.5.1, para os vários tipos de configurações gerados. O diâmetro de uma configuração corresponde à maior distância entre dois *sites* da dessa configuração. A faixa de variação dos diâmetros para cada tipo de configuração é mostrada na última coluna. Cada uma das configurações passa por 13 tipos de teste diferentes. Em cada tipo de teste são geradas várias florestas, uma para cada valor de m para cada um dos 4 protocolos, o convencional e os escaláveis.

6.2 Resultados

Nesta seção apresentamos os resultados dos testes realizados, em termos do ganho em banda média de rede de cada protocolo escalável sobre o protocolo convencional. Foram criadas entre 2047 e 3180 florestas por protocolo para cada tipo de configuração real e 21795 florestas por protocolo para as configurações sintéticas. Como os resultados foram qualitativamente semelhantes para todos os tipos de configuração, mostraremos aqui os resultados apenas para (distância global, rede global), (distância geográfica, rede local) e redes sintéticas.

Primeiro, damos uma visão geral dos resultados, comparando os protocolos através da média de resultados de várias configurações. Em seguida, analisamos o impacto de cada um dos parâmetros definidos, focando nos protocolos, configurações e parâmetros de maior impacto. Por fim, fazemos um estudo de casos de algumas das configurações que apresentaram maiores ganhos.

6.2.1 Comparação entre os protocolos: visão geral

Uma comparação entre os protocolos de roteamento pode ser feita analisando-se seus ganhos médios. O ganho médio para determinado tipo de teste executado sobre determinada configuração é a média dos ganhos obtidos para cada valor de m. A figura 6.4 mostra a distribuição acumulada dos ganhos médios dos protocolos de roteamento SP, MCO e MCI sobre o protocolo convencional, considerando todas as configurações e todos os tipos de teste executados sobre elas.

Pode-se observar que nas configurações reais esses três protocolos apresentam algum ganho em pelo menos 50% dos casos. Em menos de 5% das situações esses protocolos perdem para o protocolo convencional. Os ganhos médios chegam a cerca de 30% e as perdas, quando ocorrem, são inferiores a 10% em quase todos os casos. Perdas podem ocorrer devido ao caráter heurístico, guloso, dos protocolos avaliados. Algumas situações de perda são analisadas a fundo na seção 6.2.3.

Note que os protocolos MCI e MCO apresentam maior ganho que o SP. Além disso, ainda na figura 6.4, as curvas dos protocolos MCI e MCO estão praticamente sobrepostas, dando indícios de que na maioria das situações esses dois protocolos apresentam ganhos bem parecidos. Isso é confirmado pelo gráfico 6.5, que apresenta a distribuição acumulada das diferenças entre o ganho do protocolo MCI e o ganho do protocolo MCO em cada situação: em mais de 70% dos casos os protocolos MCI e MCO apresentam exatamente o mesmo ganho. Nas demais situações, a diferença entre o ganho de um e de outro é quase sempre menor que 10%. Quando há diferença, na metade das vezes o protocolo MCI apresenta maior ganho e na outra metade o MCO apresenta maior ganho.

Sendo assim, optamos por focar nossa análise no protocolo MCO, já que ele possui menor complexidade computacional. Por isso, daqui em diante, toda vez que nos referirmos a "o protocolo escalável", estaremos falando do protocolo MCO.

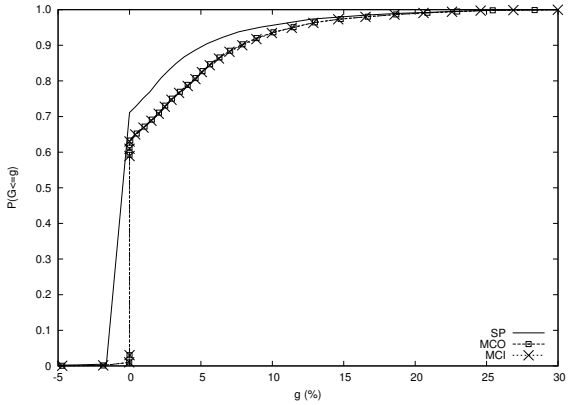
6.2.2 Efeito dos parâmetros variados

A figura 6.6 mostra a distribuição acumulada dos ganhos máximos, para cada tipo de teste, das topologias sintéticas. O ganho máximo para determinado tipo de teste executado sobre determinada configuração é o maior dentre os ganhos obtidos para cada valor de m. Veja que os tipos de teste que levaram a maiores ganhos foram aqueles em que as réplicas podem ser colocadas em quaisquer dos *sites* dos conjuntos Sites1 e Sites2, ou seja, os testes 7, 12 e 13.

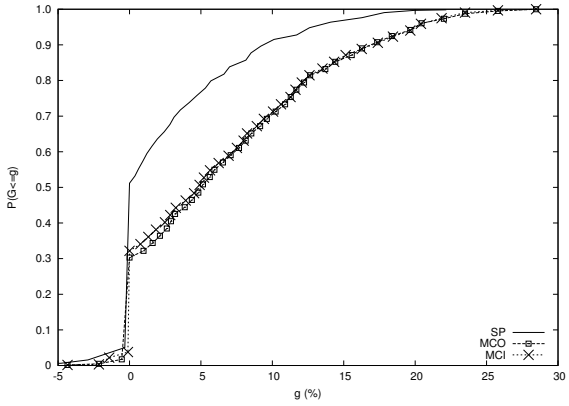
As figuras 6.7, 6.8 e 6.9 mostram o impacto isolado de cada um dos parâmetros, m, n e dispersão. Isso é feito plotando o ganho em banda média de rede obtido pelo protocolo MCO sobre o protocolo convencional em função de cada um desses parâmetros, mantendo fixos os demais parâmetros. Tais valores fixos são apresentados nas legendas. Nesses gráficos, m é expresso como uma porcentagem do número de *sites* clientes, ao invés do valor absoluto de réplicas utilizado. Por simplicidade foram exibidas somente as curvas correspondentes aos tipos de teste 7, 12 e 13, que em geral levam a maiores ganhos.

Para melhor entender o significado desses gráficos, tomemos como exemplo a avaliação do parâmetro m, mostrada na figura 6.7(a). Neste caso, tomando o ponto (42, 9), o valor 9 corresponde à média entre os valores de ganho nas amostras sintéticas nas situações em que m é igual a 42%, n é igual a 19, dispersão é igual a 6 e o tipo de teste é 13. Os valores desses últimos 3 parâmetros são apresentados nas legendas do gráfico e da curva em questão.

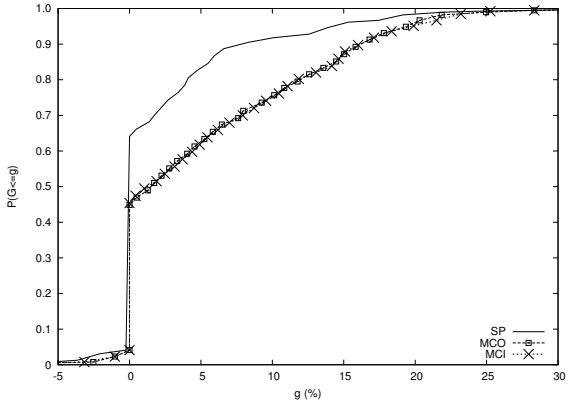
Uma primeira observação que pode ser feita a respeito das três figuras, em geral, é que há situações de ganho razoavelmente alto, da ordem de 20%-30%, tanto para topologias reais quanto para sintéticas. Há também situações de perda, por motivos que serão apresentados seção 6.2.3.



(a) Configurações sintéticas



(b) Distância global, rede global



(c) Distância geográfica, rede local

Figura 6.4: Ganho médio G dos protocolos escaláveis sobre o protocolo convencional

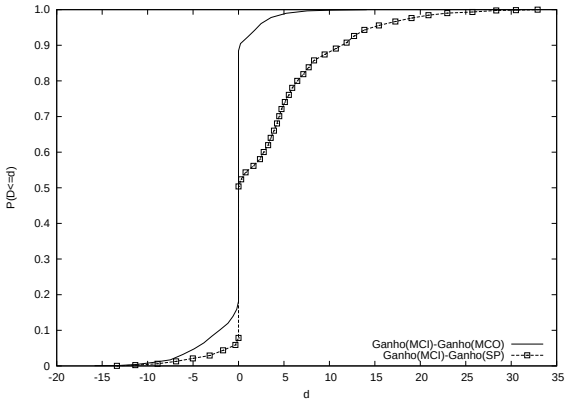


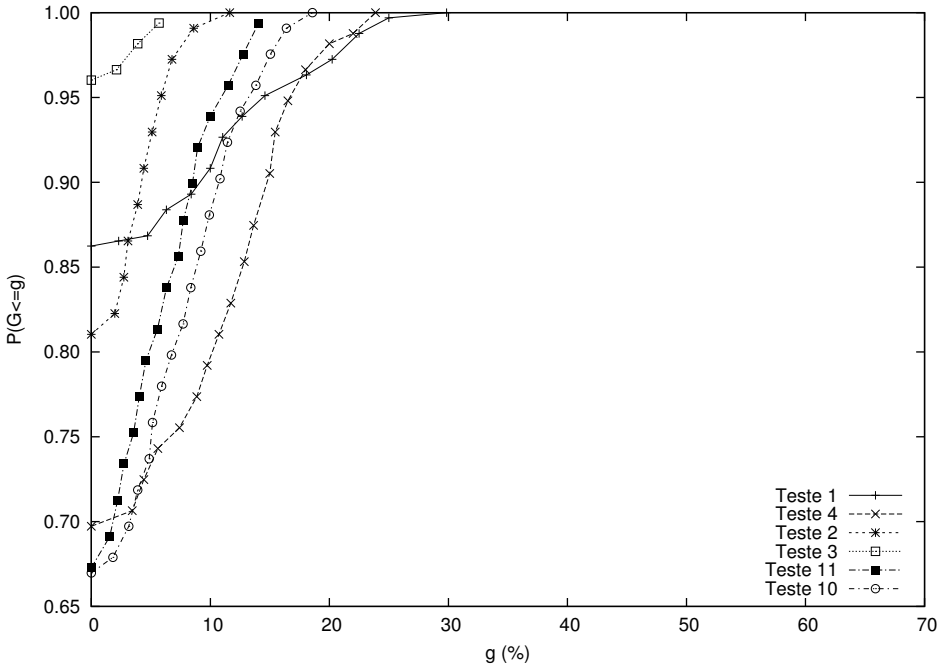
Figura 6.5: Distribuição acumulada das diferenças D entre os ganhos obtidos pelos protocolos escaláveis

Pode-se observar também uma grande variabilidade nas curvas, motivo pelo qual omitimos dos gráficos os intervalos de confiança, que muitas vezes foram grandes. Essa variabilidade nos impede tirar conclusões a respeito do impacto dos parâmetros n e dispersão sobre o ganho médio. Entretanto, podemos notar, pela figura 6.7, que maiores ganhos médios ocorrem quando entre 30% e 80% dos *sites* são usados como réplicas. Esse resultado coincide com os de Almeida et al [4].

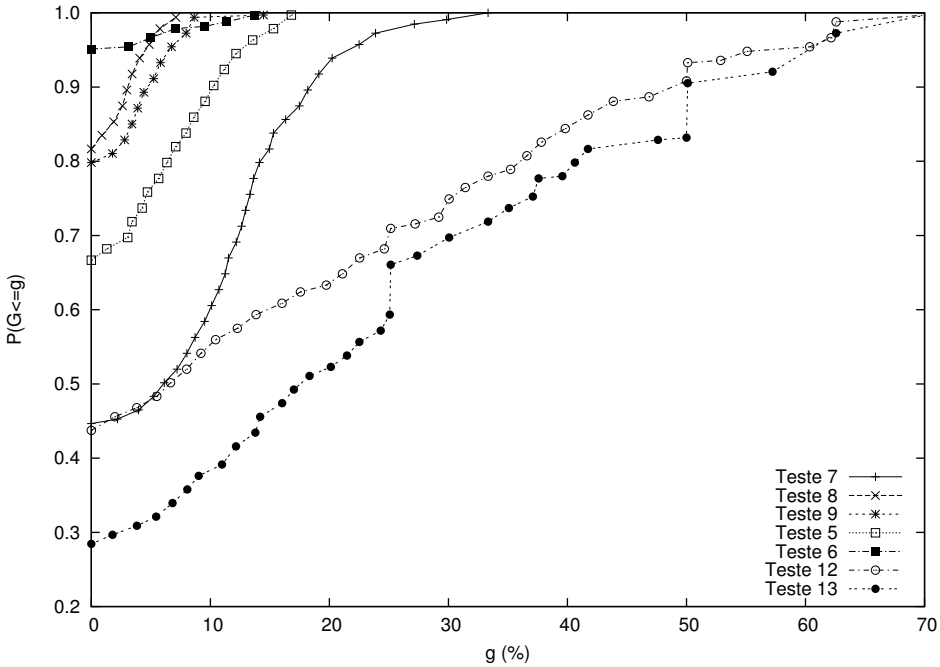
A alta variabilidade ocorre porque, ao se fixarem 3 dos 4 parâmetros (n, m, dispersão, tipo de teste), em geral há um número pequeno de amostras que se encaixam nos valores fixados.

Outro motivo para a alta variabilidade é que existe um parâmetro que foi sempre deixado livre nas nossas análises: o **conjunto de *sites* participantes** em si. Não o número de *sites* participantes, mas sim **quais** os *sites* compõem os conjuntos Sites1 e Sites2 e a topologia de rede encontrada entre esse *sites*. Especulamos que tal parâmetro tenha muito mais influência sobre o ganho do que os parâmetros propositalmente variados (m, n, dispersão e tipo de teste). Por esse motivo os resultados foram pouco conclusivos com relação a esses parâmetros. O gráfico 6.10 ilustra bem o fato de os parâmetros escolhidos serem insuficientes para caracterizar as configurações utilizadas de forma a relacioná-las com o ganho obtido. Esse gráfico mostra 3 configurações com os mesmos valores para todos os parâmetros definidos, variando somente o número de réplicas. Note que, fixando o número de réplicas m, em muitos casos existe uma diferença considerável entre os ganhos das diferentes configurações. Por exemplo, para $m = 9$, existe um ganho de 12% na configuração B, enquanto na configuração A existe uma leve perda.

Na seção 6.2.3, fixamos também o parâmetro "conjunto de *sites* participantes". Fixar esse parâmetro faz com que a análise passe a ser caso a caso, em vez de uma média de várias amostras. Em outras palavras, comparamos florestas geradas pelo protocolo convencional com florestas geradas pelo protocolo escalável. Dada a impossibilidade (e falta de necessidade) de analisar cada uma das situações individualmente, analisamos apenas casos em que houve maiores ganhos ou perdas.

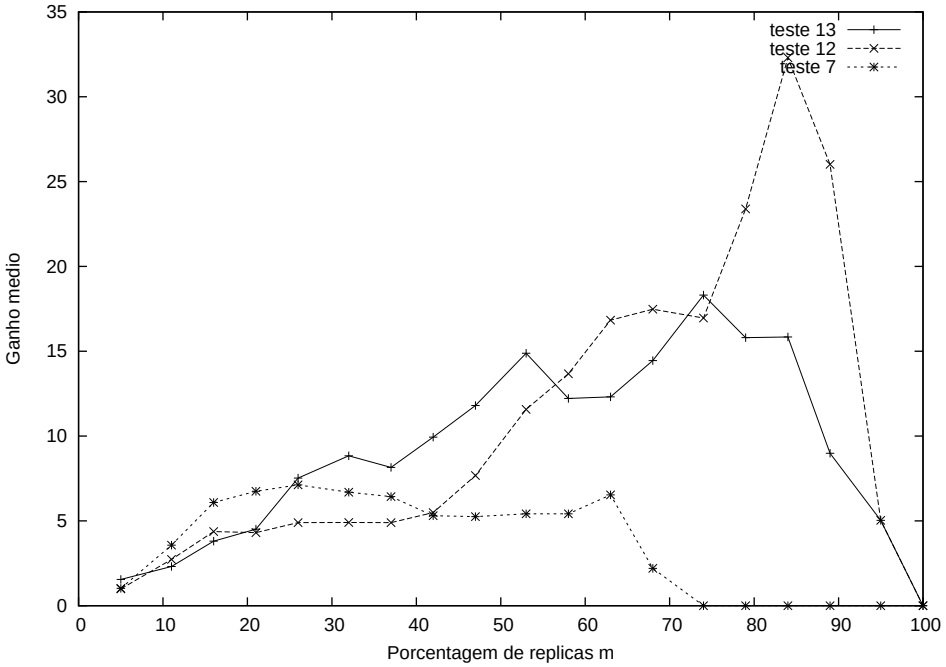


(a) Testes com menores ganhos

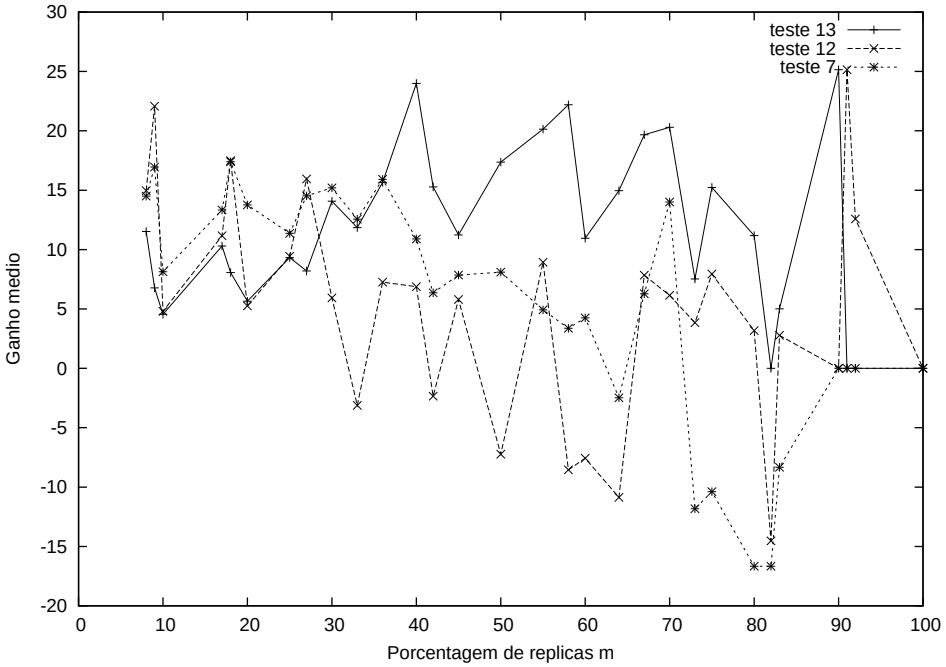


(b) Testes com maiores ganhos

Figura 6.6: Distribuição acumulada dos ganhos máximos G dos protocolos escaláveis sobre o convencional, para cada tipo de teste (Configurações sintéticas)

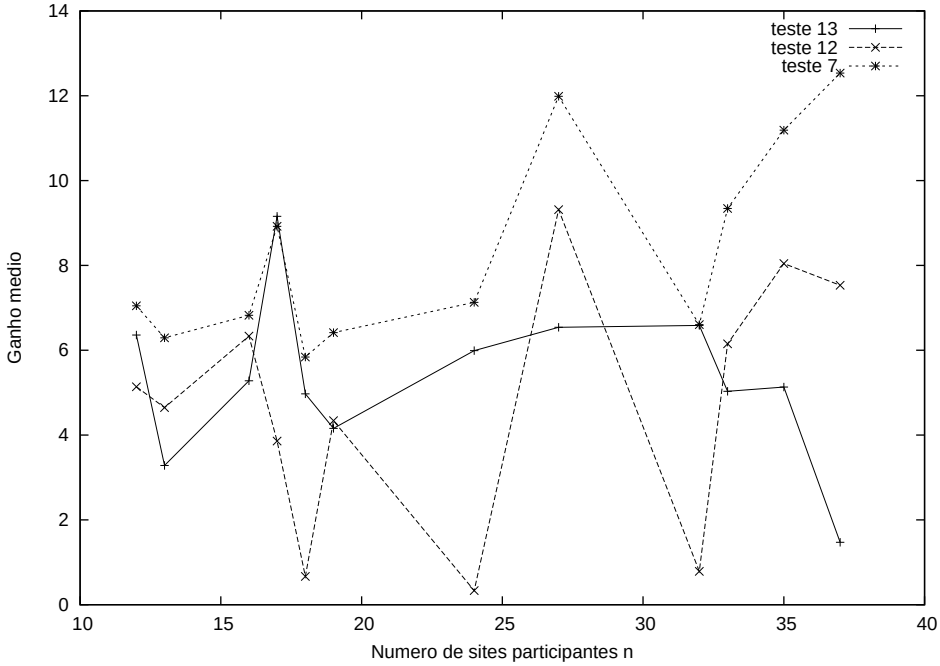


(a) Sintética, n=19, dispersão=6

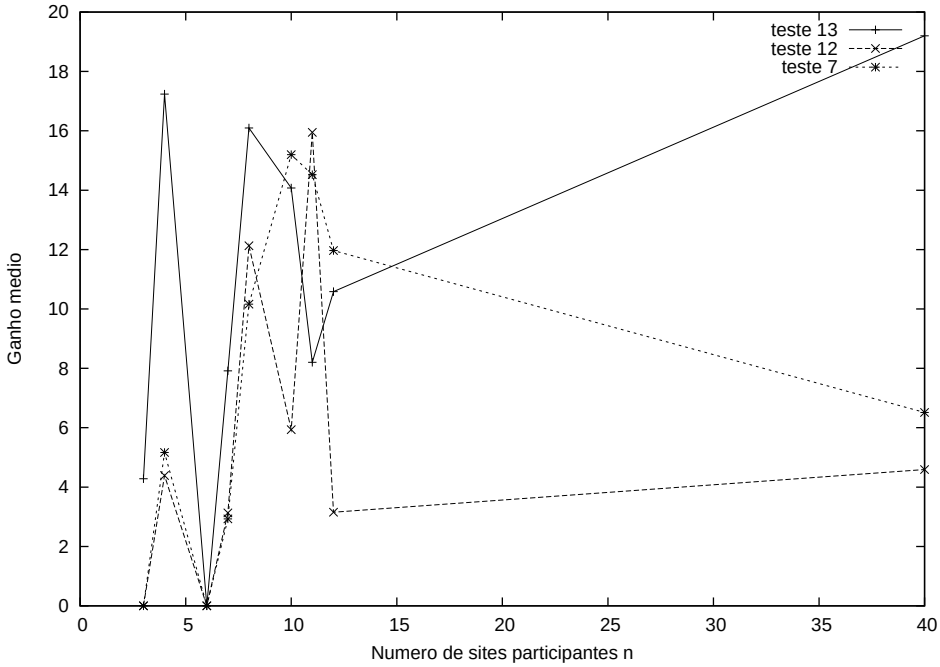


(b) Real, n=11, dispersão=7

Figura 6.7: Variação da porcentagem de réplicas m (demais parâmetros fixos)

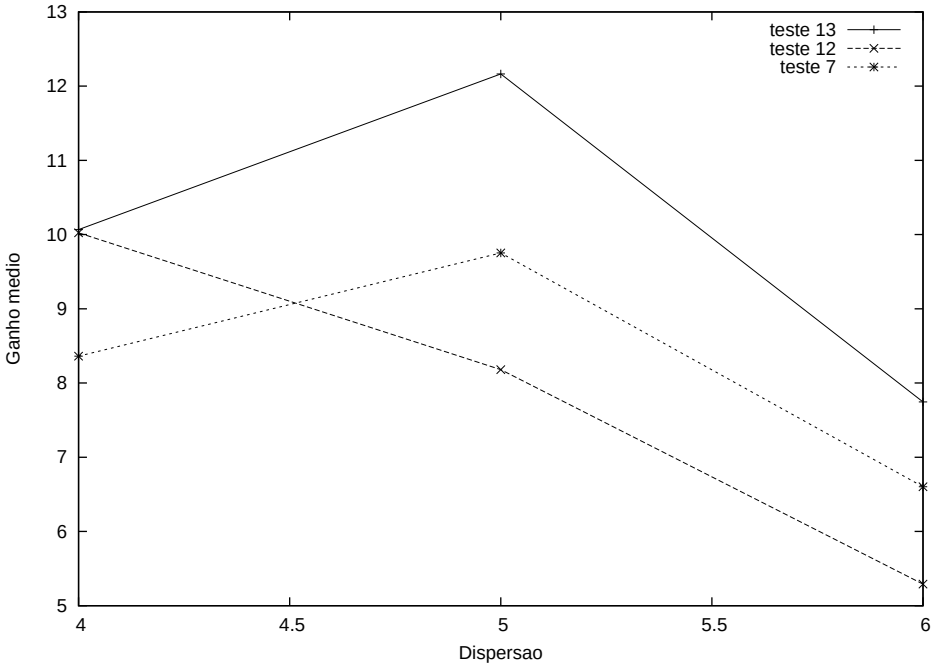


(a) Sintética, m=20%, dispersão=6

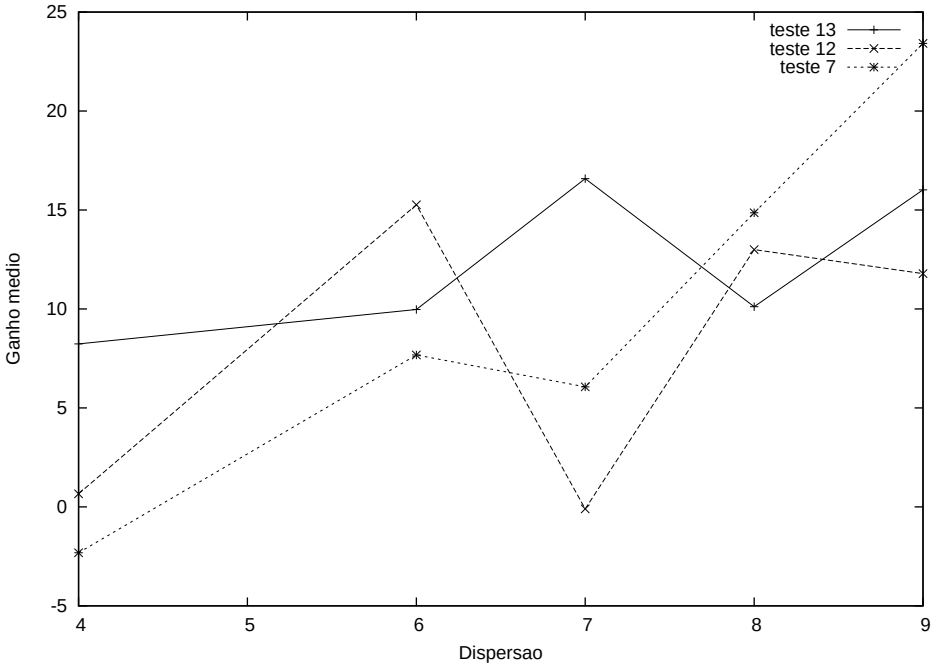


(b) Real, m=30%, dispersão=7

Figura 6.8: Variação do número de *sites* participantes n (demais parâmetros fixos)



(a) Sintética, m=30%, n=17



(b) Real, m=50%, n=12

Figura 6.9: Variação da dispersão (demais parâmetros fixos)

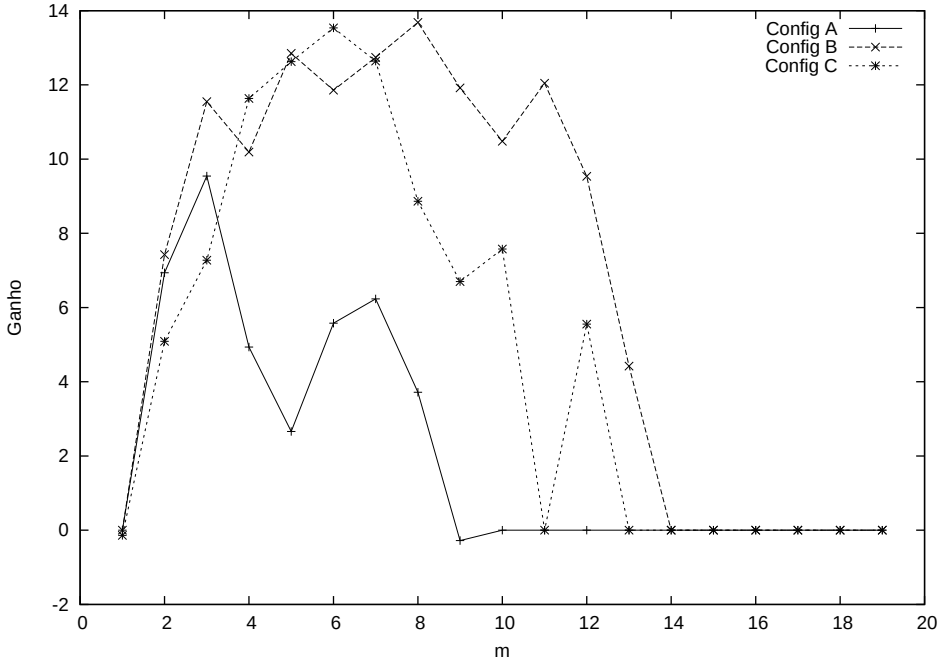


Figura 6.10: Três configurações diferentes, A, B e C. Teste=7, dispersão=6, n=19 (sendo $|Sites1|$=11 e $|Sites2|$=8)

Grupo	Rótulo	*Site*	Localização
Sites1	35	traceroute.hinet.net	Japão
Sites1	108	www.harenet.ad.jp	Japão
Sites1	38	www.kawaijibika.gr.jp	Japão
Sites2	157	traceroute.pacific.net.hk	Hong Kong
Sites2	217	traceroute.hgc.com.hk	Hong Kong
Sites2	179	202.71.136.244	Índia
Sites2	111	speedtest.indo.net.id	Indonésia
Sites2	256	140.111.1.22	Taiwan

Tabela 6.4: Ásia - Sites1 e Sites2

6.2.3 Estudo de casos

Foram analisadas diversas florestas geradas pelos protocolos convencional e escalável (MCO), especialmente em casos onde houve ganho ou perda consideráveis. Esta seção apresenta um subconjunto dos casos analisados suficiente para ilustrar cada uma das conclusões obtidas. Outros exemplos são encontrados no apêndice A. As configurações analisadas foram geradas através de aglomeração por distância geográfica e correspondem a continentes.

As florestas mostradas nas figuras desta seção apresentam a seguinte notação:

- *Sites*, sejam clientes ou servidores, são rotulados com um número inteiro. Vértices sem rótulos correspondem a roteadores.
- Todas as arestas não rotuladas têm peso igual a 1, isto é, correspondem a 1 *link* entre *sites* ou roteadores na rede real. Arestas rotuladas apresentam no rótulo o número de *links* consecutivos a que elas correspondem.
- *Sites* clientes possuem uma elipse em volta do rótulo, sendo que elipses cinzas indicam que a demanda do *site* é 1000 e elipses brancas indicam que a demanda do *site* é 100.
- *Sites* servidores estão localizados na primeira linha horizontal e possuem um "(S)" precedendo o rótulo.

6.2.3.1 Ásia

Nesta seção analisamos alguns dos testes realizados usando como configuração uma rede local composta por todos os *sites* da Ásia. O conjunto Sites1 corresponde aos 3 *sites* do Japão e o conjunto Sites2 aos demais, conforme a tabela 6.4

O gráfico da figura 6.11 mostra os tipos de teste selecionados para análise.

No teste 11, os *sites* japoneses possuem demanda $N_j = 1000$ e os demais possuem demanda $N_o = 100$. As réplicas podem ser alocadas apenas a *sites* não-japoneses. Analisamos os pontos $m = 4$ e $m = 5$ (figura 6.12) para entender de onde vem o ganho de cerca de 20% do protocolo escalável sobre o convencional no ponto $m = 5$. Nota-se que em todas as situações o servidor 157, de Hong Kong, serviu os *sites* japoneses. Para cada valor de m, ambas as árvores

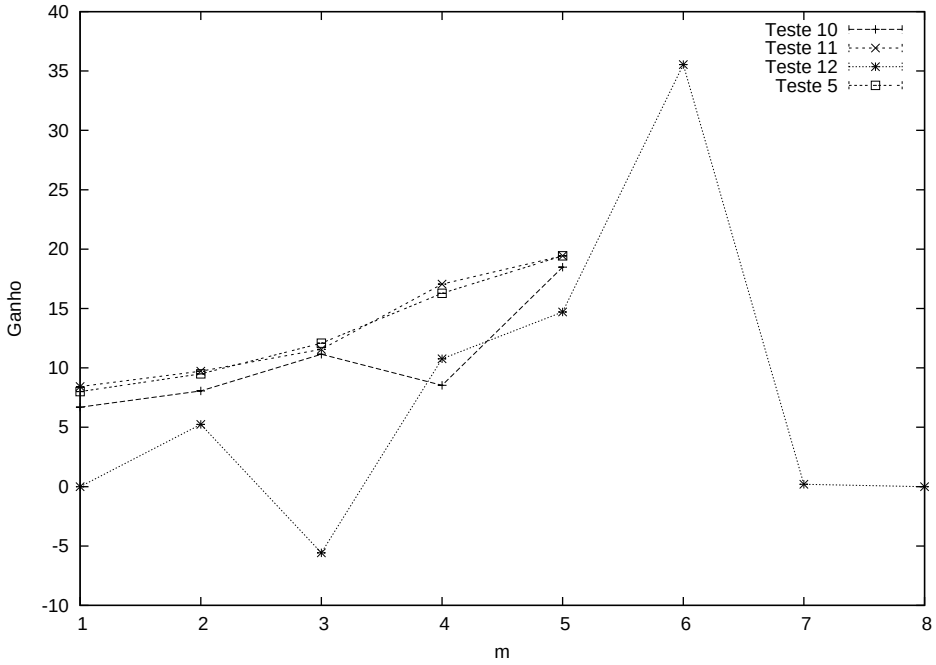
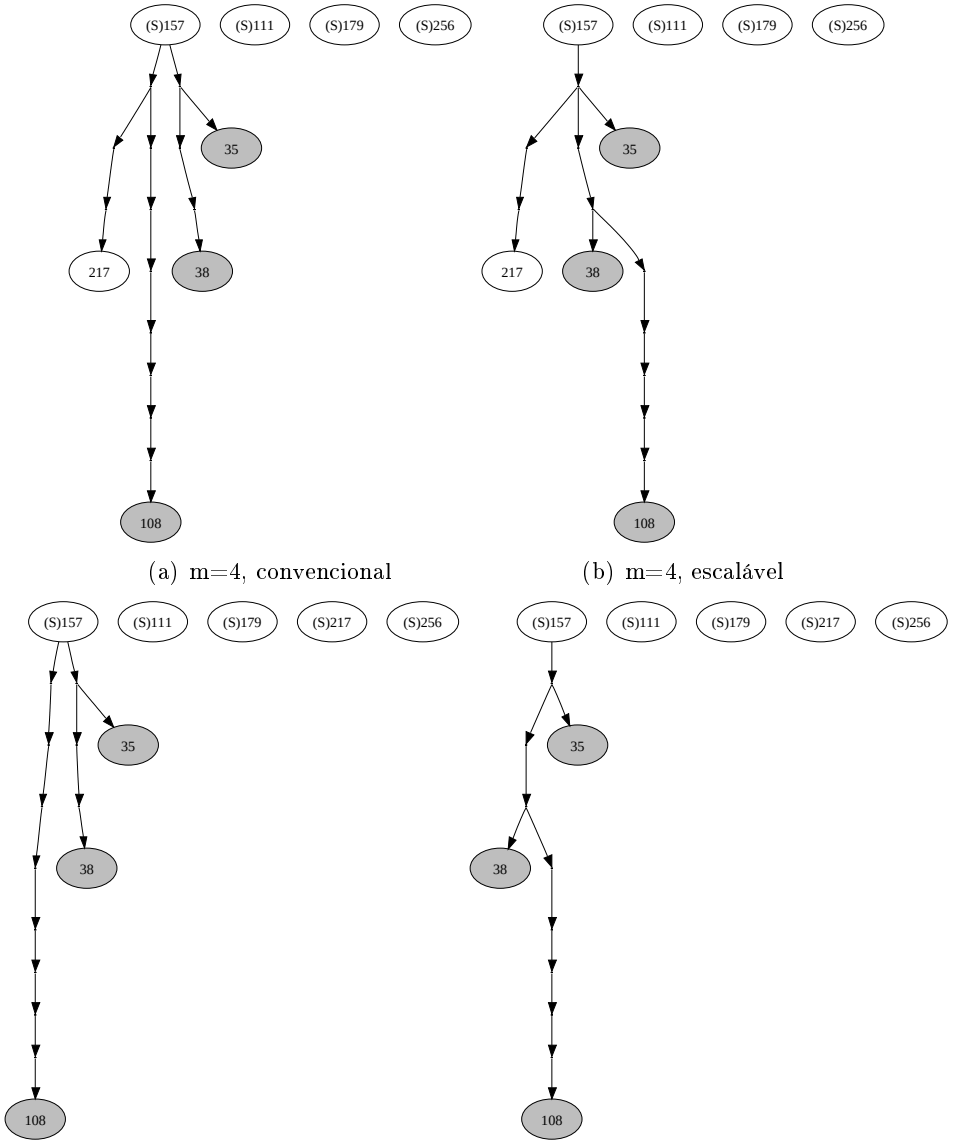


Figura 6.11: Ásia - Ganho do MCO sobre o convencional para tipos de teste selecionados

apresentam a mesma distância dos servidores até cada um dos clientes. Ou seja, a árvore criada pelo protocolo escalável também é uma árvore dos caminhos mais curtos, porém mais compartilhada e portanto de custo menor. Essa é uma limitação do protocolo de roteamento SP, que cria a árvore dos caminhos mais curtos a partir dos caminhos fornecidos pelo algoritmo de Dijkstra. Se houver outras árvores equivalentes, elas podem não ser encontradas por esse protocolo. O protocolo escalável, por outro lado, encontra essas árvores equivalentes de menor custo, já que a função de otimização é voltada para compartilhamento de fluxos, testando diferentes alternativas (e por isso mesmo apresenta complexidade computacional maior).

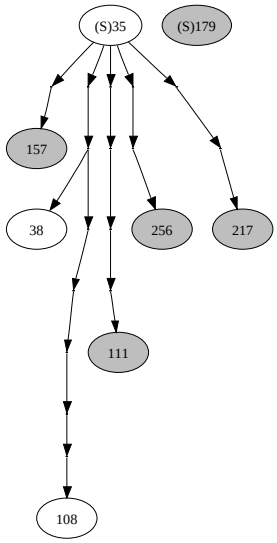
A figura 6.13 mostra as árvores para o teste 12, com m variando entre 2 e 5, faixa na qual o protocolo escalável começou obtendo ganho de 5%, em seguida houve uma perda de 5% e depois dois novos ganhos cada vez maiores, de 10% e 15%. Nesse teste os *sites* japoneses possuem demanda $N_j = 100$ e os demais *sites* possuem demanda $N_o = 1000$. As réplicas podem ser alocadas em quaisquer dos *sites*. Ambos os protocolos optam por criar uma grande árvore compartilhada saindo do nó 35 (Japão), em todas as situações. A diferença, para $m = 2$, é que o protocolo convencional escolhe como segunda réplica um *site* de demanda alta, visando minimizar o custo para *unicast* e o protocolo escalável escolhe como segunda réplica o *site* 108, distante e pouco compartilhado, minimizando o custo para compartilhamento de fluxos. Com isso o protocolo escalável mantém dois *sites* (179 e 111) de demanda alta compartilhando dois *links*, além de eliminar o *site* distante pouco compartilhado. Daí vem o ganho do protocolo escalável nesse ponto. No ponto seguinte ($m = 3$), porém, o protocolo escalável escolhe um desses dois *sites* de demanda alta para ser réplica e assim a árvore fica sem compartilhamento. Já o convencional, que tinha escolhido um deles para ser servidor no passo $m = 2$, escolhe o



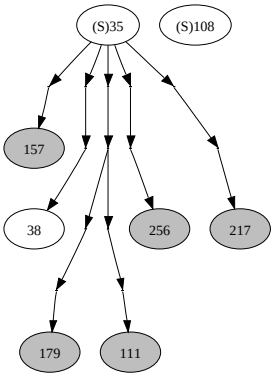
(a) m=4, convencional (b) m=4, escalável

(c) m=5, convencional (d) m=5, escalável

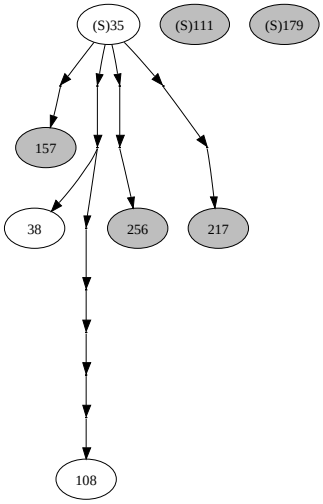
Figura 6.12: Ásia - Teste 11



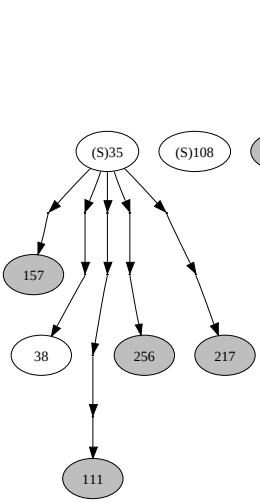
(a) m=2, convencional



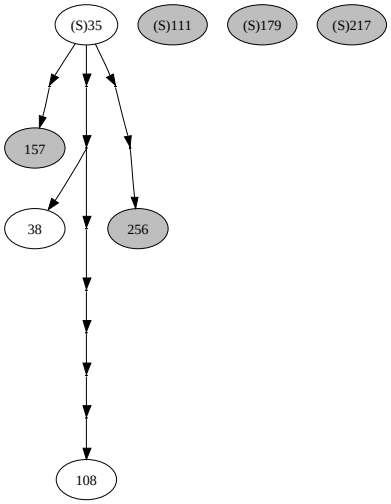
(b) m=2, escalável



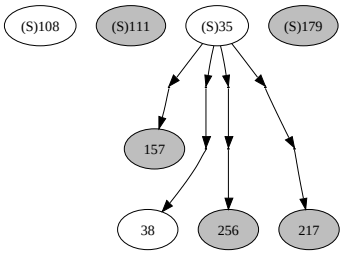
(c) m=3, convencional



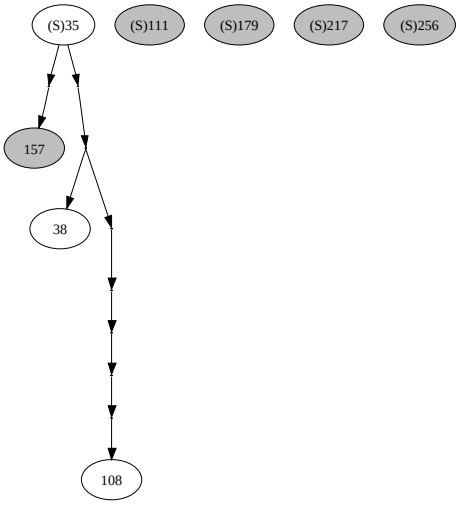
(d) m=3, escalável



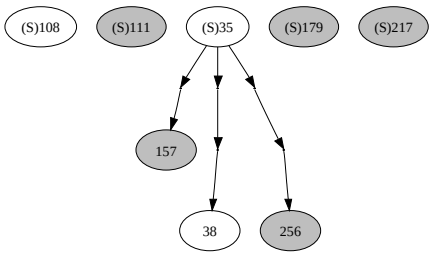
(e) m=4, convencional



(f) m=4, escalável



(g) m=5, convencional



(h) m=5, escalável

Figura 6.13: Ásia - Teste 12

segundo para ser servidor no passo $m = 3$, ficando com uma árvore de custo muito menor. Em $m = 4$, o escalável finalmente coloca o uma réplica no segundo dos dois *sites* de demanda alta mencionado, enquanto o convencional novamente escolhe um *site* de alta demanda para ser a quarta réplica. Nesse ponto, o *site* 108, distante e pouco compartilhado, que o protocolo convencional manteve na árvore, passa a fazer diferença e assim o protocolo escalável volta a ganhar. Ainda em $m = 5$ o protocolo convencional mantém esse *site* 108 e o ganho do escalável passa a ser maior ainda.

Desse teste podemos notar que, para compartilhamento de fluxos, ao contrário de *unicast*, muitas vezes é melhor colocar réplicas em *sites* de baixa demanda que estejam distantes e sem compartilhamento, em vez de *sites* de alta demanda. Isso porque, enquanto tanto a demanda quanto o número de saltos impactam linearmente no custo para *unicast*, no caso de compartilhamento de fluxos a demanda impacta apenas logaritmicamente. Isso faz com que seja melhor dar prioridade ao aumento do grau de compartilhamento em vez de tentar reduzir a distância.

Outro aspecto interessante, notado na transição de $m = 2$ para $m = 3$ do protocolo escalável, é que o fato de a alocação de réplicas funcionar de forma incremental, apesar de ser simples, pode ser ruim. Uma decisão boa para duas réplicas pode ser ruim quando se tenta manter essas duas réplicas e adicionar uma terceira. Entretanto, esse caráter incremental simplifica bastante o protocolo de localização, pois o custo de revisitar uma decisão anterior é muito alto (complexidade exponencial).

Grupo	Rótulo	*Site*	Localização
Sites1	789	www.ntua.gr	Grécia
Sites1	107	www.atcom.net.pl	Polônia
Sites1	854	cgi.ipartners.pl	Polônia
Sites1	831	lg.transtk.ru	Rússia
Sites1	665	ulda.inasan.rssi.ru	Rússia
Sites1	990	nic.dn.ua	Ucrânia
Sites2	770	sites.inka.de	Alemanha
Sites2	843	lg.inetbone.net	Alemanha
Sites2	671	ipet.as12769.net	Espanha
Sites2	809	lg.hostingfrance.com	França
Sites2	760	www.azuria.net	França
Sites2	111	www.netultra.net	França
Sites2	594	www.rhnet.is	Islândia
Sites2	834	carmen.cselt.it	Itália
Sites2	938	www.cnaf.infn.it	Itália
Sites2	358	www.mclink.it	Itália
Sites2	192	glass.cprm.net	Portugal
Sites2	395	www.nildram.net	Reino Unido
Sites2	331	ppewww.ph.gla.ac.uk	Reino Unido
Sites2	546	www.mailbox.net.uk	Reino Unido

Tabela 6.5: Europa - Sites1 e Sites2

6.2.3.2 Europa

Nesta seção analisamos alguns dos testes realizados usando como configuração uma rede local composta por todos os *sites* da Europa. O conjunto Sites1 corresponde aos *sites* do leste e o conjunto Sites2 aos demais, conforme a tabela 6.5.

O gráfico da figura 6.14 mostra os tipos de teste selecionados para análise, nesta seção e no apêndice A.

O teste 1 para a Europa consiste em todos os *sites* do leste possuindo demanda $N = 1000$ e todos eles podendo ser réplicas. Os *sites* do oeste não são usados. Veja, pela figura 6.15, que em $m = 4$ as árvores criadas pelos dois protocolos possuiriam o mesmo custo (6000) se fossem usadas para *unicast*. No entanto a árvore criada pelo protocolo convencional é mais barata para fluxos compartilhados, pois possui 1 *link* compartilhado entre os dois clientes. Logo, duas árvores podem ter o mesmo custo para *unicast* mas uma delas ser melhor para fluxos compartilhados. Isso ocorre também com árvores (dos caminhos mais curtos) equivalentes, como ocorreu no teste 11 da Ásia.

Em $m = 5$, observa-se no gráfico 6.14 um ganho de 50% em banda média de rede. Ao analisarmos, vemos que a "má" escolha inicial (em $m = 4$) do protocolo convencional leva a uma árvore ainda pior para $m = 5$. Veja que, neste ponto, mesmo para *unicast* a árvore do protocolo escalável seria melhor. Daí se vê que o fato de a escolha de réplicas ser incremental pode ser ruim, tanto para o protocolo convencional, neste caso, quanto para o protocolo

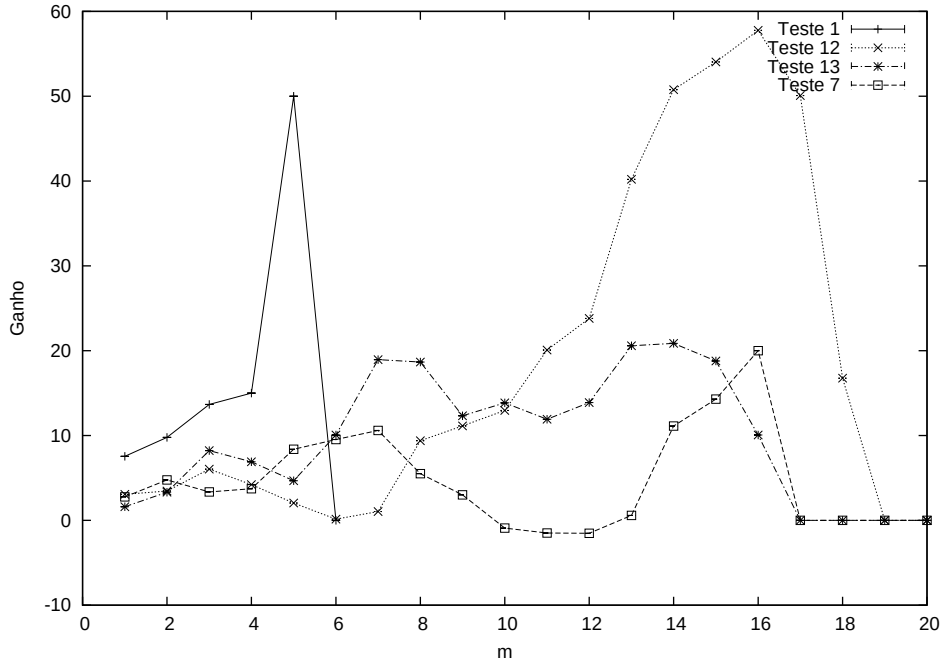
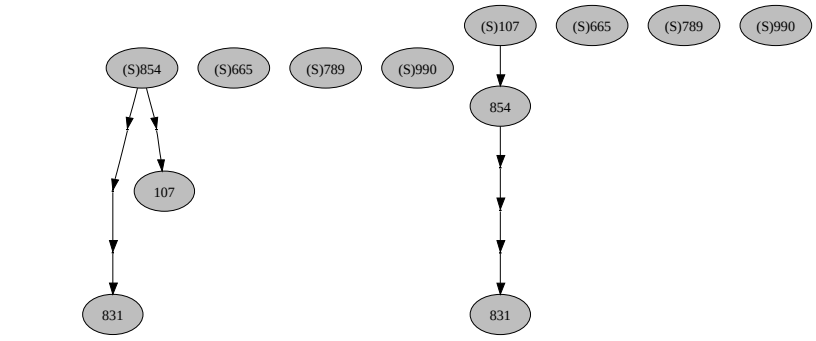
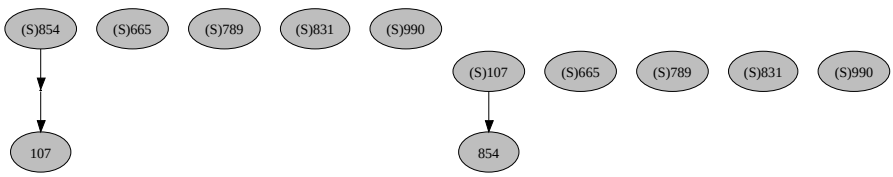


Figura 6.14: Europa - Ganho do MCO sobre o protocolo convencional para os tipos de teste selecionados

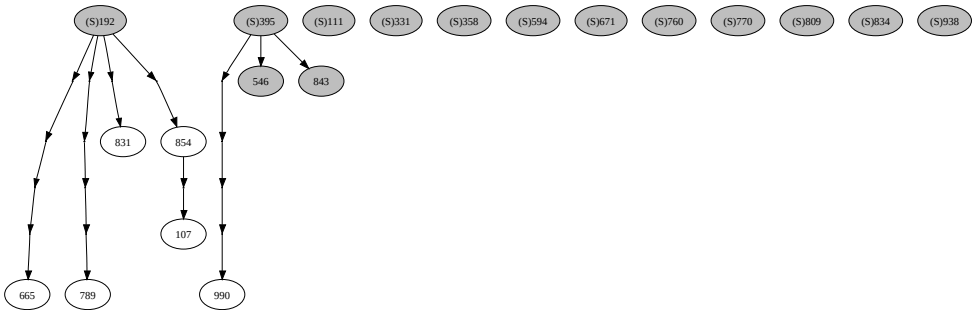


(a) m=4, convencional (b) m=4, escalável

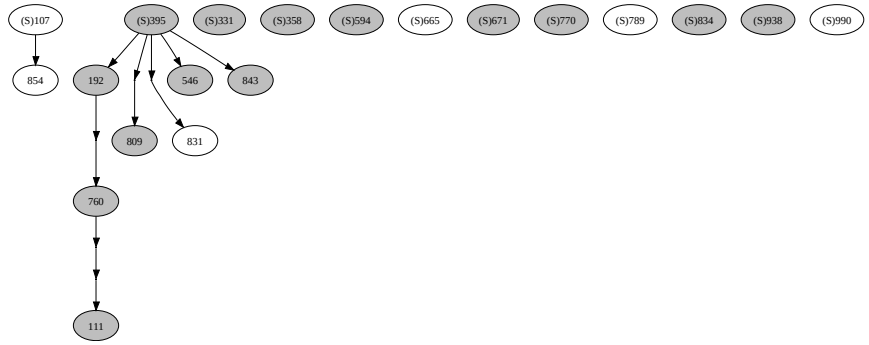


(c) m=5, convencional (d) m=5, escalável

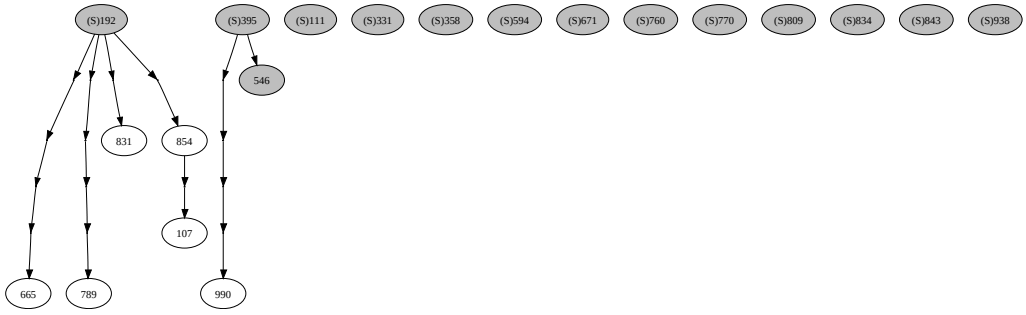
Figura 6.15: Europa - Teste 1



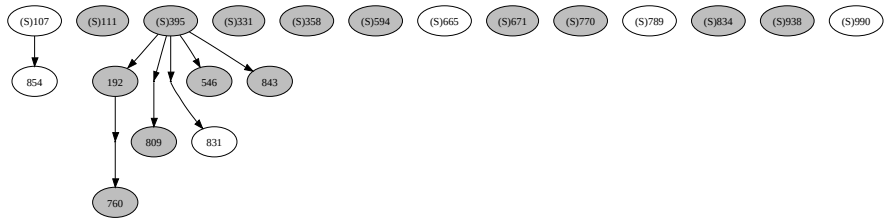
(a) m=12, convencional



(b) m=12, escalável



(c) m=13, convencional

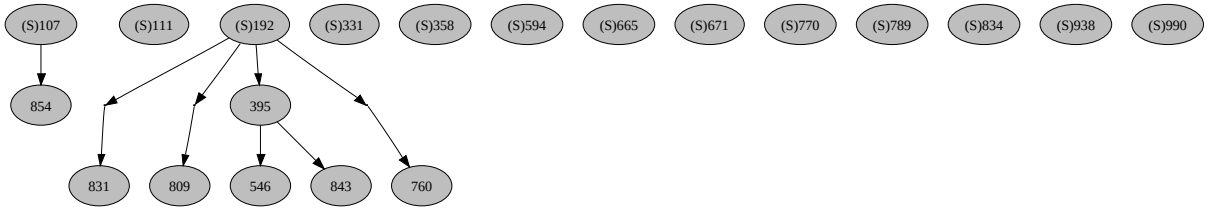


(d) m=13, escalável

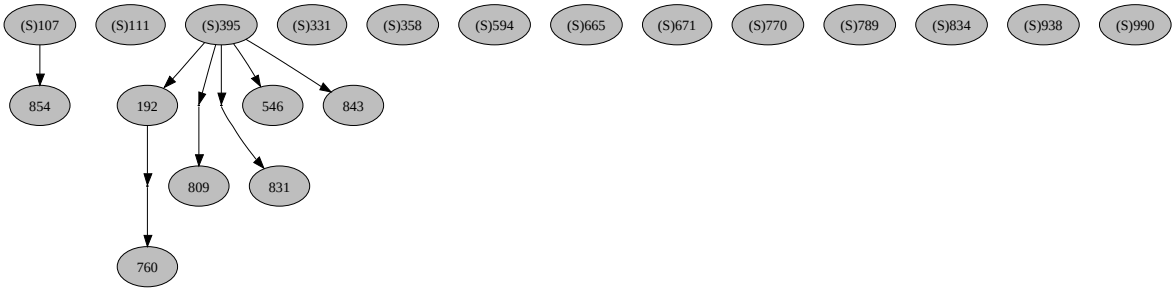
Figura 6.16: Europa - Teste 12

escalável, no caso do teste 12 da Ásia.

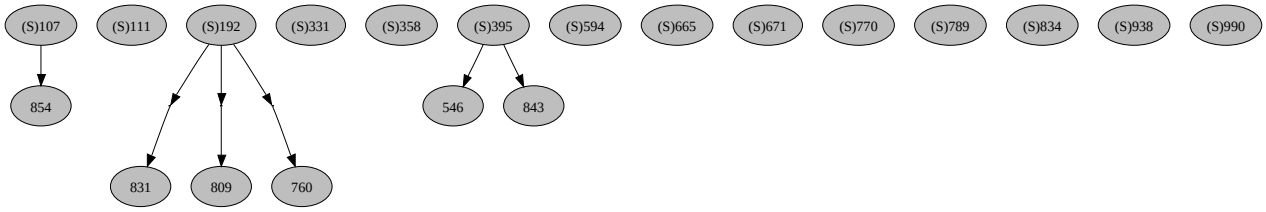
No teste 12, todos os *sites* podem ser réplicas, sendo que os *sites* do leste possuem demanda $N_e = 100$ e os do oeste possuem demanda $N_o = 1000$. Na transição de $m = 12$ para $m = 13$ (figura 6.16), há um aumento do ganho do protocolo escalável sobre o convencional, de pouco mais de 20% para cerca de 40%. O que ocorre, como muitas vezes, é o protocolo convencional alocar réplicas em *sites* de demanda mais alta, neste caso o *site* 843 e o escalável



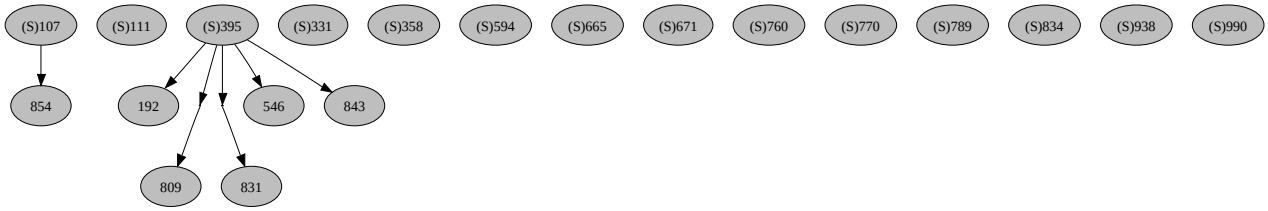
(a) m=13, convencional



(b) m=13, escalável



(c) m=14, convencional



(d) m=14, escalável

Figura 6.17: Europa - Teste 7

preferir colocá-as em *sites* distantes que apresentam pouco compartilhamento, neste caso o *site* 111, que apresentava 3 saltos sem compartilhamento entre ele e seu pai, o *site* 760. Com essa escolha, o protocolo convencional economizou em 1 *link*, de custo $ln(1001)$, enquanto o protocolo escalável economizou em 3 *links* com esse mesmo custo.

Nesse mesmo teste, no ponto $m = 16$, o protocolo escalável atinge um ganho de quase 60%. Nessa situação, a floresta do protocolo escalável consiste em 4 *links* não compartilhados, sendo que 3 deles servem *sites* de demanda 1000 e o outro serve um *site* de demanda 100. Neste caso, a banda média total é $3 \times ln(1001) + ln(101) = 25,34$. Já o protocolo convencional, por sua tendência a alocar réplicas nos *sites* de maior demanda, possui uma floresta que consiste em 4 caminhos não compartilhados levando a *sites* de demanda 100. Os comprimentos desses caminhos são 5, 5, 2 e 1, portanto a banda média total é $13 \times ln(101) = 59,99$.

O teste 7 é totalmente homogêneo: todos os *sites* da Europa possuem demanda $N = 1000$

e as réplicas podem ser alocadas em quaisquer deles. Veja pela figura 6.17 que no ponto 13, apesar de ambas as florestas possuírem mesmo custo, a floresta do protocolo escalável possui uma árvore mais alta. Na transição para $m = 14$, o protocolo convencional remove a subárvore cuja raiz era o nó 395 e de demanda total $N = 3000$. Essa subárvore era servida por um *link* de custo $ln(3001)$, que para *unicast* custaria 3000. Já o protocolo escalável poda sua árvore mais alta, alocando uma réplica no nó 760, removendo portanto 2 *links* de custo $ln(1001)$, que ligavam o nó 192 ao 760. O *link* que serve o nó 192 passa a servir uma subárvore composta apenas por esse nó, e portanto seu custo passa de $ln(2001)$ para $ln(1001)$. Assim, o protocolo escalável conseguiu reduzir a banda média em $ln(1001) + ln(2001)$, que é maior que os $ln(3001)$ economizados pelo protocolo convencional, daí o aumento do ganho. Note que se fosse considerado o uso de *unicast*, as duas alterações teriam o mesmo impacto sobre a banda média de rede, uma redução de 3000.

Grupo	Rótulo	*Site*	Localização
Sites1	48	registro.br	São Paulo
Sites1	381	guanabara.rederio.br	Rio de Janeiro
Sites1	341	nic-2.matrix.com.br	Florianópolis
Sites1	338	200.146.123.10	Florianópolis
Sites2	324	tools.telpin.com.ar	Argentina
Sites2	162	trace.megalink.com	Bolívia
Sites2	163	www.scbbs.net	Bolívia
Sites2	110	www.dcsc.utfsm.cl	Chile

Tabela 6.6: América do Sul - Sites1 e Sites2

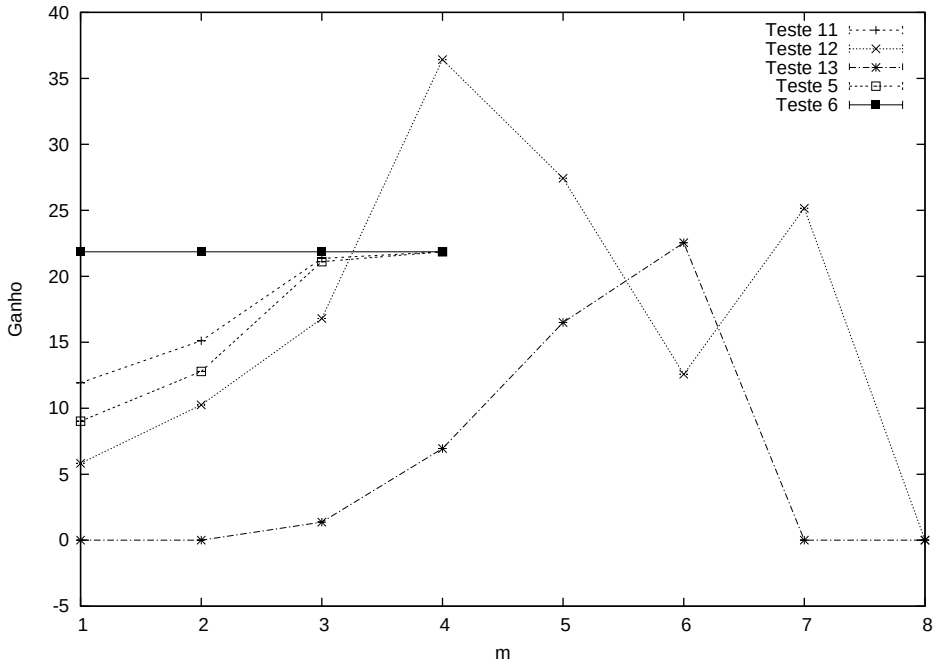
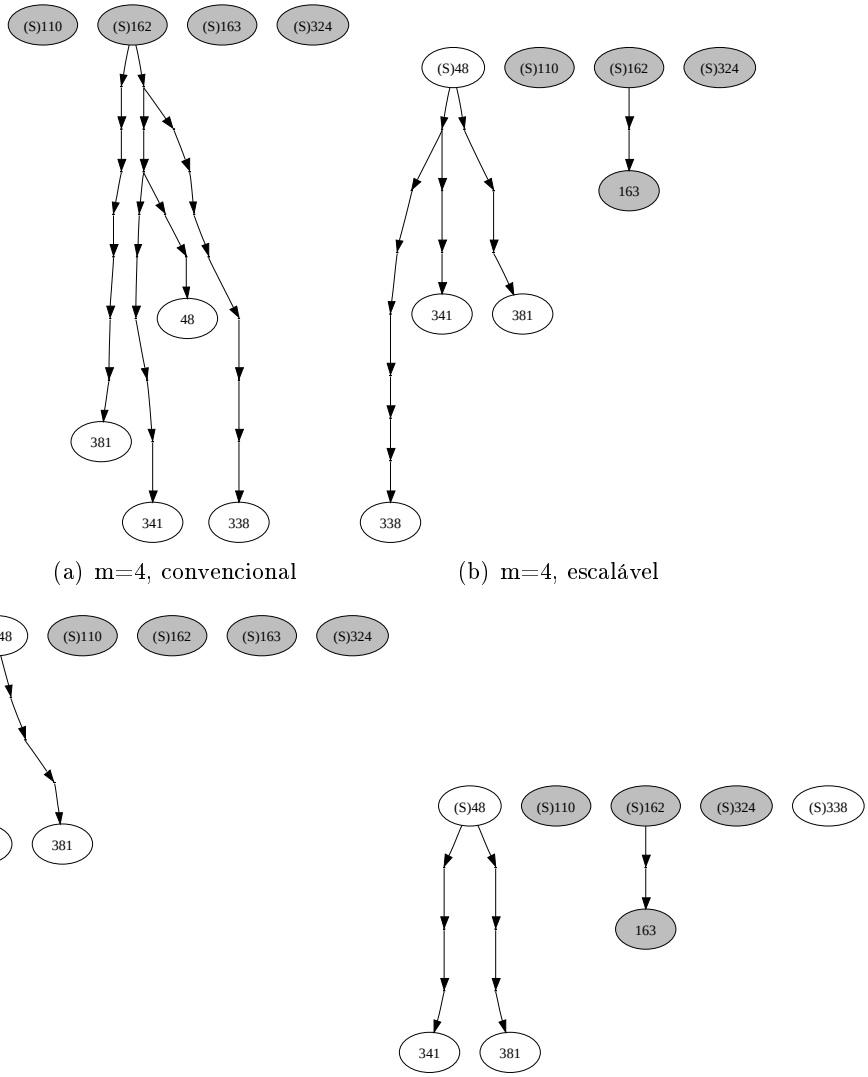


Figura 6.18: América do Sul - Ganho do MCO sobre o protocolo convencional para os tipos de teste selecionados

6.2.3.3 América do Sul

Nesta seção analisamos alguns dos testes realizados usando como configuração uma rede local composta por todos os *sites* da América do Sul. O conjunto Sites1 corresponde aos 4 *sites* do Brasil e o conjunto Sites2 aos demais, conforme mostra a tabela 6.6.

O gráfico da figura 6.18 mostra os tipos de teste selecionados para análise, nesta seção e no apêndice A. Veja que há ganhos significativos, de até 35% para essa configuração.



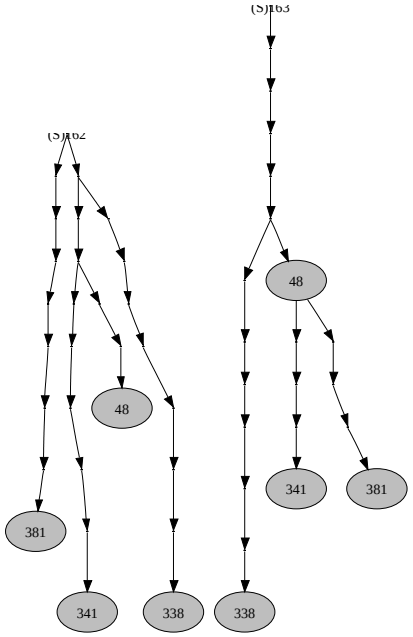
(a) m=4, convencional (b) m=4, escalável

(c) m=5, convencional (d) m=5, escalável

Figura 6.19: América do Sul - Teste 12

A figura 6.19 mostra as florestas de distribuição do teste 12 para os pontos $m = 4$ e $m = 5$. Nesse teste os *sites* brasileiros possuem demanda $N_b = 100$ e os demais demanda $N_o = 1000$ e as réplicas podem ser alocadas em quaisquer dos *sites*. Observa-se que, para $m = 4$ e $m = 5$, o protocolo convencional perde para o escalável devido à sua tendência em alocar as réplicas nos *sites* de maior demanda. Para $m = 4$ o ganho chega a 35%, conforme se vê no gráfico da figura 6.18. Isso ocorre porque, enquanto o protocolo convencional aloca todas as réplicas nos *sites* de maior demanda, ficando com uma árvore com pouco compartilhamento e com clientes distantes, o protocolo escalável constrói uma árvore com todos esses clientes distantes, que apresenta altura média bem inferior, apesar de não ter tanto compartilhamento. Com isso, o impacto do fator distância sobre a banda média, que é linear quando se usa compartilhamento de fluxos, é amenizado na floresta criada pelo protocolo escalável.

Já no teste 6 (figura 6.20), temos todos os *sites* clientes localizados no Brasil e os servidores



(a) m=1, convencional (b) m=1, escalável

Figura 6.20: América do Sul - Teste 6

podem ser alocados apenas nos *sites* fora do Brasil. Esse teste é interessante por apresentar um ganho constante de 22% para todos os valores de m. Essa constância deve-se ao fato de as mesmas florestas estarem sendo criadas para todos esses valores de m, sendo que para $m > 1$ as demais réplicas inseridas ficam ociosas, tanto para o protocolo convencional quanto para o escalável. Isso pode ocorrer quando não se pode alocar as réplicas em *sites* clientes, como é o caso do tipo de teste 6. Note nas florestas desse teste não há círculos ao redor dos *sites* servidores, o que indica que eles não são clientes. Ambos os protocolos preferem servir todos os clientes a partir de um único servidor, pois isso gera maior compartilhamento. O servidor escolhido pelo protocolo escalável (163) fica um pouco mais distante mas sua árvore provê um compartilhamento mais eficiente que a do servidor escolhido pelo protocolo convencional (162).

6.3 Conclusões

Em grande parte das situações, os protocolos escaláveis para roteamento com compartilhamento de fluxos apresentam algum ganho sobre o protocolo convencional, em termos de banda média de rede consumida. Os protocolos MCO e MCI em geral apresentam maiores ganhos que o protocolo SP em relação ao protocolo convencional e em geral apresentam ganhos parecidos entre si.

Dentre os casos analisados comparando o MCO com o convencional, observou-se um ganho de 35% com o uso de 50% dos *sites* como réplicas, na América do Sul. Na seção A.4 é mostrada uma situação com um ganho parecido, de 33%, usando apenas 44% dos *sites* como réplicas, em uma topologia que inclui *sites* da Europa, América do Norte, Ásia e África. Ganhos ainda maiores foram encontrados em outras situações em que se usou um maior percentual dos *sites* como réplicas.

Analisando caso a caso alguns testes, através da comparação entre as florestas geradas pelo protocolo MCO e pelo protocolo convencional, verificamos que:

- O caminho mais curto pode não ser a melhor opção para fluxos compartilhados.
- O protocolo convencional tende a alocar réplicas em *sites* de maior demanda, enquanto o protocolo escalável procura alocar réplicas em *sites* que estejam mais distantes e sem muita possibilidade de compartilhamento, mesmo que esses *sites* possuam demanda baixa. Isso ocorre porque a banda média de rede, quando se usa *unicast*, varia linearmente com relação à demanda e à distância, mas quando se usa compartilhamento de fluxos, a variação é linear apenas com a distância, sendo logaritmica com a demanda.
- Pode haver várias árvores que possuam o mesmo custo para *unicast* mas que tenham custos diferentes quando se usam fluxos compartilhados. Em especial, pode haver várias árvores dos caminhos mais curtos diferentes e o protocolo convencional apenas encontra uma dessas árvores, não necessariamente a melhor. Os protocolos MCO e MCI são capazes escolher a melhor, pois testam várias alternativas enquanto tentam otimizar o roteamento para compartilhamento de fluxos, levado em conta as demandas dos clientes.
- A escolha de réplicas de forma incremental simplifica bastante ambos os algoritmos, porém esse caráter incremental pode levar a árvores pouco eficientes para m>1, em algumas situações (tanto para o protocolo convencional quanto para o escalável).
- Quando há vários *sites* distantes que podem ser servidores e um conjunto de *sites* clientes próximos entre si, em geral apenas 1 dos *sites* distantes é escolhido para servir esse conjunto de clientes (veja como exemplo o caso dos 3 servidores ociosos fora do Brasil, quando $m = 4$ e todos os clientes estavam no Brasil).

Capítulo 7

Conclusão

7.1 Conclusões

Esta dissertação trata do problema de roteamento de fluxos de mídia contínua em topologias realistas da Internet, visando melhorar a qualidade de serviço provida aos usuários. Para tanto, coletamos dados que permitiram criar mapas topológicos realistas da Internet, em nível de roteadores, corrigindo problemas apontados por Teixeira et al [59] na metodologia de Spring et al [56, 57], que, até onde sabemos, é a metodologia que permite criar mapas topológicos mais precisos em nível de roteadores. A topologia global montada envolve 53 pontos cuja localização geográfica foi determinada, espalhados por todo o mundo, bem como a rede de roteadores que interconecta esses 53 pontos. Além de serem utilizadas neste trabalho, as topologias mapeadas podem ser úteis a outros grupos de pesquisa.

Como subsídio a pesquisas que se baseiam na diversidade de caminhos para prover QoS, caracterizamos as topologias coletadas com relação à diversidade de caminhos, além de realizar também uma caracterização quanto à assimetria de roteamento. A caracterização apontou para uma diversidade de caminhos alta, quando se consideram dispersões continentais ou intercontinentais, e uma diversidade baixa para regiões geograficamente reduzidas, como pequenos países. A mais alta diversidade encontrada foi na América do Sul, onde o grau de diferença médio entre os caminhos entre pares de *sites* ficou na faixa de 0,47 e o número de caminhos diferentes entre cada par ficou por volta de 14,5. A alta diversidade de caminhos encontrada em regiões geograficamente dispersas incentiva o uso de práticas [7, 24, 4, 5] que explorem essa diversidade de caminhos para melhorar a qualidade de serviço em aplicaçõs de mídia contínua.

A assimetria de roteamento também é alta. Esse resultado coincide com aqueles apresentados em He [25]. Verificou-se uma alta assimetria de roteamento até mesmo entre *sites* geograficamente próximos. Isso indica que pesquisas que assumem caminhos de ida e volta simétricos devem ser revisitadas.

Motivados pela alta diversidade de caminhos encontrada, utilizamos os mapas topológicos reais coletados para avaliar protocolos para roteamento de mídia contínua com compartilhamento de fluxos, especificamente os escaláveis MCI e MCO [4], bem como o SP associado

a localização otimizada para fluxos compartilhados. A alta diversidade de caminhos existente permite aos protocolos escaláveis encontrarem diversas outras florestas de distribuição, alternativas à dos menores caminhos e com custo menor para compartilhamento de fluxos. Almeida [2] quantifica penalidade associada ao uso da floresta dos caminhos mais curtos, dependendo da topologia de rede existente entre um servidor e dois *sites* clientes, daí a importância da avaliação de protocolos que geram árvores de distribuição alternativas. A avaliação dos protocolos mencionados se deu através da comparação da banda de rede média consumida nas florestas de distribuição geradas por esses protocolos com a banda nas florestas geradas pelo protocolo convencional. Os protocolos foram avaliados em em diversas topologias reais da Internet, cobrindo 6 continentes e variando diversos parâmetros, tais como heterogeidade de demanda, dispersão, número de *sites* participantes, número de réplicas e concentração relativa entre clientes e servidores. Nossa avaliação foi bem mais extensa que aquelas realizadas por Almeida et al [4, 2] e Zhao et al [70].

Como em Almeida et al [4], os três protocolos apresentaram ganhos sobre o convencional, em diversas situações. No entanto, enquanto naquele estudo os três protocolos apresentaram os mesmos ganhos para todas as configurações avaliadas, nas nossas análises foram encontrados casos em que os protocolos MCI e MCO mostraram-se superiores ao SP[1]. Os ganhos apresentados pelos protocolos MCI e MCO em relação ao protocolo convencional, em cerca de 20% dos casos, são mais de 10 pontos percentuais maiores que os ganhos apresentados pelo protocolo SP. Esse ganho dos protocolos que levam em conta a demanda dos *sites* clientes, MCI e MCO, sobre o protocolo mais simples, SP, foram observados especialmente em situações nas quais os servidores são forçados a ficar numa região periférica de baixa demanda, como no teste do tipo 11. Isso ocorre muito freqüentemente em situações nas quais os servidores estejam relativamente mais distantes de um conjunto de *sites* clientes e estes estão próximos entre si.

Foi observado também que o MCI e o MCO apresentam exatamente o mesmo ganho sobre o protocolo convencional em cerca de 70% das situações, apesar de o MCI possuir complexidade computacional cúbica com o número de *sites* clientes, enquanto a do MCO é apenas quadrática. Além disso, quando esses protocolos apresentaram ganhos diferentes, essa diferença foi inferior a 10 pontos percentuais. A heurística utilizada pelo MCO, que considera demanda dos clientes e distância até os servidores para determinar a ordem em que serão inseridos os clientes na criação da floresta de distribuição, é muito boa para fluxos compartilhados.

Já com relação às perdas desses protocolos em relação ao protocolo convencional, na maior parte das vezes em que as observamos, elas ocorreram devido ao problema do caráter incremental do protocolo de localização *Min-cost TSP* utilizado. Essa característica levou a perdas tanto nos protocolos escaláveis quanto no protocolo convencional, em situações distintas.

Dado o fato de o protocolo MCO apresentar ganhos semelhantes ao MCI e ambos apresentarem ganhos semelhantes ao ótimo nas situações mostradas em [4], pode-se considerar

[1] Isso ocorre apesar de o algoritmo usado para localização de todos os protocolos ser baseado em árvores dos menores caminhos, o que intuitivamente pareceria favorecer ao protocolo SP

o protocolo MCO melhor, por ele possuir menor complexidade computacional. Com relação ao percentual de *sites* utilizados como réplicas, foram notados maiores ganhos quando esse valor está entre 30% e 80%. Valores acima disso correspondem a situações em que há pouco roteamento, já que quase todos os *sites* funcionam como réplicas e portanto não são tão interessantes para estudo que analisa o roteamento. Usando até 50% dos *sites* como réplicas, foram observados ganhos em banda média de rede da ordem de 35%. Sem restrição no percentual de *sites* usados como réplicas, foram encontrados ganhos da ordem de 70%.

Em geral, ganhos maiores ocorreram quando se deu liberdade ao protocolo de localização para alocar réplicas em quaisquer dos *sites* participantes, isto é, nos testes 7, 12 e 13. O fato de o protocolo MCO considerar a heterogeneidade de demanda parece ser um fator de peso no seu desempenho superior, uma vez que maiores ganhos ocorreram nos testes 12 e 13, teste nos quais a demanda é heterogênea. Ganhos maiores ocorreram especialmente nos testes 13, nos quais a demanda da área central é superior à periférica. Isso ocorre porque o protocolo convencional tende a alocar réplicas nos *sites* de maior demanda, enquanto o MCO prefere aumentar o compartilhamento, alocando réplicas em *sites* periféricos, distantes, mesmo que de baixa demanda. Nota-se então que, numa situação na qual uma determinada comunidade está muito interessada em um conteúdo, mas há outras comunidades menores também interessadas, espalhadas por regiões distantes dessa comunidade, o protocolo MCO tenderia a apresentar desempenho superior ao convencional.

Outro fator que parece impactar positivamente o desempenho do MCO em relação ao protocolo convencional é a diversidade de caminhos. Maiores ganhos foram obtidos em situações onde foi medida maior diversidade, isto é, em topologias de dispersão continental e intercontinental, desde que o número de *sites* seja também razoável (pelo menos 8 *sites* em um continente é uma situação que geralmente leva a ganhos). Isso ocorre porque, nessas situações, há mais chances de haver *sites* clientes próximos entre si e distantes de um servidor e, como explicamos acima, nessas situações o protocolo SP (seja o convencional ou o escalável) não é tão bom quanto o MCO. O SP tende a criar diversos caminhos mais disjuntos até os clientes, enquanto o MCO tende a levar o conteúdo através de um único caminho até um dos clientes e de lá redistribuí-lo localmente.

Esses resultados incentivam o uso do protocolo MCO em situações em que é possível ter maior controle sobre o roteamento e determinar as demandas de cada *site*, isto é, dentro de um mesmo sistema autônomo, ou em CDNs implementadas via redes *overlay* sobre a Internet, especialmente em topologias esparsas, dispersas geograficamente, com *sites* distantes e alguns *sites* próximos entre si.

7.2 Trabalhos futuros

Esta dissertação motiva várias direções para trabalhos futuros. Com base na caracterização que realizamos, acreditamos ser interessante dar continuidade a pesquisas que exploram a diversidade de caminhos para prover garantias qualidade de serviço, uma vez que a diversidade existente hoje mostrou-se alta e que QoS provido pela própria infra-estrutura de rede ainda é

distante da presente realidade.

Quanto aos protocolos avaliados, acreditamos que o MCO ou alguma variante desse protocolo deve ser considerado como um candidato a protocolo padrão para utilização de fluxos compartilhados, em arquiteturas futuras da Internet, quando as questões relativas à utilização de fluxos compartilhados em uma rede administrativamente descentralizada como a Internet forem resolvidas. Para um futuro mais próximo, pode-se levar em conta o fato de que já é possível implementar esse protocolo em nível de aplicação, implantando-o em CDNs ou redes *overlay*. Assim, incentivamos a implementação de um protótipo de aplicação de distribuição de mídia contínua que utilize o BS [19] como protocolo de entrega e o MCO como protocolo de roteamento, seja por empresas que operam CDNs, seja pela comunidade científica, podendo-se testar tal protótipo em redes *overlay* ou em laboratórios globais como o PlanetLab [44].

Acreditamos que também devem ser avaliadas outras heurísticas para localização, uma vez que a heurística *Min-cost TSP* utilizada algumas vezes leva a perda de desempenho do protocolo MCO.

Com relação à coleta, seria possível obter mapas ainda mais precisos. Para tanto, teríamos que considerar rotas coletadas numa janela de tempo mais curta, por exemplo, da ordem poucos dias, ao invés de 4 meses, e a realização da resolução de interfaces sinônimas deveria ocorrer o mais rápido possível, de preferência durante o período de coletas. Esse aumento na precisão viria às custas de uma diminuição na completude. Provavelmente a diversidade de caminhos seria reduzida nesses mapas, pois caminhos alternativos normalmente usados como *backup* poderiam não aparecer em um período tão curto de tempo. Existe um compromisso entre precisão e completude que depende do estudo sendo feito. Assim, uma linha de trabalhos futuros seria montar diversos mapas, desde mapas mais precisos até mapas mais completos, que poderiam ser utilizados por outros trabalhos deste tipo, em especial simulações de aplicações que rodam sobre redes.

Apêndice A

Protocolo MCO X Convencional: outros estudos de casos

Todos os exemplos desta seção comparam florestas geradas pelo protocolo MCO com florestas geradas pelo protocolo convencional.

A.1 América do Norte

Nesta seção analisamos alguns dos testes realizados usando como configuração uma rede local composta por todos os *sites* da América do Norte. O conjunto $Sites1$ corresponde aos *sites* da costa oeste dos EUA e o conjunto $Sites2$ aos demais, conforme a tabela A.1.

O gráfico da figura A.1 mostra os tipos de teste selecionados para análise.

A figura A.2 mostra as florestas geradas pelos protocolos convencional e escalável para os pontos $m = 7$ e $m = 10$, no o teste 13 na América do Norte. Nesse tipo de teste, os *sites* do oeste dos EUA possuem demanda $N_o = 1000$ e os restantes possuem demanda $N_e = 100$ e

Grupo	Rótulo	*Site*	Localização (Estado)
Sites1	493	www.telcom.arizona.edu	Arizona
Sites1	93	www.net.berkeley.edu	California
Sites1	139	www.usc.edu	California
Sites1	460	www.slac.stanford.edu	California
Sites1	186	www.sdsc.edu	California
Sites1	282	www.washington.edu	Washington
Sites1	231	www.undergroundpalace.com	Colorado
Sites1	501	darkwing.uoregon.edu	Oregon
Sites2	143	www.net.cmu.edu	Pennsylvania
Sites2	53	home.acadia.net	Virginia
Sites2	278	noc.net.umd.edu	Maryland
Sites2	661	cgi.cs.wisc.edu	Wisconsin
Sites2	467	www.netsolutions.com.mx	México (país)

Tabela A.1: América do Norte - Sites1 e Sites2

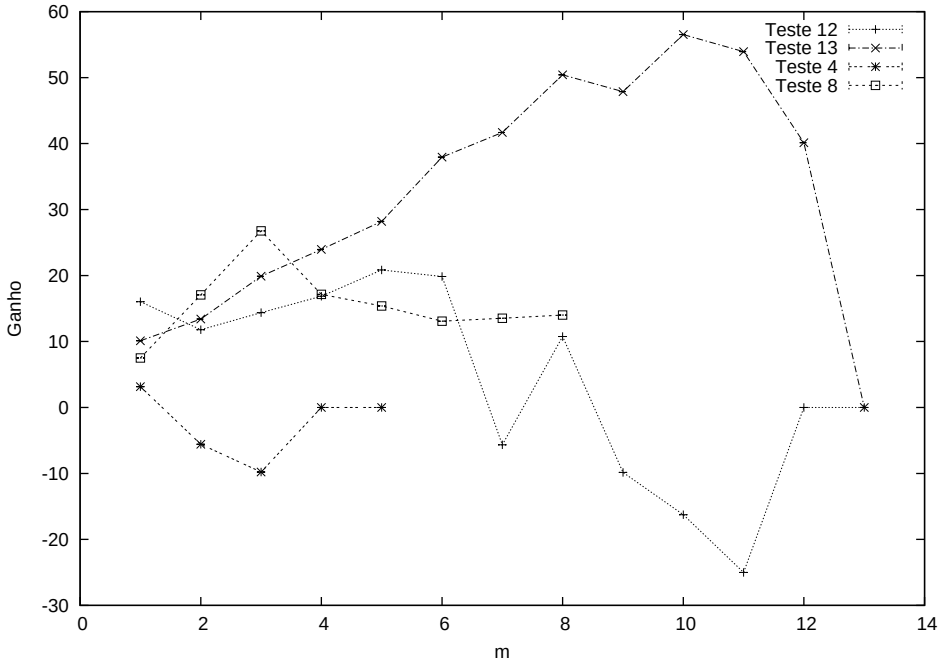


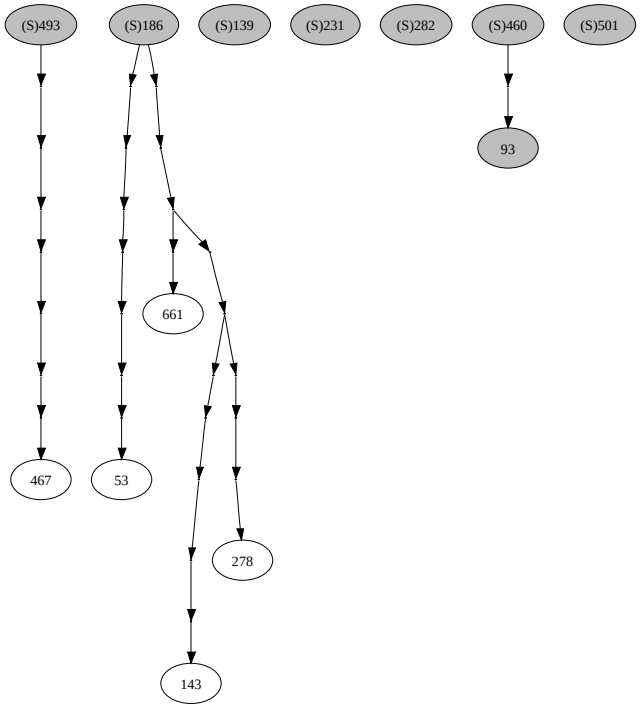
Figura A.1: América do Norte - Ganho do MCO sobre o protocolo convencional nos testes selecionados

as réplicas podem ser alocadas em quaisquer dos *sites*. Nota-se no ponto $m = 7$ a recorrente tendência do protocolo convencional em dar preferência à alocação de réplicas em *sites* de maior demanda, perdendo a chance de ter maior compartilhamento no ponto $m = 10$. Já o protocolo escalável, no ponto $m = 10$, ainda não havia alocado réplicas em 3 dos *sites* de demanda alta e obteve árvores mais baratas para compartilhamento de fluxos.

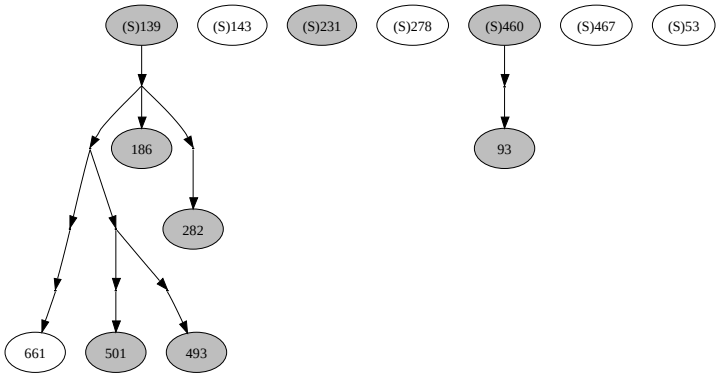
As figuras A.3 e A.4 mostram os pontos $m = 2$, $m = 3$ e $m = 4$ para o teste 8 na América do Norte. Nesse teste os *sites* do oeste possuem demanda $N_o = 100$ e os demais possuem demanda $N_e = 1000$. As réplicas somente podem ser alocadas nos *sites* do oeste (menor demanda). Neste caso não ocorre de o protocolo escalável alocar réplicas nos *sites* de maior demanda, porque não é permitido. Assim, as duas florestas, em $m = 2$, são bastante semelhantes. No entanto, enquanto o protocolo escalável prefere alocar a segunda réplica no *site* 231, que está distante e sem possibilidade de compartilhamento, o convencional prefere alocá-la no *site* 493, servindo assim os *sites* 467 e 501 por caminhos mais curtos. O escalável serve esses dois *sites* por caminhos maiores, porém mais compartilhados e, além disso, elimina o longo caminho sem compartilhamento até o *site* 231. Esse é mais um exemplo de como o menor caminho pode não ser a melhor opção para fluxos compartilhados.

Para $m = 3$, o protocolo escalável escolhe como réplica o nó 460, novamente um *site* distante e cujo caminho até ele não permite muito compartilhamento. Esse nó serve o 93, um *site* próximo dele. Com essa escolha, o protocolo escalável mantém a grande árvore com alto grau de compartilhamento e assim consegue um ganho ainda maior.

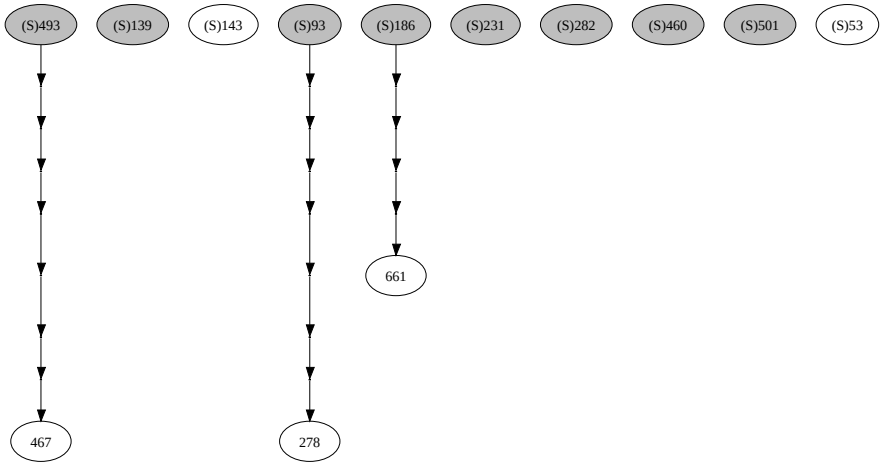
Finalmente, em $m = 4$, o protocolo convencional elimina o longo caminho sem compar-



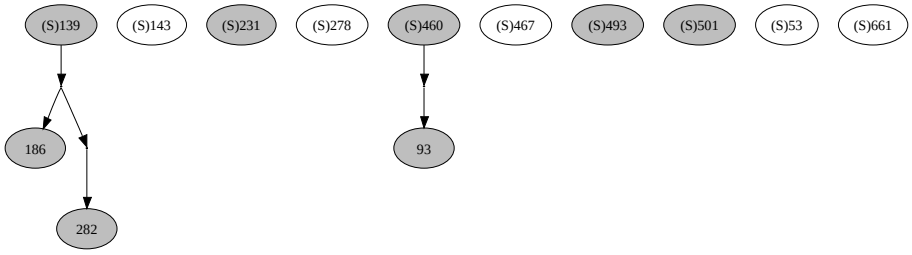
(a) m=7, convencional



(b) m=7, escalável

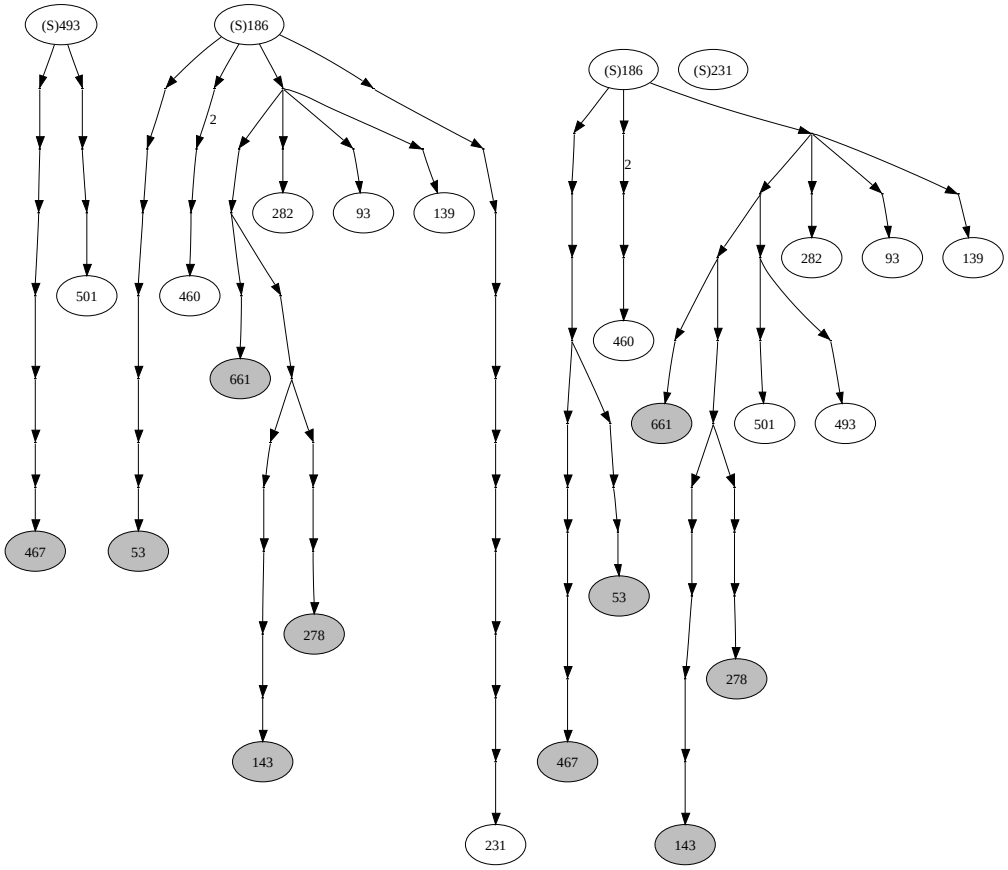


(c) m=10, convencional

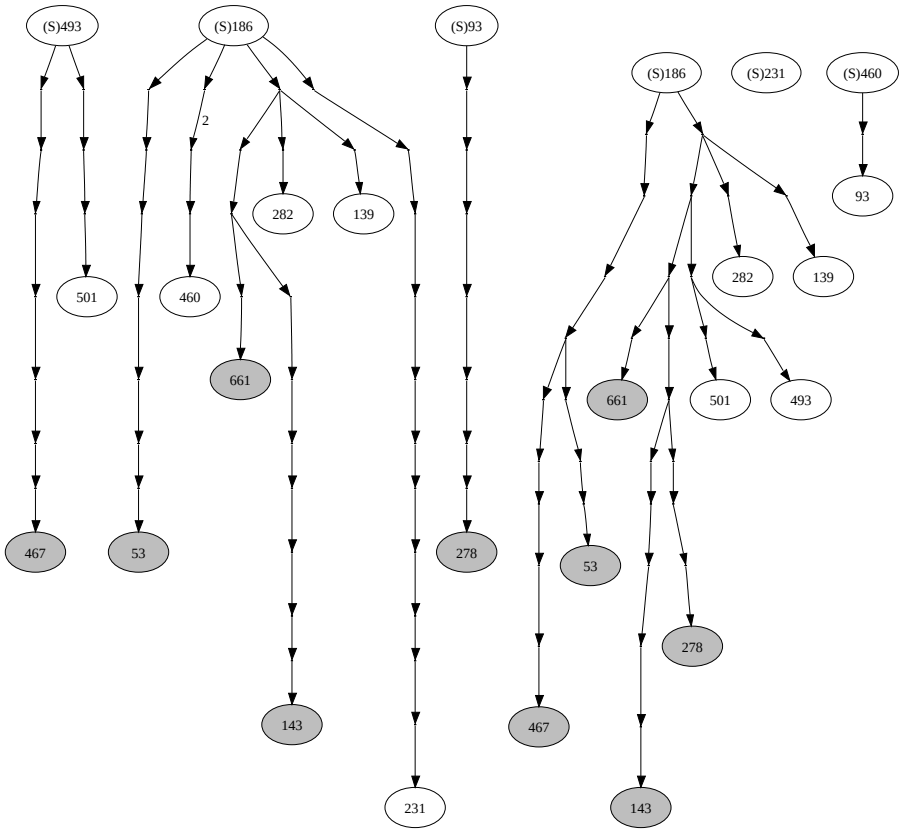


(d) m=10, escalável

Figura A.2: América do Norte - Teste 13

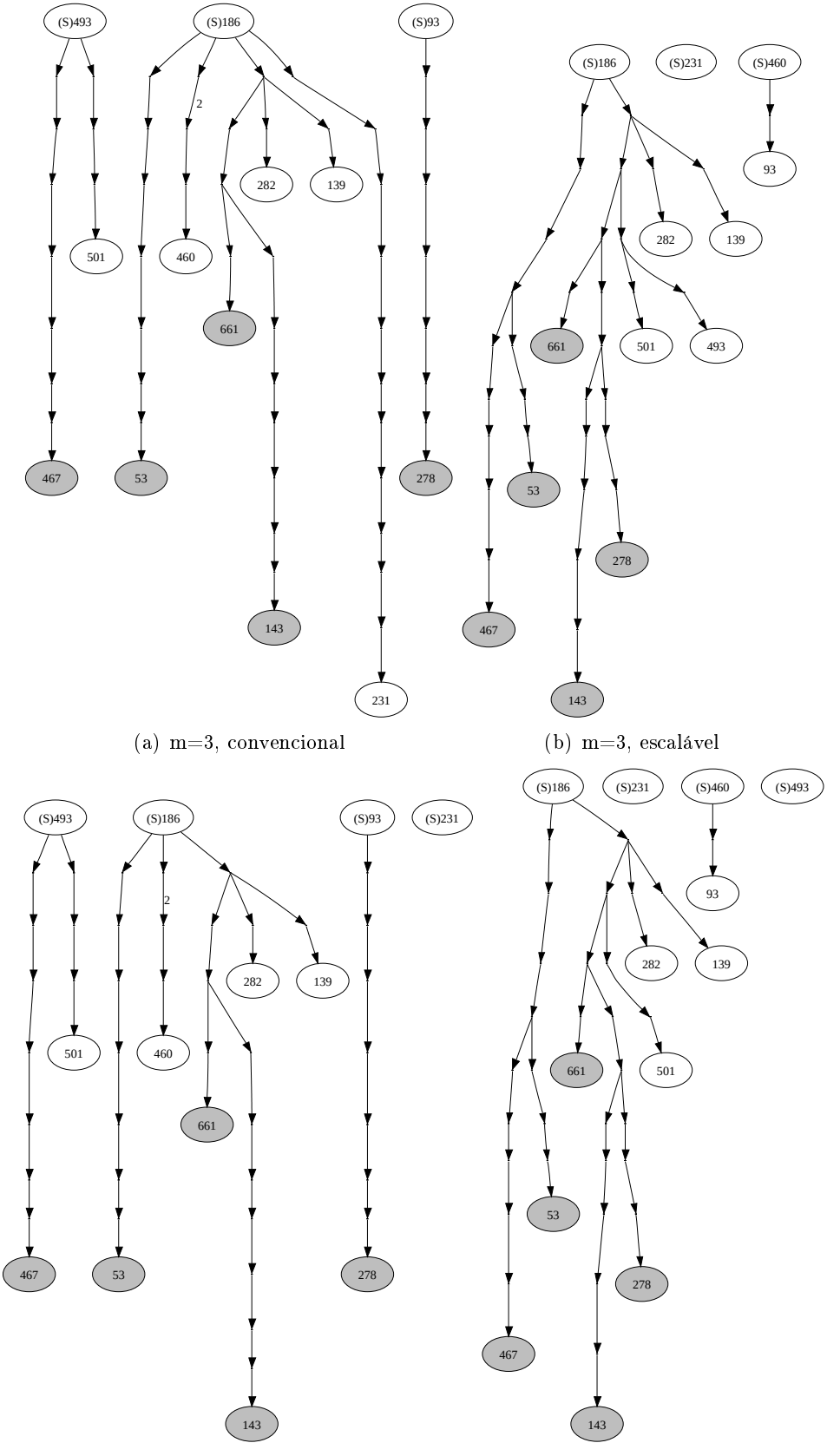


(a) m=2, convencional (b) m=2, escalável



(c) m=3, convencional (d) m=3, escalável

Figura A.3: América do Norte - Teste 8 - m={2,3}



(a) m=3, convencional

(b) m=3, escalável

(c) m=4, convencional

(d) m=4, escalável

Figura A.4: América do Norte - Teste 8 - m={3,4}

tilhamento que levava até o *site* 231, alocando uma réplica neste *site*. Com isso o ganho do escalável cai um pouco.

No teste do tipo 4 todos os *sites* possuem demanda $N = 1000$ e são usados somente os $Sites2$, isto é, o *site* mexicano e os *sites* do leste dos EUA. A figura A.5 mostra as florestas criadas para $m = 1$, $m = 2$ e $m = 3$. Mostraremos por que o protocolo escalável acumulou perda nesses pontos.

No ponto $m = 1$ o protocolo escalável consegue um pequeno ganho ($<5\%$) às custas da construção de uma árvore mais alta, porém mais compartilhada. Note que árvores muito altas podem ser ruins, pois aumentam a chance de perda de pacotes, além do atraso ser maior.

Em $m = 2$ vemos que a escolha da primeira réplica pelo escalável acabou sendo ruim. O ganho para $m = 1$ existiu mas foi baixo, além de levar a uma árvore muito mais alta. A inserção da segunda réplica implicou na redução do compartilhamento que dava o pequeno ganho inicial e o protocolo convencional acabou apresentando custo menor.

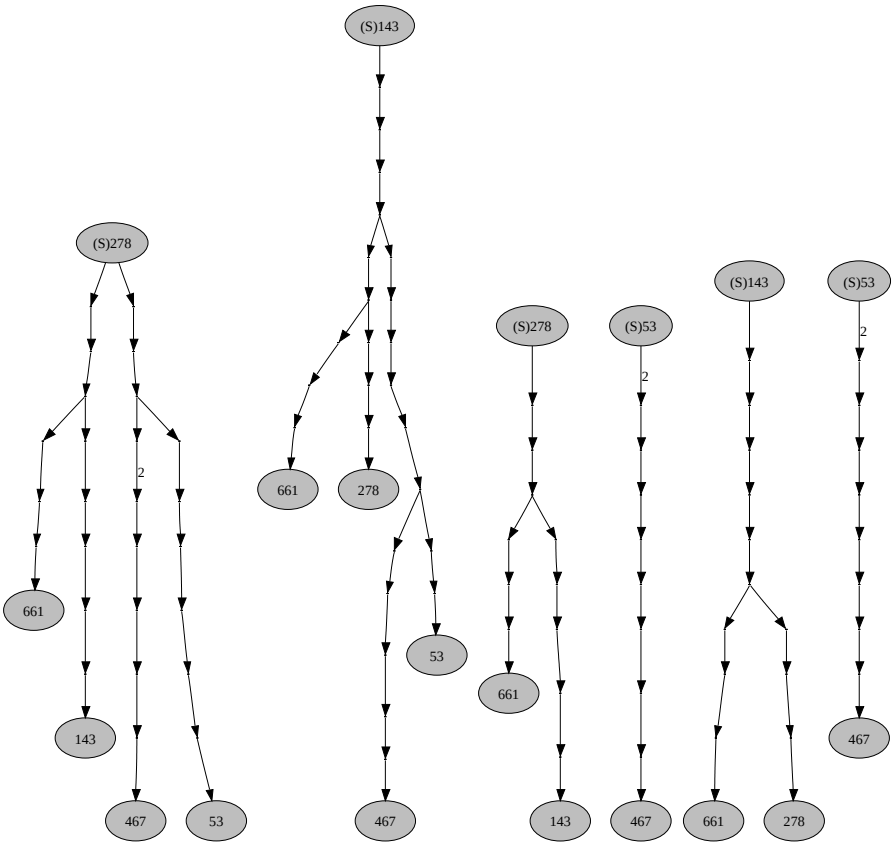
Em $m = 3$, ocorre um fenômeno interessante. A mesma modificação foi feita nas duas florestas: a eliminação da árvore do *site* 53, com a alocação de uma réplica no *site* 467, que era cliente do *site* 53 em ambas as árvores do caso $m = 2$. No entanto, a perda do protocolo escalável aumentou de pouco mais de 6% para mais de 10%. O que aconteceu foi que a perda absoluta se manteve em 9 nas duas situações ($m = 2$ e $m = 3$), porém, como a banda de rede total diminuiu devido à alocação da terceira réplica, a perda relativa aumentou.

Analisamos agora o teste 12 para a América do Norte, no qual os *sites* do oeste possuem demanda $N_o = 100$ e os restantes demanda $N_e = 1000$ e as réplicas podem ser alocadas a quaisquer dos *sites*. A figura A.6 mostra as florestas para a transição da situação de ganho do protocolo escalável para a situação de ganho do protocolo convencional, isto é, $m = 8$ e $m = 9$.

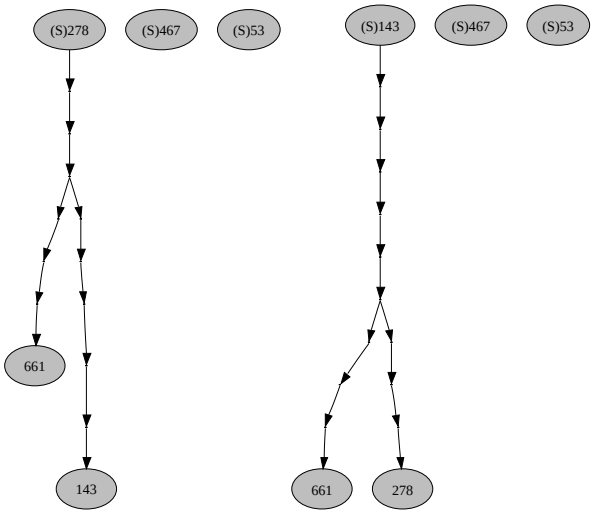
Em $m = 8$, ambos os protocolos já haviam alocado réplicas em todos os *sites* de demanda alta. Porém o protocolo escalável apresenta uma árvore bastante compartilhada e outra de pequena, de altura 2, sem compartilhamento, enquanto o protocolo convencional criou duas árvores sem compartilhamento mais longas, de alturas 4 e 5, além de uma pequena árvore compartilhada. Por ter duas árvores não compartilhadas, o protocolo convencional possui um *site* a menos na árvore compartilhada. Daí o ganho do protocolo escalável.

Na inserção da réplica seguinte ($m = 9$), o protocolo escalável acaba podando um pedaço da árvore compartilhada para criar uma árvore sem compartilhamento de altura 4, igual à que existia para o convencional em $m = 8$. Enquanto isso, o protocolo convencional elimina a árvore sem compartilhamento de altura 5 e passa a ganhar do escalável.

Nota-se que nesse ponto a única diferença entre as duas florestas é na árvore compartilhada. A do protocolo convencional é $ln(101)$ mais custosa. Para $m = 10$ e $m = 11$, essa mesma perda absoluta na árvore compartilhada se mantém, com os dois protocolos fazendo os mesmos tipos de alteração. Porém a perda relativa vai aumentando, já que a banda de rede total cai devido à inserção de novas réplicas.

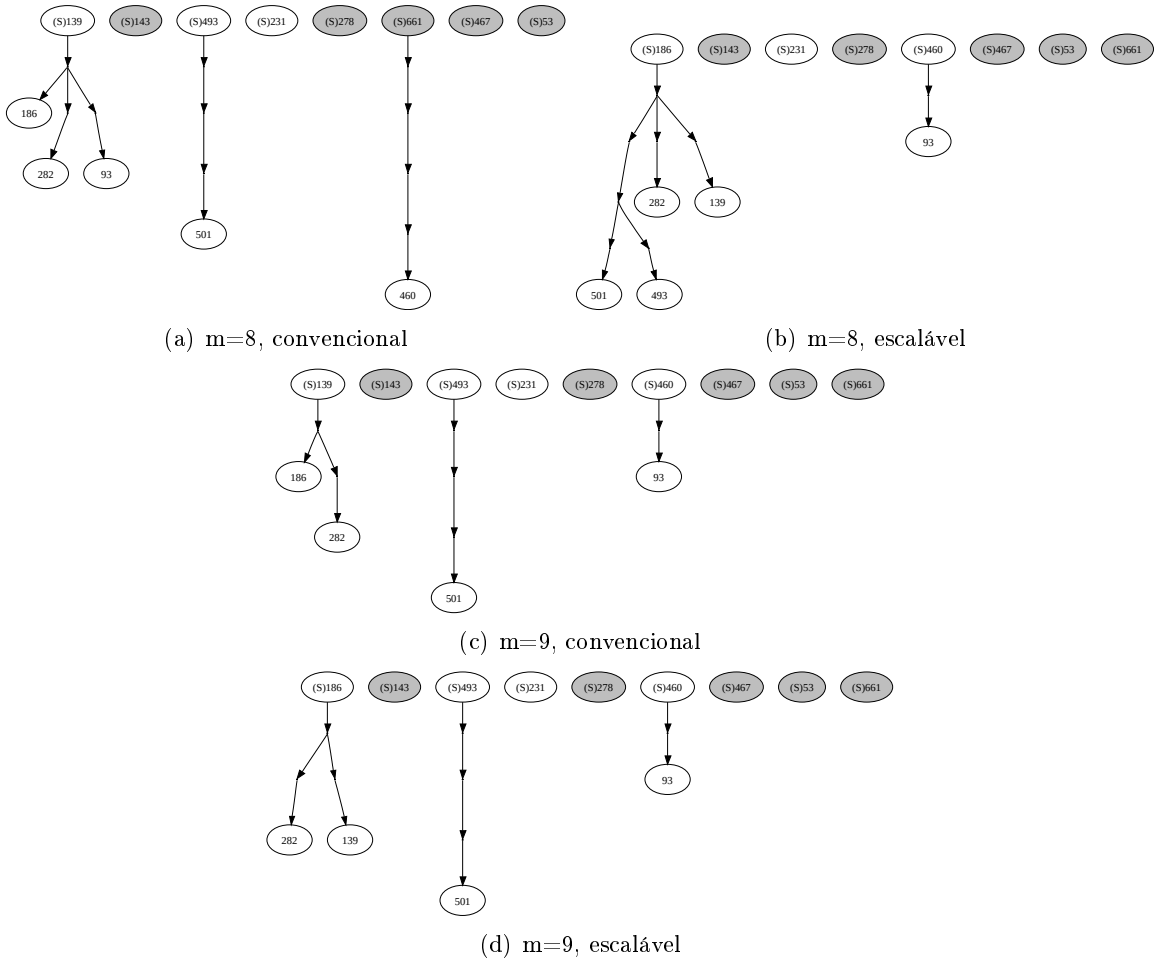


(a) m=1, convencional (b) m=1, escalável (c) m=2, convencional (d) m=2, escalável



(e) m=3, convencional (f) m=3, escalável

Figura A.5: América do Norte - Teste 4



(a) m=8, convencional (b) m=8, escalável

(c) m=9, convencional

(d) m=9, escalável

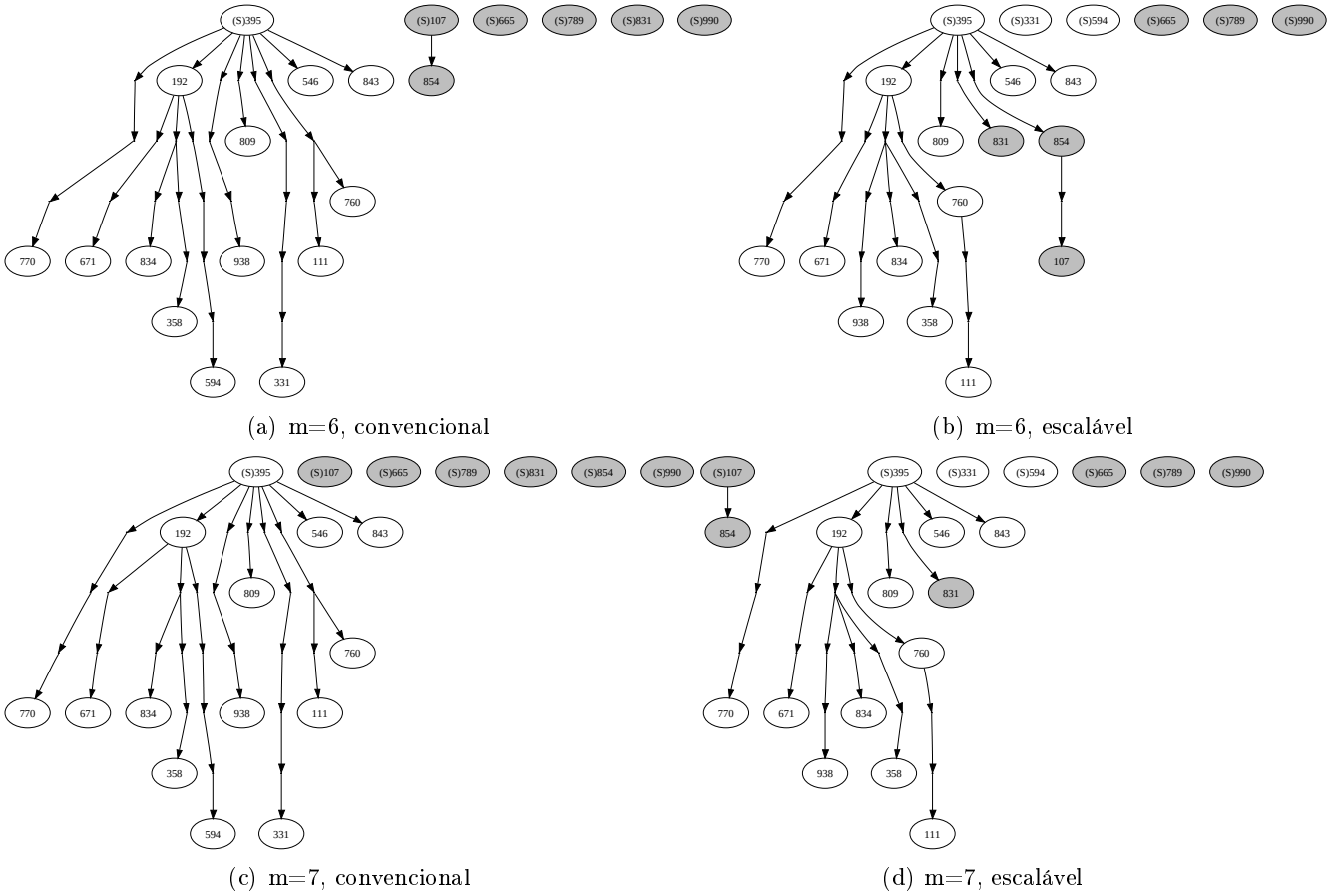
Figura A.6: América do Norte - Teste 12

A.2 Europa

Veja pelo gráfico da figura 6.14 que, na transição de $m = 6$ para $m = 7$, o ganho do protocolo escalável sobre o convencional aumenta de cerca de 10% para cerca de 20%. A figura A.7 mostra as florestas correspondentes a essa transição. Nesse teste as réplicas podem ser alocadas em quaisquer dos *sites* e os *sites* do leste possuem demanda $N_e = 1000$, enquanto os do oeste possuem demanda $N_o = 100$. Nota-se, em $m = 6$, que o protocolo convencional mais uma vez segue sua tendência de alocar réplicas nos *sites* de maior demanda, enquanto o escalável privilegia o grau de compartilhamento. Nesse ponto, o protocolo escalável já havia alocado réplicas em 2 *sites* que, na árvore dos menores caminhos (veja a árvore do convencional) apresentavam pouco compartilhamento: os *sites* 331 e 594.

A.3 América do sul

No teste 11 da América do Sul os *sites* brasileiros possuem demanda $N_b = 1000$ e os outros possuem demanda $N_o = 100$. Nesse teste não é permitido alocar servidores nos *sites* brasilei-



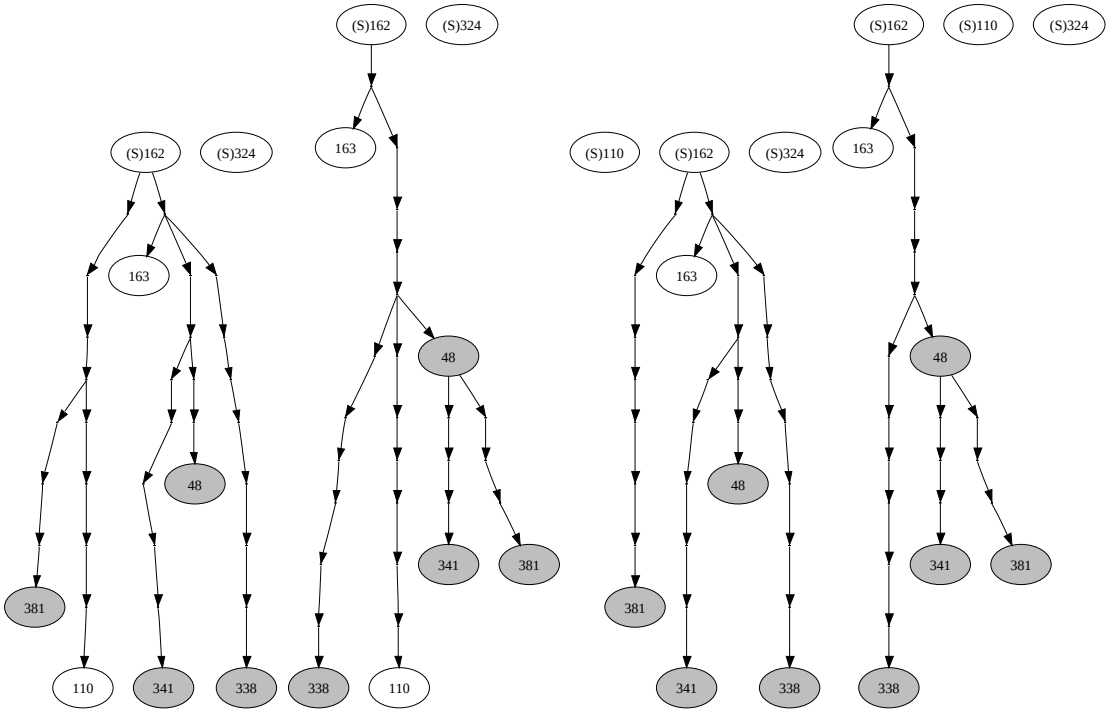
(a) m=6, convencional (b) m=6, escalável

(c) m=7, convencional (d) m=7, escalável

Figura A.7: Europa - Teste 13

ros. A figura A.8 mostra as florestas dos protocolos convencional e escalável para os pontos $m = 2$ e $m = 3$, onde o protocolo escalável obteve ganhos de 15% e 20%, conforme se vê pelo gráfico 6.18. Nota-se que para $m = 2$ e $m = 3$, ambos os protocolos escolhem os mesmos *sites* como réplicas. No entanto, similarmente ao teste 6, o protocolo escalável cria uma árvore um pouco mais alta, com altura 12, mas com maior compartilhamento, enquanto o protocolo convencional escolhe uma árvore com caminhos mais curtos de altura 9, menos eficiente para compartilhamento de fluxos. O ganho do teste 5 ocorre de maneira similar ao do teste 11.

Nos pontos $m = 4$, $m = 5$ e $m = 6$ do teste 13, vemos um ganho crescente do protocolo escalável sobre o convencional de, respectivamente, 6%, 16% e 22%. No teste 13 (figura A.9) os clientes possuem as mesmas demandas do teste 11, porém neste caso as réplicas podem ser alocadas em quaisquer dos *sites*. Com isso, no ponto $m = 4$ vemos que o protocolo convencional manteve sua tendência de alocar as réplicas nos *sites* de maior demanda (todos brasileiros), enquanto o escalável alocou duas delas em *sites* de menor demanda. Na inserção da quinta réplica, $m = 5$, o protocolo escalável obtém um ganho ao optar por deixar um *site* brasileiro servir os dois *sites* bolivianos (demanda baixa) através de um longo trecho compartilhado, além de alocar uma réplica em um *site* de maior demanda.



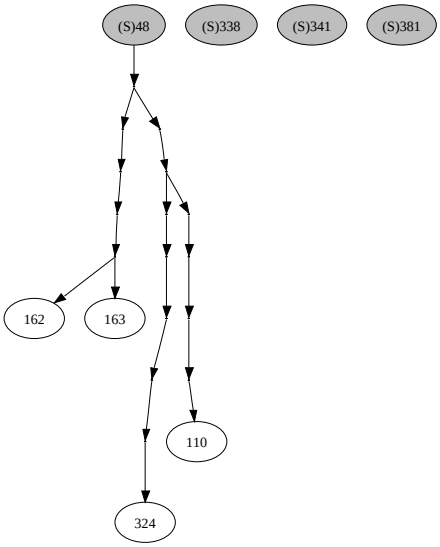
(a) m=2, convencional (b) m=2, escalável (c) m=3, convencional (d) m=3, escalável

Figura A.8: América do Sul - Teste 11

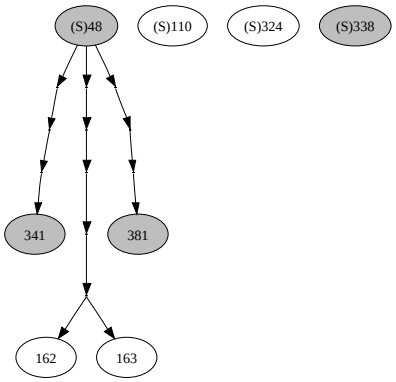
A.4 EUA e Ásia, Europa e África

A figura A.10 apresenta um exemplo de teste contendo *sites* de vários continentes. Corresponde ao teste do tipo 13, para uma configuração real cujo conjunto $Sites1$ ($N_{Sites1} = 1000$) é composto por vários *sites* localizados na Europa e um no norte da África (em Togo) e o conjunto Sites2 ($N_{Sites2} = 100$) é composto principalmente por *sites* localizados nos Estados Unidos, além de um *site* no Japão e outro em Taiwan. Esses dois conjuntos de *sites* são 2 dos 5 grupos obtidos a partir de uma aglomeração usando distância global.

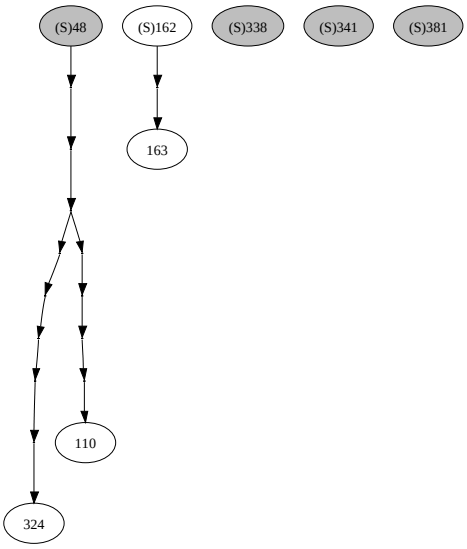
São mostradas as florestas geradas pelos protocolos escalável e convencional para o caso em que há 8 réplicas. Nesse caso o ganho do protocolo escalável, em termos de banda média de rede, é de 33%. Nota-se uma tendência do protocolo convencional em alocar réplicas preferencialmente nos *sites* de maior demanda (já que isso tende a ser melhor para *unicast*), enquanto o protocolo escalável aloca algumas réplicas em *sites* de menor demanda, obtendo assim um ganho quando se usa compartilhamento de fluxos. Repare que, caso usássemos ambas as florestas para distribuir dados usando *unicast*, realmente a floresta gerada pelo protocolo convencional teria um custo bem menor (da ordem de 50% menor).



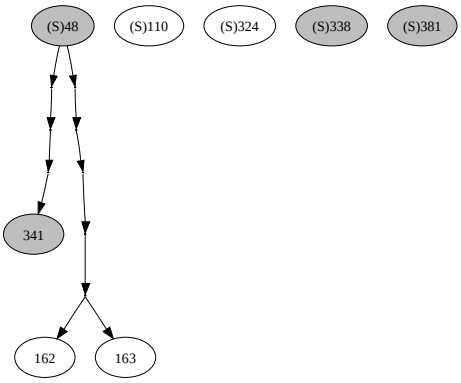
(a) m=4, convencional



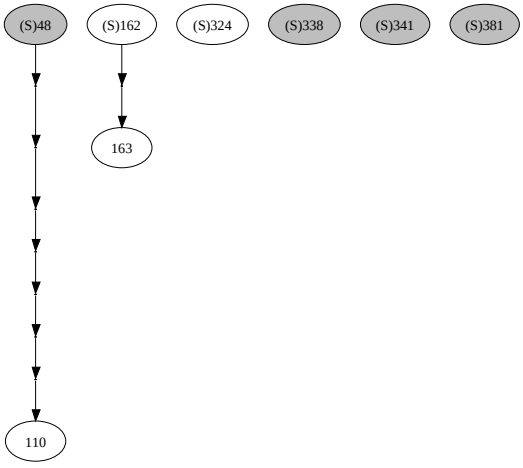
(b) m=4, escalável



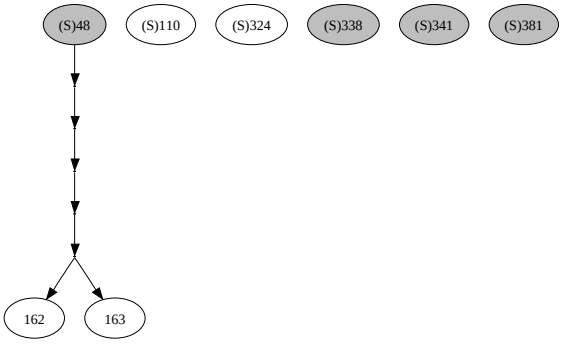
(c) m=5, convencional



escalável]
(d)
m=5,

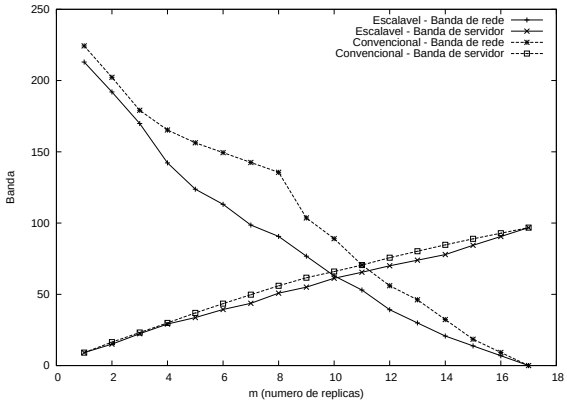


(e) m=6, convencional

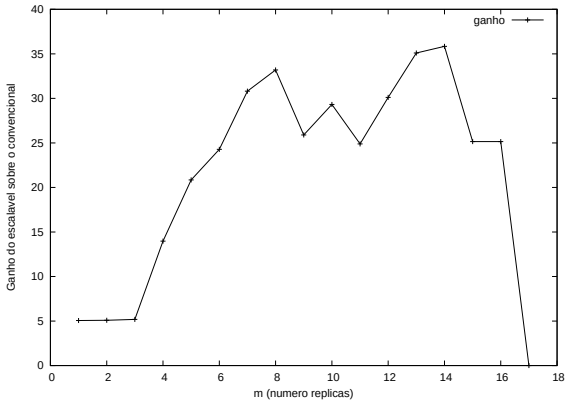


(f) m=6, escalável

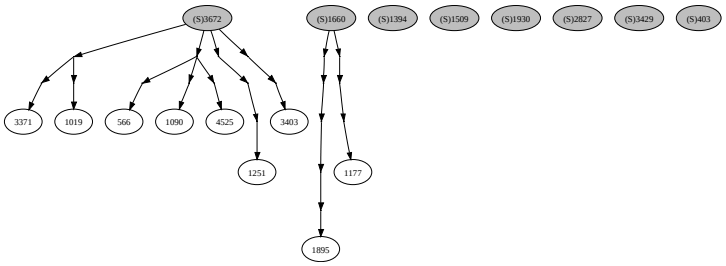
Figura A.9: América do Sul - Teste 13



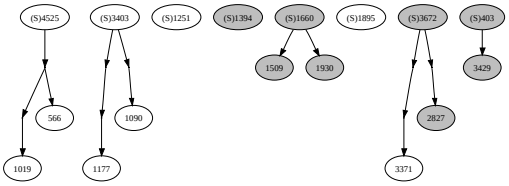
(a) banda média de rede e de servidor para os protocolos escalável e convencional



(b) Ganho percentual do protocolo escalável sobre o convencional (banda média de rede)



(c) Floresta gerada pelo protocolo convencional no ponto m=8



(d) Floresta gerada pelo protocolo escalável no ponto m=8

Figura A.10: Clientes nos EUA, Japão e Taiwan, com demanda igual a 100 (sites brancos). Clientes na Europa e África com demanda igual a 1000 (sites cinzas)

Referências Bibliográficas

[1] J. Almeida. Provisioning content distribution networks for streaming media. Em *INFOCOM 2002*, 2002.

[2] J. Almeida. *Streaming content distribution networks with minimum delivery cost.* PhD thesis, Madison, Wisconsin, 2003.

[3] J. Almeida, D. Eager, and M. Vernon. A hybrid caching strategy for streaming media files. Em *SPIE/ACM Conference on Multimedia Computing and Networking*, 2001.

[4] J. Almeida, D. Eager, M. Vernon, and S. Wright. Minimizing delivery cost in scalable streaming content distribuition systems. Em *IEEE Transactions on Multimedia*, volume 6, páginas 356 – 365, abril 2004.

[5] D. Andersen, H. Balakrishnan, M. Kaashoek, and R. Morris. Resilient overlay networks. páginas 131–145, Banff, Canada, 2001. ACM Press.

[6] J. Apostolopoulos. Reliable video communication over lossy packet networks using multiple state encoding and path diversity. Em *Visual Communication and Image Processing VCIP '01*, 2001.

[7] J. Apostolopoulos, T. Wong, W. Tan, and S. Wee. On multiple description streaming with content delivery networks. Em *IEEE Infocom*, 2002.

[8] Apple. Darwin Streaming Server. http://developer.apple.com/darwin/projects/streaming.

[9] H. Balakrishnan, V. Padmanabhan, and R. Katz. The effects of asymmetry on TCP performance. Em *Mobile Computing and Networking*, páginas 77–89, 1997.

[10] CAIDA. AS datasets - http://sk-aslinks.caida.org/.

[11] CAIDA. http://www.caida.org.

[12] K. Calvert and E. Zegura. Internetwork topology models - http://www-static.cc.gatech.edu/projects/gtitm/.

[13] Q. Chen, H. Chang, R. Govidan, S. Jamin, S. Shenker, and W. Willinger. AS dataset - http://topology.eecs.umich.edu/archive/asgraph.tar.gz.

[14] L. Cherkasova and M. Gupta. Characterizing locality, evolution, and life span of accesses in enterprise media server workloads, 2002.

[15] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling policies for an on-demand video server with batching. Em *ACM Multimedia*, páginas 15–23, 1994.

[16] D. Eager, M. Ferris, and M. Vernon. Optimized regional caching for on-demand data delivery. Relatório técnico CS-TR-1998-1385, 1998.

[17] D. Eager, M. Ferris, and M. Vernon. Optimized caching in systems with heterogeneous client populations. *Performance Evaluation*, 42(2-3):163–185, 2000.

[18] D. Eager and M. Vernon. Dynamic skyscraper broadcasts for video-on-demand. *Lecture Notes in Computer Science*, 1508, 1998.

[19] D. Eager, M. Vernon, and J. Zahorjan. Bandwidth skimming: A technique for cost-effective video-on-demand. Em *Proc. MMCN*, San Jose, California, Jan 2000.

[20] D. Eager, M. Vernon, and J. Zahorjan. Minimizing bandwidth requirements for on-demand data delivery. *Knowledge and Data Engineering*, 13(5):742–757, 2001.

[21] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. RFC 1998 - Protocol Independent Multicast-Sparse Mode (PIM-SM): especificação de protocolo, 1998.

[22] Z. Fei, M. Ammar, and E. Zegura. Multicast server selection: Problems, complexity and solutions. Em *IEEE Journal on Selected Areas in Communications*, volume 20, páginas 1399–1413, setembro 2002.

[23] World Gazetteer. (coordenadas geográficas de cidades). http://www.world-gazetteer.com/wg.php.

[24] L. Golubchik, J. C. S. Lui, T. F. Tung, A. L. H. Chow, W.-J. Lee, G. Franceschinis, and C. Anglano. Multi-path continuous media streaming: what are the benefits? *Perform. Eval.*, 49(1-4):429–449, 2002.

[25] Y. He. How asymmetric is internet routing? A systematic approach. Em *SIGCOMM*, Philadelphia, PA, USA, 2005.

[26] Y. He, M. Faloutsos, and S. Krishnamurthy. Quantifying routing asymmetry in the internet at the as level. Em *IEEE GLOBECOM 2004 - Global Internet and Next Generation Networks*, Dallas, Texas, USA, 2004.

[27] A. Heydon and M. Najork. Mercator: A scalable, extensible web crawler. *World Wide Web*, 2(4):219–229, 1999.

[28] K. A. Hua, Y. Cai, and S. Sheu. Patching : a multicast technique for true video-on-demand services. Em *MULTIMEDIA '98: Proceedings of the sixth ACM international conference on Multimedia*, páginas 191–200, New York, NY, USA, 1998. ACM Press.

[29] indo.com. How far is it - http://www.indo.com/cgi-bin/dist.

[30] V. Jacobson. traceroute (manual online). Disponível em http://www.zytek.com/traceroute.man.html.

[31] S. Jamin, C. Jin, A. Kurc, D. Raz, and Y. Shavitt. Constrained mirror placement on the internet. Em *INFOCOM*, páginas 31–40, 2001.

[32] G. Karypis. Cluto clustering toolkit. http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview.

[33] T. Kernen. Public traceroute servers - http://www.traceroute.org.

[34] P. Krishnan, D. Raz, and Y. Shavitt. The cache location problem. *IEEE/ACM Transactions on Networking*, 8(5):568–582, 2000.

[35] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. Em *Soviet Physics Doklady*, volume 10, páginas 707–710, 1966.

[36] B. Li, M. Golin, G. Italiano, and X. Deng. On the optimal placement of web proxies in the internet. Em *Proceedings of the Conference on Computer Communication, IEEE Infocomm*, New York, 1999.

[37] K. Lougheed, Y. Rekhter, and T. Li. RFC 1771 - A Border Gateway Protocol 4 (BGP-4), 1995.

[38] Z. Mao, J. Rexford, J. Wang, and R. Katz. Towards an accurate AS-level traceroute tool. Em *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, páginas 365–378, New York, NY, USA, 2003. ACM Press.

[39] Microsoft. Windows Media Server. http://www.microsoft.com/windows/windowsmedia/9series/server.aspx.

[40] Microsoft. Windows Netmeeting. http://www.microsoft.com/windows/netmeeting/.

[41] J. Moy. RFC 1584 - multicast extensions to OSPF, 1994.

[42] H. Niksic. GNU Wget - the non-interactive network downloader - http://www.gnu.org/software/wget/.

[43] NLANR. The National Laboratory for Applied Network Research - http://www.nlanr.net.

[44] PlanetLab. An open platform for developing, deploying, and accessing planetary-scale services - http://www.planet-lab.org.

[45] PUC. PUC Virtual - http://www.virtual.pucminas.br/default.htm.

[46] L. Qiu, V. Padmanabhan, and G. Voelker. On the placement of web server replicas. Em *INFOCOM*, páginas 1587–1596, 2001.

[47] P. Radoslavov, R. Govindan, and D. Estrin. Topology-informed internet replica placement. Em *Proceedings of WCW'01: Web Caching and Content Distribution Workshop, Boston, MA*, junho 2001.

[48] S. Ramesh, I. Rhee, and K. Guo. Multicast with cache (mcache): An adaptive zero delay video-on-demand service. Em *INFOCOM*, páginas 85–94, 2001.

[49] Real. Servidores Real Media. http://www.realnetworks.com/products/media_delivery.html.

[50] RIPE. The RIPE Routing Information Services - http://www.ris.ripe.net/.

[51] Routeviews. The route views project - http://www.routeviews.org/.

[52] D. Rubenstein, J. Kurose, and D. Towsley. Detecting shared congestion of flows via end-to-end measurement. Em *Measurement and Modeling of Computer Systems*, páginas 145–155, 2000.

[53] D. Sandras. GnomeMeeting - http://www.gnomemeeting.org.

[54] S. Sen, J. Rexford, and D. Towsley. Proxy prefix caching for multimedia streams. Em *INFOCOM (3)*, páginas 1310–1319, 1999.

[55] Skype. Skype - free internet telephony - http://www.skype.com.

[56] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel, 2002.

[57] N. Spring, R. Mahajan, and D. Wetherall. Rocketfuel maps and data, 2005. http://www.cs.washington.edu/research/networking/rocketfuel/.

[58] R. Teixeira, K. Marzullo, S. Savage, and G. Voelker. Characterizing and measuring path diversity of internet topologies. *SIGMETRICS Perform. Eval. Rev.*, 31(1):304–305, 2003.

[59] R. Teixeira, K. Marzullo, S. Savage, and G. Voelker. In search of path diversity in ISP networks. Em *Proceedings of the ACM SIGCOMM Internet Measurement Conference*, 2003.

[60] umass.edu. Ripples - http://ripples.cs.umass.edu/.

[61] UOL. Rádio UOL - http://musica.uol.com.br/radiouol/.

[62] UOL. TV UOL - http://tvuol.uol.com.br/.

[63] Usina do Som. http://www.usinadosom.com.br.

[64] O. Verscheure, C. Venkatramani, P. Frossard, and L. Amini. Joint server scheduling and proxy caching for video delivery. *Computer Communications*, 25(4):413–423, 2002.

[65] P. Vixie. cron - daemon to execute scheduled commands.

[66] B. Wang, S. Sen, M. Adler, and D. Towsley. Optimal proxy cache allocation for efficient streaming media distribution. *IEEE Transaction on Multimedia*, 6(2):366–274, abril 2004.

[67] wisc.edu. eTEACH - http://eteach.engr.wisc.edu/neweteach/home.html.

[68] E. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. Em *IEEE Infocom*, volume 2, páginas 594–602, San Francisco, CA, março 1996. IEEE.

[69] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-level topology. *SIGCOMM Comput. Commun. Rev.*, 35(1):53–61, 2005.

[70] Y. Zhao, D. Eager, and M. Vernon. Network bandwidth requirements for scalable on-demand streaming. Em *Proc. INFOCOM*, New York, NY, junho 2002.