Uma Abordagem para Investigação de Sistemas Distribuídos baseados em Grupo na Internet

Marinho P. Barcellos, Maglan C. Diemer, André Detsch

[1] PIPCA - Programa de Pós-Graduação em Computação Aplicada
Centro de Ciências Exatas e Tecnológicas - UNISINOS
Av. Unisinos, 950 - São Leopoldo, RS - CEP93022-000
{marinho,maglan,detsch}@exatas.unisinos.br

Resumo. Embora uma tecnologia madura e com casos de sucesso, comunicação em grupo ainda não é aplicada de forma geral na Internet. Uma das razões para isso é a percepção de que sistemas de comunicação em grupo exibem requisitos de comunicação e sincronização incompatíveis com cenários de larga escala da Internet. Avaliação experimental desses sistemas oferece resultados seguros, porém é difícil de ser realizada devido a questões de logística e é difícil de ser reproduzida. (Quando IP multicast é necessário, experimentos são ainda mais difíceis.) Simulação, nesse caso, pode ser usada, pois permite a investigação do sistema em questão sob cenários arbitrários, com e sem defeitos.

Este artigo propõe o uso de um simulador de redes para investigação de sistemas distribuídos baseados em grupo, permitindo avaliar comportamento geral e desempenho dos protocolos em cenários "realísticos" em termos de Internet. A metodologia proposta é apresentada através de um caso prático: a modelagem e simulação de um sistema de comunicação em grupo (Newtop) que oferece entrega de mensagens em ordem total e controle consistente de composição de grupo. Cenários simplificados são usados para ilustrar o potencial da abordagem.

Abstract. Although a mature technology and with known cases of deployment success, group communication has not been extensively used in the Internet. One of the reasons for such is the perception that group communication systems exhibit communication and synchronization requirements that are incompatible with Internet large-scale scenarios. Although experimental evaluation of these systems in the Internet offers safer results, it is difficut to be carried out (due to logistics) and its results are hard to be reproduced elsewhere. (When IP multicast is needed, experiments are even more difficult to setup.) Simulation, in this case, may be used, because it allows the investigation of the target system under arbitrary scenarios, with and without failures.

This paper proposes the use of a network simulator for the investigation of group-based distributed systems, allowing a researcher to evaluate the general behaviour and performance of protocols. The proposed methodology is presented through a practical case: the modeling and simulation of a group communication system (Newtop) which offers total order delivery of messages and consistent group membership control. A small set of scenarios are used to illustrate the power of this approach.

Palavras-chave: Multicast, Aspectos de desempenho, escalabilidade e confiabilidade, Tolerância a falhas.

1. Introdução

Sistemas de comunicação em grupo são uma ferramenta importante na construção de sistemas distribuídos, oferecendo ao projetista poderosas abstrações para transmissão ordenada de mensagens e controle de composição de grupo. A implementação desses sistemas pode utilizar um mecanismo de *transporte confiável* multicast, tais como os protocolos sendo padronizados pelo IETF ([1]), ou operar diretamente sobre IP multicast.

Durante duas décadas, a tecnologia de sistemas de grupo evoluiu e amadureceu, resultando no projeto e desenvolvimento de diversos protocolos e sistemas ([2, 3]); alguns destes foram implementados e avaliados em configurações de rede limitadas. Mais recentemente, novos protocolos de comunicação em grupo tem sido projetados e avaliados em função das características da Internet; exemplos são o Spinglass ([4]), Spread ([3]) e o InterGroup ([5]).

Entretanto, supreendentemente, apesar de todos os protocolos de grupo que foram projetados e validados, sua aplicação tem sido restrita a redes "pequenas", usualmente compostas por um conjunto restrito de nodos próximos. Existem diversas razões para isso, mas aqui hipotetiza-se que o principal obstáculo para o emprego de sistemas de comunicação em grupo em larga escala na Internet seja o receio da inadequação de protocolos de comunicação em grupo para tais redes e seus protocolos de transporte (TCP, UDP). É em geral aceito que esquemas de comunicação em grupo exibem requisitos de sincronização e comunicação inaceitáveis em cenários em larga escala típicos da Internet, tanto em termos de tráfego de rede como em latência para a aplicação. Estas limitações de comunicação em grupo estão sendo revistas ([6]), sendo o assunto rediscutido à luz das novas tecnologias presentes na Internet, com maior largura de banda e menor latência (redes ópticas), emergência de aplicações *peer-to-peer* ([7]) e estratégias como *overlay* multicast ([8]).

Apesar do interesse relativamente antigo em escalabilidade de sistemas de comunicação em grupo ([9]), existem poucos dados utilizáveis para estimação de sua viabilidade na Internet, particularmente considerando combinações específicas de configurações de rede e aplicações. Seria bastante desejável dispor de uma ferramenta que possibilitasse a investigação de comunicação em grupo em um ambiente de rede típicos da Internet. *Avaliação analítica* permite expressar matematicamente as propriedades fundamentais de um protocolo e seus mecanismos, mas está limitada a observações de cunho abstrato. Outra solução potencial é a *avaliação experimental*: implementação de uma versão do protocolo de grupo, sua instalação, configuração e execução de experimentos práticos. Assumindo-se que procedimentos apropriados sejam seguidos, seus resultados são precisos (para a configuração empregada). No entanto, esse tipo de abordagem é limitada pelos fatores práticos, pois demanda grande investimento logístico, e seus dados resultam específicos das configurações avaliadas. A coleta e interpretação dos resultados pode ser complicada também, por tratar-se de um sistema distribuído. Quando estes sistemas implementam tolerância a falhas[1], é necessário avaliá-los igualmente em cenários com defeitos. Com experimentos práticos em um cenário distribuído, é difícil reproduzir as situações desejadas. Isto é particularmente verdadeiro para as situações mais complexas de concorrência e/ou envolvendo temporizadores[2].

Neste trabalho demonstra-se o uso de simulação como ferramenta de avaliação de sistemas de comunicação em grupo em cenários de larga escala, tipicamente encontrados na Internet. A metodologia proposta é apresentada através de um caso prático: a modelagem no simulador de rede VINT ns-2 ([10]) de um sistema de comunicação em grupo que oferece entrega de mensagens em ordem total, apropriada aos sistemas com replicação ativa, e controle consistente de composição de grupo. Este protocolo é o Newtop ([11]); o mesmo possui características típicas de sistemas de comunicação em grupo, como um esquema de transmissão periódica de mensagens, necessário aos mecanismos de entrega ordenada de mensagens e de suspeita de defeitos. Outro protocolo poderia ter sido escolhido; a justificativa para o Newtop residiu no conhecimento prévio do mesmo e o contato com os autores.

[1] os termos *falhas* e *defeitos* são utilizados neste artigo como sinônimos de *faults* e *failures*.

[2] empiricamente, verifica-se que esses problemas tendem a ocorrer em sistemas reais, exceto quando se deseja que eles ocorram.

A contribuição do trabalho é apresentar a modelagem de um protocolo de comunicação em grupo (que captura as propriedades típicas destes) em um simulador de redes, e mostrar o potencial dessa abordagem. A avaliação de comunicação em grupo pode ser então conduzida de forma controlada em configurações adequadas à realidade atual, com simulações que consideram topologias e protocolos de transporte e roteamento encontrados na Internet.

O restante do artigo está organizado como segue. A Seção 2 define e contrasta simulação de redes de computadores com simulação de sistemas distribuídos. A Seção 3 se baseia nestes conceitos para apresentar um modelo de simulação de um sistema de comunicação em grupo (Newtop) projetado sobre um simulador de redes, o ns-2. A Seção 4 instancia o modelo em função de uma aplicação de comunicação em grupo, um bancos de dados replicado, demonstrando a variedade de experimentos que pode ser realizada e o potencial da abordagem proposta neste trabalho. A Seção 5 encerra o artigo com conclusões e trabalhos futuros.

2. Simulação de Redes e de Sistemas Distribuídos baseados em Grupo

Simulação de redes é uma prática popular no dimensionamento de redes de computadores, e tem sido amplamente usada no projeto de novos protocolos de comunicação ou avaliação de desempenho de protocolos existentes em novos cenários de rede. Exemplos de simuladores de redes são OPNET ([12]), VINT ns-2 ([10]), SSF ([13]) e Simmcast ([14]). Simuladores oferecem um ambiente controlado para validar o comportamento de protocolos existentes, fornecem infra-estrutura para desenvolvimento de novos protocolos e oportunizam o estudo de suas interações.

Estas facilidades seriam importantes também para sistemas distribuídos, particularmente aqueles com quesitos de dependabilidade. Simulação complementaria as fases de formalização e validação dos protocolos distribuídos. No entanto, supreendentemente, ela não tem sido extensivamente usada como ferramenta auxiliar no projeto de sistemas distribuídos. São raros os exemplos de simuladores de sistemas distribuídos na literatura ([15, 16]).

Para simulação de sistemas distribuídos, é fundamental que os níveis subjacentes ao protocolo alvo sejam "emulados" de forma completa e precisa. Ou seja, o simulador deve oferecer ao usuário implementações simplificadas porém funcionais dos protocolos da camada de transporte (particularmente TCP e UDP), incluindo a transferência de dados, o controle de sessão e o de fluxo.

A utilização de modelos de simulação como ferramenta para avaliação de protocolos de comunicação em grupo, apesar de suas potencialidades, ainda é pouco explorada. Muitas vezes, a sua utilização se restringe ao desenvolvimento de simuladores específicos para o estudo de algum protocolo, como em [17], [18] e [19]. Existem também simuladores genéricos desenvolvidos para estudar aspectos específicos de sistemas de comunicação em grupo. O SimUTC ([15]) visa simular algoritmos distribuídos de sincronização de relógios. Possui como proposta o uso de código comum entre simulação e experimentação real. O CESIUM - *Centralized Distributed-Execution Simulator with Failure Modeling* ([16]) oferece um ambiente orientado a objetos para teste de implementações de protocolos distribuídos tolerantes a falhas. Seu foco é em sistemas de tempo-real, cujo processo de testes em ambientes reais é em especial complexo. Seguindo uma linha mais genérica, o Simmcast ([14]) é um *framework* de simulação para o desenvolvimento de protocolos de rede e distribuídos em Java; sua proposta é permitir a simulação de sistemas distribuídos com a mesma facilidade que a simulação de protocolos de rede. No entanto, o Simmcast ainda não possui suporte com-

pleto aos protocolos em nível de rede e de transporte necessários, em especial, uma versão funcional de TCP. Embora não aborde comunicação em grupo, o trabalho mais próximo a este é [20], onde os autores sugerem o uso do ns-2 como simulador de sistemas distribuídos, e descrevem experimentos com algoritmos distribuídos em cenários com defeitos.

Vislumbra-se duas abordagens básicas para simulação de sistemas distribuídos: o desenvolvimento de uma nova ferramenta ou a adaptação de um simulador de redes existente. No primeiro caso, seria necessário despender um tempo considerável no projeto de um novo simulador; por outro lado, este simulador seria melhor "moldado" ao problema em questão. No segundo caso (aproveitar um simulador de redes), o suporte aos principais protocolos de rede e transporte está usualmente presente. Entretanto, existem outras características desejáveis para um simulador de sistema distribuído, não necessariamente satisfeitas. A seguir, analisa-se os diversos tipos de experimentos ou características consideradas desejáveis na simulação de sistemas distribuídos em grupo na Internet.

1. Gama de topologias de rede, de redes locais (barramento, comutadas) a redes de longo alcance, permitindo um número potencialmente grande de nodos (centenas) com diferentes organizações (p.ex., em Sistemas Autônomos) e uma variedade de larguras de banda e de latências.
2. Emulação de protocolos da camada de transporte e de rede; no primeiro caso, com transporte de dados entre dois pares via datagramas UDP ou fluxo de bytes TCP, sujeitos a controle de fluxo, de congestionamento e de sessão; no segundo caso; disponibilidade de versões simplificadas de protocolos de roteamento unicast e multicast, além de IGMP (*Internet Group Membership Protocol*).
3. Modelagem de nodos que executam a lógica do protocolo distribuído podendo ter diferentes velocidades de execução, diferentes tempos de ativação e relógios físicos independentes.
4. Modelagem de defeitos de nodo e de comunicação; no primeiro caso, incluindo defeitos de colapso (com ou sem recuperação) e bizantinas (permitindo ao usuário modelar um comportamento arbitrário para o nodo)*;* no segundo*,* incluindo reordenamento de mensagens (devido ao roteamento com múltiplos caminhos, potencialmente assimétrico), descarte de parte das mensagens (tipicamente devido a congestionamento na rede), falhas de comunicação temporais (longo período em que todas as mensagens são descartadas) devido a mudanças no roteamento, etc. Modelagem de particionamento temporário ou permanente da rede, quando um enlace defeituoso é o único elo de ligação entre dois segmentos da rede.
5. Facilidades para depuração de protocolos, incluindo visualização pós-experimento ou interativa durante experimento.

Os fatores acima justificam a escolha da abordagem proposta neste trabalho: a customização de um simulador de rede para investigação de sistemas distribuídos baseados em grupo em configurações típicas da Internet. Foram identificados diversos candidatos, tendo sido escolhido (inicialmente) o simulador ns-2 devido à sua popularidade e à quantidade de recursos oferecidos (tecnologias e protocolos implementados). Apesar disso, nota-se que o ns-2 apresenta limitações em relação ao tipo de simulação pretendida; por exemplo, adota o modelo baseado em eventos, forçando o protocolo a ser especificado dessa forma; um modelo baseado em múltiplas *threads* simples seria preferido como abordagem por muitos pesquisadores ao lidar com concorrência em programas.

As características acima apresentadas são abordadas com maior clareza nas próximas seções, através da modelagem do Newtop no ns-2 e de uma aplicação de comunicação grupo sobre o Newtop, e a experimentação com o mesmo em diversos cenários de rede.

3. Modelagem e Simulação do Newtop

3.1. Protocolo Newtop

O Newtop é um sistema de comunicação em grupo que possui mecanismos de ordenamento total, controle consistente de grupo e atomicidade na troca de mensagens ([21, 11]). Ele permite múltiplos grupos sobrepostos, e opera sobre uma rede *assíncrona* (nada se assume sobre o tempo de propagação das mensagens). A camada de rede pode sofrer um particionamento, sendo porém preservada a funcionalidade de comunicação entre os membros que permaneceram na mesma partição.

O ordenamento total é implementado através de relógios lógicos como definido em [22]. Através deste mecanismo, o Newtop detecta quais mensagens recebidas são *entregáveis*. Além disso, o ordenamento total é apoiado pelo controle de atomicidade, o qual garante que qualquer mensagem entregável já tenha sido recebida por todos os membros do grupo. Para garantir a atomicidade, e apoiar a identificação de defeitos, cada membro implementa um mecanismo de *timesilence*. Este mecanismo consiste no envio de uma mensagem de controle, sem dados (*mensagem nula*) se e somente se nenhuma mensagem com dados tenha sido enviada após um intervalo de tempo fixo pré-determinado. Considerações sobre o ajuste desse intervalo aparecem na Seção 4.

O controle de membros é implementado através de *visões do grupo*. Cada membro possui uma visão do grupo que é atualizada sempre que se detecte/suspeite a sua alteração. O serviço de controle de grupo possui uma visão consistente da presença de partições, permitindo que um grupo de entidades seja dividido em dois ou mais subgrupos de membros conectados. O *suspeitor de defeitos* é responsável pela retirada de um membro do grupo. Similarmente ao mecanismo de *timesilence*, o suspeitor de defeitos se baseia em um temporizador com intervalo de tempo fixo pré-determinado. Cada membro monitora a atividade dos demais e quando detecta inatividade por um intervalo de tempo superior ao pré-estabelecido, ocorre uma suspeita. A remoção de um membro somente é realizada se a suspeita for confirmada por todos os membros. Caso haja concordância, o membro cujo defeito foi confirmado é excluído e uma nova visão do grupo é instalada.

As características acima estão presentes nas implementações do Newtop: existem dois protótipos implementados ([23, 21]) ambos baseados na arquitetura CORBA. Embora essas implementações tenham demonstrado a adequação do protocolo a uma série de aplicações e oferecido uma visão inicial sobre seu desempenho, elas não permitiram experimentos mais extensos ou detalhados sob diferentes cargas de rede e/ou defeitos. Conforme citado na seção anterior, esta é uma limitação intrínseca às avaliações experimentais, baseadas em implementação: devido à falta de controle sobre o ambiente, não é possível analisar o comportamento do protocolo em configurações arbitrárias. Como exemplos de cenários, é possível citar topologias com grande número de nodos, enlaces ou servidores com latências arbitrariamente longas e situações de falha das mais diversas, das mais simples às intrincadas ou raras.

O restante desta seção apresenta a modelagem do Newtop no ns-2, descrevendo a estrutura implementada e a dinâmica do seu funcionamento.

3.2. Estrutura implementada[3]

O Newtop assume a existência de uma camada subjacente de transporte multicast para disseminação 1-N "confiável". A arquitetura do modelo no ns-2 reflete a existência dessas

[3] o código fonte relativo aos protocolos implementados e alterações no ns-2 pode ser obtido contactando-se os autores.

duas camadas, conforme explicado a seguir (pressupõe o conhecimento do funcionamento interno do ns-2; vide [10]).

Camada de Grupo Newtop

O Newtop foi implementado como um novo processo no simulador, estendendo a classe abstrata `Process` (oferecida pelo ns-2 para modelar uma entidade capaz de processar uma *Application Data Unit,* ou ADU). A seguir, as principais classes utilizadas na implementação do protocolo são brevemente comentadas.

A classe `NewTopGroup` é utilizada para definir um grupo Newtop, onde cada grupo possui um identificador e uma lista de membros pertencentes. Um *agente Newtop* utiliza a primitiva `join()` para se juntar a um grupo. As mensagens envolvidas na comunicação entre os membros de grupos Newtop são implementadas pela classe `NewtopMessage`, estendida de `AppData`; esta última é necessária para o processamento das ADUs. Para isso, foi definido um novo tipo de dados na estrutura `AppDataType`, `NEWTOP_DATA`. A classe `NewtopMessage` define o formato da mensagem que será trocada entre os membros do grupo Newtop: `MT_DATA`: mensagens de dados enviadas pela aplicação; `MT_TIMESILENCE`: mensagem nula, utilizada para estabilizar as mensagens recebidas; `MT_SUSPECT`: mensagem que contém informações sobre o suspeitor de defeitos; `MT_REFUTE`: mensagem que recusa a suspeita de defeito de um membro; `MT_CONFIRMED`: mensagem que confirma o defeito de um membro; e, `MT_REMOVE`: mensagem que confirma remoção e instala uma nova visão.

A estrutura principal do protocolo é implementada pela classe `Newtop`. Ela deve ser instanciada para cada nodo participante de algum grupo, independente da quantidade de grupos que este nodo participa. Esta classe oferece a interação com o usuário e oferece os seguintes métodos: (a) `join`: inserção do nodo em um determinado grupo Newtop; (b) `msend`: envio de mensagens para um determinado grupo; (c) `setTimeSilence`: configuração do temporizador que rege o envio de mensagens nulas; e, (d) `setSuspectorInterval`: configuração do temporizador que é utilizado pelo suspeitor de defeitos.

Camada de Transporte Multicast Confiável

O Newtop pressupõe a existência de um camada de transporte "confiável" que implementa a transmissão 1-N de mensagens com semântica similar ao TCP. Esta camada pode ser implementada seguindo diferentes estratégias, com ou sem uso de IP multicast. Naturalmente, o desempenho do sistema de comunicação em grupo depende do protocolo de transporte utilizado (conforme demonstrado na Seção 4). Poucos protocolos de transporte multicast oferecem as garantias exigidas nesse caso (a maioria utiliza *erasure codes*/FEC ou confirmação negativa de recebimento). Exemplos de protocolos que oferecem confirmação positiva incluem PRMP ([24]), suas variantes baseadas em *polling* ([25]) e TCP-SMO ([26]).

Realizou-se a modelagem de forma a possibilitar a transição transparente (do ponto de vista do agente Newtop) da camada de transporte multicast. Isto se deu através da prototipação de uma classe abstrata `ProtocolStrategy`, tendo como subclasses `MultiTCPStrategy` e `MulticastStrategy`, explicadas a seguir.

Para a modelagem do `MultiTCPStrategy`, onde a comunicação entre os membros de um grupo Newtop se dá através de *streams* TCP, utilizou-se uma malha de comunicação com $(N-1) \times (N-1) \div 2$ conexões TCP bidirecionais (onde N é o número de integrantes

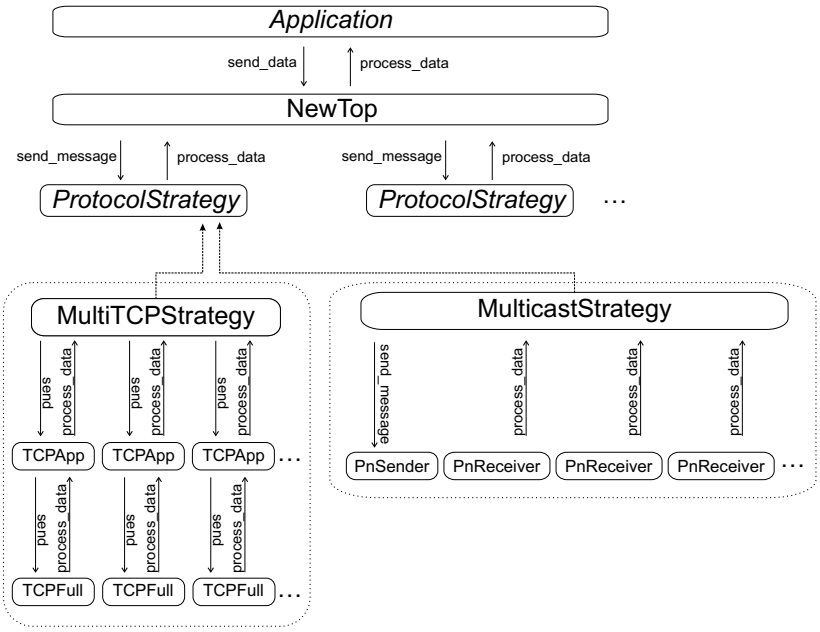


Figura 1: Dinâmica de funcionamento

do grupo), estabelecendo conexões de transporte entre todos integrantes. Portanto, uma mensagem é enviada aos integrantes do grupo via $N-1$ transmissões da mensagem pelos canais TCP. A implementação desta camada usa em especial duas classes do ns-2: `TCPApp` e `TCPFull`: a primeira funciona como um invólucro para o envio dos dados (`AppData`) via TCP, enquanto a segunda implementa a camada de transporte TCP, incluindo controle de congestionamento e de erro.

No caso da classe `MulticastStrategy`, onde existe uma camada de transporte multicast confiável, a distribuição para todos os membros do grupo se dá de forma automática havendo uma relação 1-1 entre grupo Newtop e grupo IP multicast. O transporte multicast deve apresentar semântica TCP estendida para comunicação unidirecional 1-N. A maioria dos protocolos de transporte confiável, tais como instâncias do protocolo NORM ([1]), são altamente escaláveis, mas em detrimento de tais garantias. Neste trabalho, foram utilizados os protocolos baseados em *polling* apresentados em [25]. Estes protocolos demonstraram possuir um comportamento adequado e eficiente no que diz respeito a transmissões multicast confiável para grupos de média escala (tendo sido realizadas simulações exaustivas com grupos de até 450 receptores). Em [27], foi apresentado um modelo de controle de congestionamento (com suporte a ECN - *Explicit Congestion Notification*) amigável ao TCP para estes protocolos, tornando-os aptos a serem utilizados na Internet.

A seguir, é comentada a interação entre as camadas acima.

3.3. Dinâmica de funcionamento

A Figura 1 ilustra a hierarquia e a comunicação entre os principais componentes do modelo implementado. Um agente Newtop que faz parte de G grupos possui G instâncias de `ProtocolStrategy`, uma para cada grupo. Quando da instanciação, deve-se optar por uma `ProtocolStrategy` específica (atualmente, `MultiTCPStrategy` ou `MulticastStrategy`). Associado ao `MultiTCPStrategy`, existem $N-1$ instâncias de `TCPApp` e `TCPFull`, cada uma conectada a um dos $N-1$ demais componentes do grupo relativo a este `MultiTCPStrategy`. Já no caso do `MulticastStrategy`, existe um transmissor multicast (conectado aos receptores dos demais componentes do grupo) e $N-1$ receptores (conectados aos transmissores).

A comunicação entre o agente de grupo Newtop e a camada de transporte multicast subjacente se dá através dos métodos `send_message` e `process_data`. Para enviar uma mensagem para um grupo g, o agente Newtop determina qual a instância de `ProtocolStrategy` a ser acionada e chama o respectivo método `send_message`. Caso esta instância seja um `MultiTCPStrategy`, a mensagem será diretamente repassada para cada um dos $N - 1$ `TCPApps` (ligados a agentes `TCPFull`), que encapsula questões de fragmentação e retransmissão. Já no caso de uma instância de `MulticastStrategy`, o transmissor multicast efetua a fragmentação da mensagem (baseado em um tamanho máximo pré-estabelecido), envia os pacotes utilizando IP multicast, configura temporizadores, etc. segundo os protocolos descritos em [25].

O recebimento de uma mensagem em um nodo percorre o caminho contrário. Em ambas as implementações da camada de transporte, mensagens são encaminhadas para a camada Newtop obedecendo a uma ordem FIFO, ou seja, dado um par transmissor-receptor, todas as mensagens são repassadas ao Newtop na ordem de saída do transmissor. A instância de `ProtocolStrategy` atua apenas como um invólucro, repassando as mensagens consolidadas diretamente para o Newtop.

4. Estudo de Caso

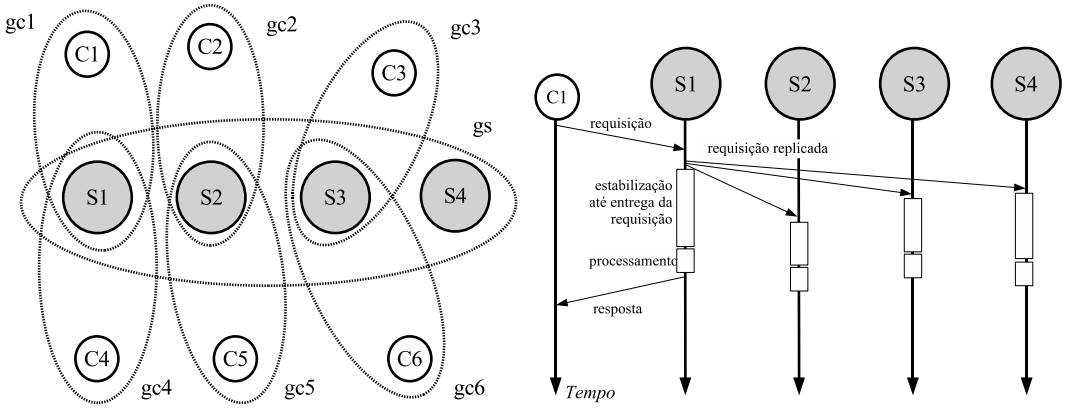
Na Seção 2 foram indicadas várias características desejáveis na simulação de sistemas distribuídos baseados em grupo, enquanto na Seção 3 foi descrita a versão do Newtop modelada sobre o simulador de rede ns-2. O modelo acima descrito é exercitado através de um conjunto de experimentos cujo objetivo não é determinar resultados de desempenho específicos para dadas configurações e parâmetros de entrada do Newtop, mas sim demonstrar o poder da abordagem proposta.

Para tal, uma aplicação de comunicação em grupo é simulada sobre um conjunto de topologias. O restante da seção está organizada em função disso: descreve-se a aplicação modelada, as topologias escolhidas sobre qual roda essa aplicação, o processo de avaliação dos mecanismos básicos do protocolo de grupo, e a avaliação do impacto do mecanismo de transporte utilizado.

4.1. Aplicação distribuída

A aplicação modelada para o estudo de caso simula um banco de dados replicado com organização cliente/servidor. Servidores implementam tolerância a falhas através de replicação ativa: cada servidor mantém uma réplica exata dos dados e sobre os quais executa o mesmo conjunto de operações. Para manter a consistência dos dados, mensagens são entregues aos servidores em ordem total. O conjunto de S servidores compõe um grupo, gs, que é sobreposto com C grupos gc_i contendo apenas um cliente e um servidor, onde C é o número de clientes. Na Figura 2.(a), um exemplo é dado para $S = 4$ e $C = 6$.

Dois tipos de mensagens são trocadas entre os clientes e servidores: *requisição* e *resposta*. Conforme ilustrado na Figura 2.(b), cada cliente envia requisições para o seu grupo cliente e aguarda uma resposta. Quando uma requisição de um cliente é entregue a um servidor, o mesmo re-envia a mensagem (*requisição replicada*) ao grupo de servidores. A requisição replicada é processada pelo Newtop e, quando a mesma se torna *estável*, é entregue em cada um dos servidores para que seja processada (o *tempo de estabilização* é representado na Figura 2.(b) pelos retângulos superiores). Após o processamento da requisição replicada (retângulos inferiores), o servidor que recebeu a requisição do cliente envia uma resposta ao grupo do cliente correspondente.



(a) Grupos, clientes e servidores (b) Diagrama para uma requisição/resposta

Figura 2: Exemplo de configuração para a aplicação banco de dados replicado.

A aplicação é completamente parametrizável, e especificada fundamentalmente em termos de número de clientes e de servidores, da composição dos grupos, e da localização na rede de cada um dos processos (vide a seguir). É possível configurar valores de entrada como tamanho das mensagens de requisição e resposta; número de requisições a ser gerado por cada cliente; modelo de geração de carga (parâmetros de um modelo *on-off*); além de parâmetros que definem o tempo médio de processamento de uma requisição em um servidor, após a entrega da mesma.

4.2. Topologias de rede

A especificação da topologia é a principal entrada de um simulador de rede. A quantidade de informação varia entre simuladores, mas o denominador comum é o conjunto de nodos, as ligações entre os mesmos, e os atributos básicos destes elementos. No caso do ns-2, uma topologia é especificada através do arquivo *tcl*; não há suporte à geração automática de topologias, mas é possível importar uma topologia produzida por um gerador automático como o gt-itm ([28]) ou Brite ([29]).

No estudo de protocolos de comunicação em grupo, é desejável a utilização de topologias em diferentes níveis de abstração: desde modelos abstratos (anel, hipercubo, etc), sem modelagem de elementos internos da rede, até modelos mais realísticos, onde elementos como roteadores podem ser configurados individualmente. O ns-2 oferece facilidades principalmente no que diz respeito a topologias com baixo nível de abstração, cabendo ao usuário mapear os elementos da rede a ser simulada. Para este estudo, foram modeladas três tipos de topologias, conforme a seguir.

- **Rede local:** implementa um rede simples onde um conjunto de máquinas (p.ex., 20) executa instâncias do aplicativo cliente, um por máquina; máquinas estão conectadas fisicamente a um mesmo *switch*, e um pequeno número de servidores (p.ex., 4) a outro *switch*; ambos *switches* estão ligados entre si, sendo o segundo deles ligado a um roteador de borda.
- **Múltiplos Sistemas Autônomos:** a rede local acima é encarada como um Sistema Autônomo (SA) e instanciada múltiplas vezes, interligando os sistemas via *domínio de trânsito central*. A interligação é realizada pelos roteadores de borda de cada SA, através de enlaces com menor largura de banda e maior latência. A configuração total

da topologia comporta dezenas de servidores e um número bem maior de clientes, uniformemente distribuídos entre os SAs.

- **Sistema Autônomo com Múltiplas Redes:** a topologia teve como base a estrutura encontrada em uma dada universidade. É formada por uma dúzia de redes locais identificadas por "Centros", interligadas por um roteador central; cada Centro possui até 3 *switches* interligados em cascata com até 20 clientes conectados em cada um. Além disso, os Centros possuem um roteador que está conectado ao roteador central. O objetivo da topologia é mostrar o impacto do desempenho sob condições com grande número de clientes e servidores, mas abundante largura de banda.

Estas topologias foram escolhidas de forma a representar uma variedade de cenários; naturalmente, não existe um único cenário que seja representativo da Internet. Em cada uma delas há propriedades que podem ser ajustadas de acordo com o objetivo do experimento. Destacam-se os atributos dos enlaces: latência de propagação, largura de banda, percentagem de erro (descarte de pacotes) e tamanho das filas de pacotes. Além disso, fluxos de pacotes podem ser adicionados de forma a simular a existência de outras aplicações e induzir não-determinismo na comunicação.

Estas topologias (e seus atributos de rede) podem ser ajustadas de forma a criar cenários de rede específicos e avaliar seu impacto nos sistemas de grupo. Por exemplo, variando-se o tráfego de fundo ou banda total em um enlace compartilhado, é possível induzir uma situação onde pacotes são atrasados, ou *jittering* é forçado; estes podem afetar decisivamente o funcionamento dos mecanismos de temporização tipicamente encontrados nos protocolos de grupo. Além disso, em topologias em geral, a redundância de enlaces e o uso de enlaces unidirecionais geram assimetrias no roteamento e podem ser empregados para simular reordenamento de pacotes. Experimentos podem ser compostos para medir o impacto nos mecanismos de detecção de defeitos dos protocolos de grupo: por exemplo, um membro suspeita da ocorrência de defeito em outro, mas a recíproca não é verdadeira. Por fim, é possível modelar a ocorrência de defeito temporário ou permanente de um nodo ou elemento de rede (enlace, roteador, *switch*...). A seguir, um pequeno conjunto de experimentos é oferecido como exemplo.

4.3. Exemplos de experimentos

Usualmente existem diversos parâmetros a serem ajustados/testados em um sistema de comunicação em grupo. O número deles e seu caráter variará de acordo com o protocolo em questão; freqüentemente, são parâmetros que representam intervalos de temporização. No caso do Newtop (vide Seção 2), os parâmetros fundamentais presentes no modelo simulado são *timesilence* e *suspector* (outros protocolos de grupo possuem variáveis similares, com outros nomes). Estes valores são importantes para o bom funcionamento do protocolo, afetando a *precisão* e a *eficiência* do mecanismo de suspeita de defeitos. Precisão consiste em remover de um grupo apenas membros que tenham sido vítimas de um defeito ou que tenham sido isolados em uma outra partição, enquanto eficiência é o inverso da sobrecarga de mensagens e atrasos provocados pelo mecanismo. Note-se que não existe um conjunto único de parâmetros que seja bom para todas as configurações de rede e aplicação, justificando a especialização do modelo de simulação segundo as características de rede e aplicação presentes.

No estudo de caso, o espaço de parâmetros do protocolo de grupo foi investigado considerando uma série de configurações de rede, três das quais incluídas neste artigo (descritas anteriormente). O protocolo foi avaliado sob diversos aspectos, incluindo demanda de largura de banda e tempo médio entre requisição e resposta (no cliente). A título de

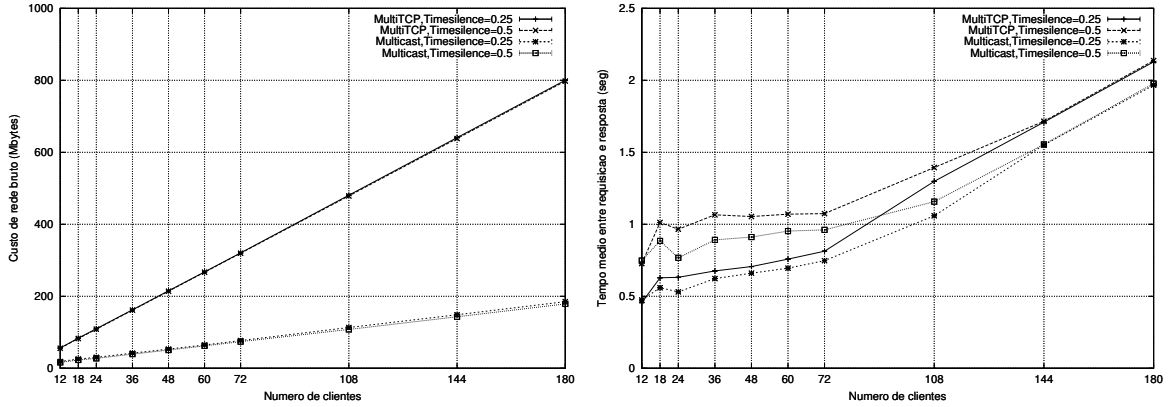


Figura 3: Exemplo de avaliação - impacto do número de clientes no tráfego de rede e no tempo de requisição e resposta

exemplo[4], são apresentados nas Figuras 3.(a) e 3.(b) resultados demonstrando o impacto do número de clientes no tráfego de rede e no tempo entre requisição e resposta, para a topologia Sistema Autônomo com Múltiplas Redes, fixando-se em 8 o número de servidores no grupo. Observa-se na Figura 3.(a), que o uso de um transporte baseado em IP multicast (ao invés de múltiplos TCPs) aumenta a escalabilidade global, trazendo uma importante redução no custo de rede. A Figura 3.(b) mostra que, para um número pequeno de clientes, a variação do valor de *timesilence* utilizado pelo grupo de servidores traz uma diferença considerável no tempo entre requisição e resposta percebido pelos clientes. À medida que o número de clientes aumenta, este aspecto torna-se menos importante, enquanto o protocolo de transporte utilizado na camada subjacente passa a exercer um papel mais importante, posto que consegue reduzir o volume de tráfego gerado.

Um modelo de simulação de um sistema de comunicação em grupo permite investigar cenários com defeitos na rede e em nodos. No estudo de caso, foram realizados experimentos para determinar o tempo até alcançar concordância após defeito de colapso de um membro, e tempo até alcançar concordância após particionamento de rede, para diferentes valores de intervalo *suspector*. Em ambos os experimentos, fixou-se em 80 o número de clientes conectados aos servidores, bem como o uso de *streams* TCP (`MultiTCPStrategy`) como forma de comunicação na camada de transporte. No primeiro caso, nodos são dinamicamente removido da simulação durante o experimento; todas as mensagens que são enviadas ao nodo são perdidas, e naturalmente o mesmo não envia mais mensagens. Conforme pode ser notado no primeiro gráfico da Figura4, a medida que o intervalo cresce, aumenta o tempo necessário para retirada do nodo defeituoso do grupo.

No segundo caso, particionamento, o mesmo é provocado via defeito de colapso do enlace que conecta um dos SAs ao domínio de trânsito central; todas as mensagens que sairiam ou chegariam ao SA particionado são perdidas. Isto faz com que o grupo original com 8 servidores seja particionado em dois grupos: um com 6 e um com 2 elementos. O segundo gráfico da Figura 4 apresenta o tempo para consenso tanto no grupo menor quanto no maior, bem como a variação resultante do uso de um *timesilence* de 0,25 e 0,5 segundo. Os resultados revelam, à exemplo do gráfico anterior, que o intervalo do *suspector* tem influência significativa no tempo para consenso.

Estes são exemplos simples de experimentos que podem ser conduzidos com o mo-

[4]Para acesso ao conjunto completo de resultados e análises, incluindo parâmetros de entrada utilizados nos experimentos, vide [30]

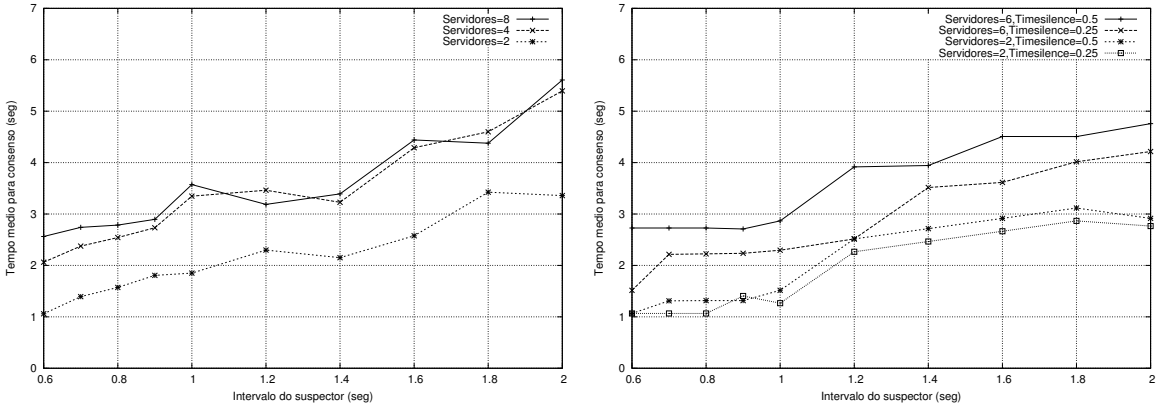


Figura 4: Exemplo de avaliação - impacto do intervalo *suspector* no tempo para remoção de nodo com defeito de colapso, e consenso nos grupos após partição

delo. Extensões diretas seriam a avaliação do impacto de reordenamento de pacotes; da ausência temporária de alcançabilidade, devido ao colapso de um enlace seguido de uma mudança de rota (tal fenômeno pode fazer com que todas as mensagens sejam perdidas durante alguns segundos); e do atraso substancial de um conjunto sucessivo de mensagens, por exemplo, devido a um recálculo de rotas em um roteador.

5. Conclusão

Este trabalho demonstrou o uso de simulação discreta como ferramenta de investigação de sistemas de comunicação em grupo em cenários típicos da Internet. A abordagem proposta emprega um simulador de rede, ns-2, que suporta topologias com elementos de rede, múltiplas rotas, protocolos de rede e de transporte. O sistema de grupo Newtop foi modelado neste simulador, e um estudo de caso realizado com o mesmo. O estudo de caso emprega uma aplicação de bancos de dados replicados. Essa combinação de protocolo de grupo e aplicação foi avaliada em diversas combinações de rede e topologias, com o objetivo de determinar a variedade de experimentos que poderia ser realizada, e a utilidade dos resultados obtidos. Os resultados confirmaram expectativas iniciais, e ao mesmo mostraram fenômenos interessantes (o comportamento do protocolo com e sem multicast, mais resultados de desempenho e escalabilidade serão abordados em um outro trabalho). Produziu-se um modelo de avaliação para o Newtop e protocolos de grupo similares, e comprovou-se que esta é uma abordagem apropriada para investigação de protocolos de grupo.

Como metodologia, simulação é inerentemente mais fácil do que a experimentação, porque permite a abstração de aspectos que não são tão relevantes ao alvo de estudo, ao mesmo tempo que mantém a dinamicidade de uma comunicação envolvendo diversos protocolos. Por outro lado, essa abordagem ainda requer um esforço de projeto e implementação significativo, pois o(s) protocolo(s) e aplicações precisam ser modelados no simulador. Além disso, é necessário projetar cenários de teste que sejam adequados e que explorem a amplitude de possíveis cenários/situações.

Uma das limitações da proposta é a dificuldade em generalizar a metodologia. Neste trabalho, foram feitas três escolhas: um simulador (ns-2), um protocolo de comunicação em grupo (Newtop), e uma categoria de aplicação distribuída baseada em grupo (banco de dados replicado). Portanto, o trabalho poderia ser estendido variando-se qualquer uma destas escolhas; naturalmente, existe um esforço de implementação associado a cada nova

variação tentada.

Como trabalhos futuros, considera-se a avaliação de outros sistemas de comunicação em grupo, particularmente aqueles sistemas recentes cuja arquitetura autores sustentam ser adequada à Internet (p.ex., Spinglass, Spread, InterGroup). Note-se que a implementação de múltiplos protocolos e a avaliação experimental adequada na Internet com cada um deles seria uma tarefa muito maior. Vislumbra-se também avaliações com outros simuladores, particularmente o Simmcast ([14]), que não possui suporte a tantas tecnologias e protocolos, mas por outro lado possui facilidades mais adequadas ao desenvolvimento de sistemas distribuídos. Por fim, outras aplicações podem ser utilizadas como geradores de carga; o grupo de pesquisa está atualmente usando essa abordagem para investigar a viabilidade de comunicação em grupo em suporte à jogos online na Internet.

Referências

[1] B. Adamson, C. Bormann, M. Handley, and J. Macker. NACK-Oriented Reliable Multicast (NORM) Building Blocks. Internet-Draft, November 2003. http://www.ietf.org/internet-drafts/draft-ietf-rmt-bb-norm-08.txt.

[2] L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, R. K. Budhia, and C. C. Lingley-Papadopoulos. Totem: A Fault-Tolerant Multicast Group Communication System. *Communications of the ACM*, 39(4):54–63, 1996.

[3] Y. Amir and J. Stanton. The spread wide area group communication system. Technical Report CNDS 98-4, 1998.

[4] K. Birman. Scalable Fault-Tolerant Aggregation in Large Process Groups. In *International Conference on Dependable Systems and Networks, DSN*, 2001.

[5] K. Berket, D. A. Agarwal, P. M. Melliar-Smith, and L. E. Moser. Overview of the Inter-Group Protocols. In *Proceedings of the 2001 International Conference on Computational Science*, 2001.

[6] T. Anker, D. Dolev, G. Greenman, and I. Shnayderman. Evaluating Total Order Algorithms in WAN. In *International Workshop on Large-Scale Group Communication*, 5 October 2003. In conjunction with SRDS'2003.

[7] K. Albrecht, R. Arnold, and R. Wattenhofer. Clippee: A Large-Scale Client/Peer System. In *International Workshop on Large-Scale Group Communication*, 5 October 2003. In conjunction with SRDS'2003.

[8] M. Castro, M. B. Jones, A. Kermarrec, A. Rowstron, M. Theimer, H. Wang, and A. Wolman. An Evaluation of Scalable Application-level Multicast Built Using Peer-to-peer Overlays. In *INFOCOM2003*, 2003.

[9] O. Babaoglu, R. Davoli, L. A. Giachini, and M. G. Baker. Relacs: A communication infrastructure for constructing reliable applications in large-scale distributed systems. In IEEE, editor, *28th Hawai Int. Conf. on System Sciences*, pages 612–621, 1995.

[10] The Network Simulator VINT ns-2. http://www.isi.edu/nsnam/ns.

[11] P. Ezhilchelvan, R. Macêdo, and S. Shrivastava. Newtop: A Fault-Tolerant Group Communication Protocol. In *IEEE 15th Intl. Conf. Distributed Computing Systems*, pages 296–306, Vancouver, May 1995.

[12] Opnet technologies. http://www.opnet.com/.

[13] Scalable simulation framework - ssf. http://www.ssfnet.org/.

[14] M. P. Barcellos, H. H. Muhammad, and A. Detsch. Simmcast: a simulation tool for multicast protocol evaluation. In *XIX Simpósio Brasileiro de Redes de Computadores, SBRC 2001*, volume 1, pages 418 – 433, Florianópolis, Brasil, 2001.

[15] B. Weiss, G. Gridling, U. Schmid, and K. Schossmaier. The SimUTC Fault-Tolerant Distributed Systems Simulation Toolkit. In *7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, March 1999.

[16] G. A. Alvarez and F. Cristian. Cesium: Testing hard real-time and dependability properties of distributed protocols. In IEEE, editor, *3rd Workshop on Object-Oriented Real-Time Dependable Systems - (WORDS '97)*, Newport Beach, 1997.

[17] C. Ciarfella, L. Moser, P. Melliar-Smith, and D. Agarwal. The Totem Protocol Development Environment. In *International Conference on Network Protocols, ICNP'94*, 1994.

[18] A. Garyfalos, K. Almeroth, and J. Finney. A Hybrid of Newtork and Application Layer Multicast for Mobile IPv6 Networks. In *International Workshop on Large-Scale Group Communication*, 5 November 2003.

[19] M. J. Monteiro, J. Pereira, and L. Rodrigues. Integration os Flight Simulator 2002 with an epidemic multicast protocol. In *International Workshop on Large-Scale Group Communication*, October 2003.

[20] R. Trindade, M. P. Barcellos, and I. Jansch-Porto. Simulação de sistemas distribuídos em cenários com defeitos. In *III Workshop de Testes e Tolerância a Falhas - WTF 2002*, May 2002.

[21] G. Morgan and P. D. Ezhilchelvan. Policies for using Replica Groups and their effectiveness over the Internet. In *Proceedings of the International Workshop on Networked Group Communication*, Palo Alto, California, USA, November 2000.

[22] L. Lamport. Time, Clocks and the Ordering of Events in a Distributed System. *Comunications of ACM*, 21(7):558–565, jul 1978.

[23] G. Morgan, S.K. Shrivastava, P.D. Ezhilchelvan, and M.C. Little. Design and Implementation of a CORBA Fault-Tolerant Object Group Service. In *2nd International Working Conference on Distributed Applications and Interoperable Systems (DAIS 1999)*, Helsinki, Finland, 1999.

[24] M. P. Barcellos, A. Detsch, H. H. Muhammad, and G. B. Bedin. Efficient TCP-like Multicast Support for Group Communication Systems (SCTF2001). In *IX Brazilian Symposium on Fault-Tolerant Computing (SCTF)*, Florianópolis, 2001.

[25] M. P. Barcellos and A. Detsch. Avaliação de Desempenho de Protocolos Multicast com Conhecimento de Grupo Baseados em Polling. In SBC, editor, *XX Simpósio Brasileiro de Redes de Computadores, SBRC 2002*, volume 1, Búzios, Brazil, 2002.

[26] S. Liang and D. Cheriton. Tcp-smo: Extending tcp to support medium-scale multicast applications. In *INFOCOM*, 2002.

[27] A. Detsch and M. P. Barcellos. Controle de congestionamento com suporte a ecn em protocolos multicast de taxa única. In SBC, editor, *21º Simpósio Brasileiro de Redes de Computadores*, Natal, Brasil, 2003.

[28] Gt-itm: Georgia tech internetwork topology models. http://www.cc.gatech.edu/projects/gtitm/.

[29] Boston university representative internet topology generator. http://www.cs.bu.edu/brite/.

[30] M. C. Diemer. Modelagem e Simulação do Protocolo NewTop para Avaliação de Comunicação em Grupo em Configurações Típicas da Internet. Master's thesis, Universidade do Vale do Rio dos Sinos - UNISINOS, February 2003.