Avaliação Experimental de Protocolos de Multicast Confiável: um *choque de realidade*

Marinho P. Barcellos

[1]PIPCA - Programa Interdisciplinar de Pós-Graduação em Computação Aplicada
Unisinos - Universidade do Vale do Rio dos Sinos
Av. Unisinos, 950 - São Leopoldo, RS - 93022-000

marinho@unisinos.br

Resumo. *Embora o assunto multicast confiável tenha sido amplamente abordado na literatura, e seus benefícios potenciais reconhecidos, as aplicações do mesmo ainda são bastante restritas e pouco se sabe sobre o desempenho real destes protocolos. Historicamente, novos mecanismos e protocolos de multicast confiável têm sido avaliados através de análise e simulação. Ao contrário de trabalhos anteriores, este artigo apresenta resultados derivados de uma extensa avaliação experimental com o estado-da-arte na área, em redes de produção propriamente ditas. Na avaliação, são enfatizadas as aplicações de Grade com transmissão intensiva de dados, para grupos de pequeno porte. São empregados cenários de rede locais e de longo alcance, tanto em taxa fixa como em taxa adaptativa (controle de congestionamento). Os resultados são de certa forma inesperados, representando um "choque de realidade" em relação à pesquisa anterior na área.*

Abstract. *Although reliable multicast has been extensively addressed in the literature, and its benefits widely recognized, its applications are still limited and little is known about their actual performance. Historically, new reliable multicast mechanisms and protocols have been evaluated using analysis and simulation. In contrast to previous work, this paper presents results derived from an extensive experimental evaluation with the state-of-the-art protocol implementations, in production networks. The evaluation focuses on data-intensive Grid Computing applications, which exchange large amounts of data but typically require small groups. The scenarios employed include a Wide-Area Network, using protocols both with and without congestion control (adaptive and fixed rate, respectively). The results are somewhat surprising, and represent a reality shock in terms of previous research in the area.*

1. Introdução

Computação em Grade pode ser caracterizada como uma infra-estrutura distribuída composta por uma coleção de recursos distribuídos atuando como uma entidade única. Neste contexto, aplicações com acesso intensivo a dados (*data-intensive*) demandam a transmissão de grande quantidade de dados entre *sites* geograficamente dispersos. Em certos casos, dados são transferidos de um *site* para vários, via **transmissão multiponto confiável**.

Existem basicamente duas formas de se implementar transmissão multiponto confiável: com ou sem suporte de IP multicast. Esta tecnologia surgiu no início dos anos 90, oferecendo roteamento eficiente de pacotes a grupos de receptores (fracamente acoplados) em aplicações multiponto. **Protocolos de multicast confiável** agregam valor ao IP multicast

através de mecanismos de controle de erro (que detectam e corrigem, ou previnem, perdas de pacotes), de fluxo (que evitam atropelamento de máquinas fim de baixa capacidade ou sobrecarregadas), de congestionamento (que obedecem uma divisão justa de largura de banda) e sessão (que sincronizam transmissor e receptores, e coletam informações globais de estado).

A outra forma é utilizando multicast em nível de aplicação *(Application-Level Multicasting*, ALM), que dispensa IP multicast pois distribui dados usando uma estrutura tipo *overlay*. ALM surgiu devido à histórica falta de suporte a IP multicast por parte dos provedores de Internet, e é uma solução paleativa. Limitações conhecidas são a dificuldade em se construir uma árvore de distribuição eficiente, devido à falta de conhecimento sobre a topologia, e a necessidade de replicar os fluxos dentro nos sistemas-fim, sobrecarregando máquinas e enlaces.

O assunto multicast confiável e escalável já foi extensivamente explorado na literatura, levando à proposta de novos protocolos e mecanismos. Diversos trabalhos tecem comparações entre protocolos; por exemplo, [S. Pingali et al., 1994] e [B. N. Levine and J.J. Garcia-Luna-Aceves, 1998] usam avaliação analítica para comparar modelos de protocolos, enquanto [C. Hänle and M. Hofmann, 1998] emprega simulação para comparar protocolos. [M.W. Koyabe and G. Fairhurst, 2001] faz uma comparação funcional de protocolos multicast confiáveis para comunicação em satélite.

O presente artigo se difere de pesquisa anterior de maneira significativa. Não existe na literatura, ao que tudo indica, uma comparação de **múltiplos** protocolos multicast baseada em **avaliação experimental**, nem que tenha sido conduzida com em uma **rede de produção** (em contraste às metodologias baseadas em avaliação analítica, simulação, ou emulação de cenários de rede com *testbeds*). Neste trabalho, um conjunto de protocolos é comparado experimentalmente em condições reais e idênticas, viabilizando pela primeira vez uma comparação justa e realística.

Adicionalmente, este trabalho difere da pesquisa anterior em multicast confiável por não enfatizar escalabilidade**,** e sim **alto desempenho**. Segundo [V. Sander (Ed.), 2004], a maioria dos usos de multiponto em computação em grade envolverão grupos pequenos, porém estima-se que os ganhos em custo e desempenho oferecidos por multicast sejam significativos mesmo nestes casos. Isto está de acordo com [Chalmers and Almeroth, 2001], que demonstra que a disseminação via árvore multicast nativo oferece ganhos de eficiência significativos em comparação com unicast, com melhorias de 60% para grupos com 20 receptores. Além desse ganho, a transmissão massiva de dados em aplicações com requisitos de desempenho pode gerar um gargalo no computador transmissor ou no enlace de saída desta rede.

O restante do artigo está organizado como segue: na Seção 2, é descrita a aplicação alvo e os requisitos relativos a multicast confiável correspondentes. A Seção 3 apresenta o estado-da-arte em protocolos multicast confiável, à luz do trabalho executado na Internet Engineering Task Force (IETF), e como isso reflete em termos de implementações. A Seção 4 descreve os cenários de avaliação empregados, incluindo ambientes de rede, parâmetros de entrada e métricas. A Seção 5 apresenta os resultados da avaliação e oferece uma análise dos mesmos. A Seção 6 encerra o artigo, com conclusões gerais e projeção de trabalhos futuros.

2. Aplicação Alvo e Requisitos

Aplicações em Grade enxergam a rede como um recurso (chave), cujo acesso se dá através de um "serviço" que deve oferecer alto desempenho, confiabilidade e configurabilidade.

Prevê-se que a demanda sobre tais serviços seja, no futuro próximo, de taxas na ordem de 1 Gbps ou mais [V. Sander (Ed.), 2004], o que só pode ser obtido com uso parcimonioso dos recursos. Multicast confiável é uma tecnologia que otimiza o uso da rede e do tempo de processamento no transmissor, podendo aumentar sensivelmente o desempenho final da comunicação.

Exemplos de aplicações de Grade que podem usufruir de multicast confiável são, segundo [V. Sander (Ed.), 2004], acessos a bases de dados, compartilhamento e replicação (*DataGrid*, *Encyclopedia of Life Science*), mineração de dados distribuída, transferências de dados e código para submissões de *jobs* massivamente paralelos, bem como aplicações colaborativas em *e-Science*. Essas aplicações podem ser encaixadas nos quatro tipos de multicast definidos em [C .Diot et al., 2000]: distribuição de áudio/vídeo e multimídia, aplicações do tipo *push* para disseminação de informações *(messaging),* transferência de arquivos um para vários, e aplicações em grupo colaborativas com troca de áudio/vídeo.

Este trabalho se concentra em aplicações de multicast confiável que demandem o envio de grandes quantidades de dados a um conjunto pequeno de receptores. A seguir, o serviço de transmissão multiponto confiável almejado é melhor descrito, através de seus requisitos e premissas.

Dados podem ser oferecidos pela aplicação ao serviço (protocolo) através de uma *stream* de dados, por meio de um descritor, ou via arquivo(s). Em ambos os casos os dados devem ser integralmente entregues a todos os receptores, desde que os mesmos estejam alcançáveis pela rede e continuem operando durante a sessão (não falhem). Unidades de dados a serem transmitidas, arquivos individuais ou *streams*, variam tipicamente de dezenas de Megabytes a dezenas de Gigabytes.

Um conjunto de máquinas deve receber uma cópia integral dos dados. O transmissor conhece a composição do grupo receptor, cujo tamanho varia tipicamente entre 2 e 20. Os receptores se encontram geograficamente espalhados e conectados por uma rede de longo alcance que propicia, minimamente, conectividade multicast do transmissor para os receptores, e unicast em ambos os sentidos. Não se assume conectividade unicast ou multicast entre receptores, pois na prática isso pode limitar a implantação do protocolo.

Quanto à ativação, assume-se que o transmissor é iniciado com um lista não-vazia de endereços de computadores destinatários (e portas), um endereço multicast (e porta) a ser usado, e uma lista não-vazia de arquivos locais a ser transmitida ou um descritor através do qual o conteúdo de uma *stream* arbitrária é lido e enviado. Assume-se ainda que receptores são iniciados e permanecem em estado de espera (como *daemons*) por um contato do transmissor, estabelecendo a sessão. Se o protocolo multicast não oferece um mecanismo de controle de sessão, então uma camada externa deve ser usada de forma a sincronizar (e potencialmente autenticar) transmissor e receptores, configurando endereços multicast e outros parâmetros necessários.

Por fim, assume-se que o tempo de transmissão é limitado por um *deadline*, que é configurado pelo usuário como um requisito da aplicação (uma transmissão que excede o *deadline* deve ser abortada). Além disso, a transmissão deve ser executada no menor tempo possível, de acordo com as condições da rede: largura de banda disponível, latência e taxas de erros. Quando a rede for compartilhada, o protocolo deve empregar controle de congestionamento amigável ao TCP, ou alternativamente ser configurado com uma taxa de envio bastante conservadora.

Apesar de o serviço de transmissão a ser prestado à aplicação acima estar relativamente bem definido, acredita-se que represente uma parcela significativa e importante de aplicações, não apenas em Computação em Grade. A próxima seção discute o estado-da-arte em

multicast confiável, e como o mesmo se situa em relação aos requisitos e premissas desse serviço.

3. Protocolos Multicast Confiável

A literatura é rica em exemplos de protocolos multicast confiável, fruto de pesquisa precursora na segunda metade dos anos 90. O amadurecimento dessa pesquisa levou à criação do Grupo de Trabalho (GT) em Multicast Confiável no âmbito da IETF [IETF Working Group on Reliable Multicast Transport,]. "Famílias" de protocolos multicast foram identificadas em [B. Whetten et al., 2001], e a partir daí os trabalhos do GT foram guiados por "blocos de construção" (*building blocks*) e "instanciações de protocolos" (*protocol instantiations*). Os blocos são componentes modulares de granulosidade grossa que são comuns a múltiplos protocolos, enquanto instanciações são especificações que definem a "cola" lógica entre blocos e a funcionalidade adicional necessária para criar um protocolo a partir de um ou mais blocos de construção. Uma instância não define exatamente a funcionalidade presente em um protocolo, nem fixa como ela é implementada, servindo apenas de guia geral de como uma implementação deve funcionar e que tipo/formato de pacotes deve usar. Na prática, as instanciações são guiadas por implementações em andamento. A seguir, são apresentadas famílias, instanciações e implementações de protocolos multicast confiável.

3.1. Famílias de Protocolos

[B. Whetten et al., 2001] identifica famílias de protocolos, de acordo com o tipo de mecanismo de confiabilidade empregado, conforme resumido a seguir. A família **Tree-based ACK (TrACK)** corresponde a protocolos que empregam ACKs e organizam receptores de acordo com uma árvore lógica, para processamento (agregação) de *feedback* e retransmissões localizadas. **Asynchronous Layered Coding (ALC)** são os protocolos que empregam mecanismos baseados em Forward Error Correction (FEC), sem *feedback* dos receptores ou da rede para o remetente, e usam divisão em camadas e mecanismos orientados a receptor para controle de congestionamento multi-taxa. Protocolos que aproveitam software presente em roteadores mais recentes para restringir NACKs e retransmissões são categorizados na família **Router Assist**. Protocolos da família **NACK only** tentam reduzir a quantidade de pacotes de *feedback* através de mecanismos de recuperação de erro baseados em NACKs, e supressão de duplicatas através de temporizadores. A lista é complementada pela família **protocolos baseados no transmissor**: neste caso, o controle da transmissão é realizado pelo remetente dos dados, que conhece a identidade dos receptores e mantém estado sobre os mesmos, tipicamente em uma ou mais janelas de transmissão. Os mecanismos podem ser adaptações daqueles usados pelo TCP.

3.2. Instâncias de Protocolos

Trabalhos sobre as famílias de protocolos progrediu em direção à definição de instâncias e implementações. A instanciação **TRAM** - *Tree-based Reliable Multicast* foi impulsionada pela Sun Microsystems. Em 2002, os trabalhos com TrAM foram interrompidos, e a implementação equivalente (JRMS - *The Java Reliable Multicast Service,* [P. Rosenzweig et al., 1998]) descontinuada, embora isso não tenha sido anunciado nem tenha ficado claro quão funcional ou robusta é a implementação de JRMS disponível em [Sun Microsystems, 2005].

A instanciação do **ALC** [M. Luby and et all, 2002a] define um protocolo "massivamente escalável" para distribuição de dados. O ALC combina os blocos de construção LCT - *Layered Coding Transport* ([M. Luby and et all, 2002c]) e FEC

([M. Luby and et all, 2002b]) para oferecer distribuição assíncrona confiável de dados, com controle de congestionamento multi-taxa, para um número ilimitado de receptores: não há *feedback* para o transmissor. Cada receptor pode iniciar a recepção de um objeto **assincronamente**. A taxa de cada receptor na sessão é ajustada de acordo com sua própria capacidade, até a taxa máxima sendo usada pelo remetente, assinando um conjunto de grupos IP multicast.

A instanciação do **NORM** [Adamson et al., 2004] visa oferecer transporte confiável fim-a-fim de grandes quantidades de dados de um remetente para vários receptores. A instanciação do NORM combina os blocos de construção NACK [Adamson et al., 2004] e FEC para controle de erro.

O assincronismo e a ausência de *feedback* da instância ALC tornam a mesma, em princípio, inadequada para o cenário descrito na seção anterior, onde a massa de dados deve ser transmitida em alta velocidade, de forma síncrona, a um grupo de tamanho restrito e cuja composição é bem conhecida. Desta forma, foram adotadas para análise, no restante deste artigo, as instâncias NORM e TRAM, embora a implementação dessa última tenha sido descontinuada pela Sun. Em contrapartida, foram incluídos no estudo protocolos para a família "baseado no transmissor", para a qual não existe uma instância no WG RMT, mas sim implementações.

3.3. Implementações de Protocolos

Cada uma das instâncias acima possui uma ou mais implementações correspondentes, em variado grau de aderência às especificações. Foi realizada uma busca exaustiva de implementações de protocolos de multicast confiável, tanto no meio acadêmico como na indústria. Em alguns casos, o projeto não existia mais, ou os *links* web estavam quebrados. Como resultado desta consulta, selecionou-se um conjunto de implementações de protocolos a ser investigado mais a fundo, conforme listado na Tabela 1.

Tabela 1: Lista de protocolos investigados

Sigla	Nome	Instituição
LGMP	Local Group Multicast Protocol	Univ. of Karlsrhue
MDP	Multicast Dissemination Protocol	Naval Research Lab
NORM/NRL	Nack-Oriented Reliable Multicast	Naval Research Lab
NORM/INRIA	Nack-Oriented Reliable Multicast	INRIA
JRMS	Java Reliable Multicast Service	Sun Microsystems
TCP-XM	TCP eXtended for Multicast	University of Cambridge
DF	Digital Fountain Multicast	Digital Fountain, Inc

Ambos **LGMP** [M. Hofmann, 2004, M. Hofmann, 1996] e **JRMS** [Sun Microsystems, 2005, P. Rosenzweig et al., 1998] são protocolos hierárquicos e assim pertencem à família TrACK, porém apenas o segundo adere à instância TRAM. O LGMP foi um dos primeiros protocolos hierárquicos; JRMS é um conjunto de bibliotecas e serviços em Java para construção de aplicações explorando multicast.

MDP/NRL [J. P. Macker, 1999] e **NORM/NRL** [B. Adamson et al., 2004, NRL,] são desenvolvidos pelo mesmo grupo (NRL) e compartilham a mesma base de código (o NORM é derivado do MDP). Ambos usam um mecanismo seletivo de NACK para obter confiabilidade, com potencial auxílio de FEC. O MDP/NRL possui um mecanismo de controle de congestionamento (na sua terminologia, "controle de fluxo"), mas o mesmo não é amigável ao TCP. Já o mecanismo de controle de congestionamento do NORM/NRL

seleciona dinamicamente o pior receptor, denominado "CLR", e faz com que o mesmo envie conformações de recebimento ao transmissor. O *feedback* do receptor mais lento é usado para alimentar um algoritmo similar ao do TCP. O **NORM/INRIA** [V. Roca, 2005], outra implementação da instância NORM, faz parte da biblioteca MCL - *Multicast Library* e é menos madura e estável que o NORM/NRL.

A implementação **DF** [Byers et al., 1998], que não segue nenhuma família específica, está baseada em blocos de construção FEC. A mesma é fechada e proprietária, tendo sido cedida pela Digital Fountain especificamente para esta avaliação.

O **TCP-XM** [K. Jeacle and J. Crowcroft, 2003], da família "baseado no transmissor", é um protocolo que estende (uma implementação leve em nível de usuário) o Transport Control Protocol de forma a oferecer transmissão confiável similarmente ao TCP. O transmissor estabelece conexões com cada um dos receptores, para troca de informações de controle, porém envia pacotes usando IP multicast (sempre que possível). A característica que distingue o TCP/XM é exatamente essa: a possibilidade de mesclar em uma mesma sessão receptores unicast e multicast.

Das implementações acima, a única que não pôde ser aproveitada na avaliação foi o LGMP, que apresentou erros de compilação e execução e não é mais mantido pelos autores. Em compensação, foi avaliado também o protocolo TCP para transmissão multiponto confiável; neste caso, o transmissor estabelece múltiplas conexões paralelas (independentes), uma por receptor. O protocolo, que é da família "baseado em transmissor", é doravante denominado **MultiTCP**. Foi avaliado também o uso de IP multicast "puro", com o objetivo de assegurar existência de largura de banda suficiente para acomodar o tráfego multicast fim-a-fim. Neste caso, não há garantias de confiabilidade, sendo portanto o mesmo denotado como **BestEffort**.

Na avaliação a seguir, foi empregada a versão mais recente disponível para cada protocolo. Em alguns casos, as implementações incorporaram melhorias recentes em função de colaboração com os autores dos protocolos, que tiveram acesso a resultados parciais.

4. Cenários de Avaliação

Para avaliar experimentalmente as implementações, foram definidos cenários, conforme descrito a seguir. Em linha com os requisitos colocados na Seção 2, um arquivo deve ser transmitido confiavelmente por um protocolo para um conjunto de máquinas remotas, cuja identidade é conhecida. O conteúdo do arquivo é verificado em cada receptor através de um *checksum* md5, de forma a garantir sua integridade, embora essa conferência não seja incluída na contabilização do tempo de transferência.

Cada experimento foi repetido um número estatisticamente significativo de vezes: pelo menos 20 para cada um dos pontos de uma curva (mais vezes quando necessário). Barras de erro são incluídas nos gráficos de forma a indicar a variação nos resultados.

4.1. Métricas

Três métricas foram empregadas na avaliação, como segue. A primeira, aqui considerada a mais importante, é denominada ***goodput***, ou ***G,*** e computada dividindo-se o tamanho do arquivo a ser transmitido pelo tempo necessário para transmiti-lo confiavelmente. O tempo de transmissão é dado pelo momento entre o início da transmissão dos dados e o momento em que o último receptor termina, assinalando o recebimento completo do arquivo (a verificação da integridade do arquivo é feita em seguida).

A segunda métrica é denominada **sobrecarga de rede**, ou ***N,*** e reflete a quantidade extra de largura de banda utilizada pelo protocolo. Este valor é calculado subtraindo-se

da quantidade total de bytes enviada ou recebida pelo transmissor menos o tamanho do arquivo de dados. No caso de uma transferência unicast, transmissões redundantes contarão múltiplas vezes.

Robustez, ou ***S***, é computada de acordo com o resultado da transferência. O resultado, sucesso ou falha, deriva de dois fatores: primeiro, se o arquivo é entregue de forma integral em todas as máquinas destinatárias, assumindo que elas continuam funcionando, assim como o caminho de rede até as mesmas, durante a sessão; segundo, se a transferência é completada respeitando o *deadline* proposto (vide Seção 2). O valor de *S* é então computado como a percentagem de experimentos bem sucedidos. Note-se, finalmente, que apenas experimentos bem sucedidos são utilizados no cálculo de *G*.

4.2. Ambientes de Rede Empregados

Conforme argumentado na Introdução, o principal diferencial desse trabalho é oferecer resultados obtidos através de uma avaliação experimental, com implementações atuais de multicast confiável e aplicações com troca intensiva de dados, em ambientes de produção. Para execução dos experimentos, foram utilizados dois ambientes de rede diferentes: WAN e LAN, conforme explicado a seguir.

No ambiente **WAN** (*Wide-Area Network*) empregou-se um conjunto geograficamente disperso de 11 PCs com GNU/Linux interligadas à rede acadêmica de alta velocidade britânica, a SuperJanet [UKERNA, 2005], nos seguintes sites: BT Research, University of Manchester, University of Cardiff, University College London, University of Cambridge, Imperial College, Daresbury Labs, University of Southampton, e University of Newcastle upon Tyne. A largura de banda efetiva fim-a-fim foi previamente medida com UDP e foi de aproximadamente 75Mbps. O *round-trip time* (RTT) médio, em todos os casos, foi inferior a 15ms (e 16 *hops*). A taxa média de perda de pacotes verificada, com um fluxo multicast de 70Mbps do remetente aos receptores, foi inferior a 5% para o receptor de menor capacidade (para a maior parte dos receptores, a taxa média de perdas foi inferior a 1%).

Constatou-se uma grande heterogeneidade no conjunto de máquinas quanto ao poder de processamento, variando entre 2×4194 e 592bogomips. Como as implementações avaliadas correspondem, à exceção do MultiTCP, a protocolos de taxa-única, a ordem das máquinas no grupo afeta diretamente os resultados de desempenho. A ordem empregada neste trabalho é da melhor máquina para a pior, de forma a salientar o potencial de desempenho.

O ambiente de **LAN** (*Local-Area Network*) conta também com 11 máquinas, em uma rede FastEthernet. As máquinas são estações de trabalho convencionais Intel com sistema operacional GNU/Linux e kernel 2.6.x, com capacidade variando entre 2×4194 e 2400bogomips (usado apenas como indicativo geral de capacidade de processamento). A ordem das máquinas foi escolhida de acordo com a capacidade de processamento (decrescente).

Os ambientes utilizados são **redes de produção**, em contraste a *testbeds*. Experimentos com taxas fixas e altas poderiam prejudicar usuários da rede e das máquinas onde os receptores executam. Assim, para viabilizar este trabalho, experimentos foram restritos a horários fora de pico. Por um lado, como há menos fontes de interferência (na rede e nas estações-fim), aumenta-se a confiança nos resultados e torna-se mais fácil avaliar os resultados. Por outro, os valores encontrados oferecem um limite superior em termos de goodput.

4.3. Parâmetros de Entrada

Além da capacidade da rede e suas condições atuais, o desempenho de um protocolo dependerá dos valores atribuídos aos parâmetros de entrada presentes na implementação. Este é

o caso, particularmente, de protocolos de taxa fixa. Previamente, foi avaliado o impacto dos principais parâmetros de entrada de cada protocolo. Em todas as execuções, foram usados *buffers* de 16MB no remetente e no receptor. Um arquivo de 64MB é enviado a grupos que variam entre 1 e 10 receptores, com um *deadline* configurado em 512 segundos (imposição de um *goodput* mínimo de 1Mbps).

Apenas parte das implementações avaliadas possuem mecanismo de controle de congestionamento; adicionalmente, apenas parte permite a configuração de uma taxa de envio (máxima). Dois cenários foram avaliados, tanto em WAN como em LAN: um "cenário de taxa fixa", e um "cenário adaptativo". A diferença entre ambos é o uso, pelo protocolo, de um mecanismo de controle de congestionamento.

Os protocolos avaliados no cenário adaptativo foram o MDP/NRL, NORM/NRL, JRMS, TCP-XM e MultiTCP, enquanto os protocolos avaliados no cenário de taxa fixa foram MDP/NRL, NORM/NRL, NORM/INRIA, além de BestEffort. As seguintes taxas foram empregadas: 70Mbps para BestEffort, 60Mbps para MDP/NRL, NORM/NRL e DF, e 10Mbps para NORM/MCL (este último trabalha com "perfis", sendo o mais alto deles o "LAN", correspondente a 10Mbps). O protocolo JRMS possui controle de congestionamento, no entanto aceita como entrada uma taxa máxima de envio; neste caso, a taxa usada foi de 30Mbps.

Para os protocolos MDP/NRL e NORM/NRL, foram usados blocos de 127 pacotes com 10 de paridade. No caso de NORM/INRIA e do DF, foi selecionado 25% de paridade. Por fim, no caso do JRMS, foi empregada uma janela de ACKs de 32 pacotes.

Porque o espaço de parâmetros é vasto e inter-relacionado (por exemplo, taxa de envio e quantidade de redundância pró-ativa em protocolos com FEC), e a melhor combinação de parâmetros é algo que varia com as condições da rede e as características do grupo de receptores, não é possível **garantir** que a melhor combinação de parâmetros tenha sido encontrada em cada caso. Entretanto, os autores das implementações foram consultados, e foi realizada uma extensa avaliação de valores de entrada.

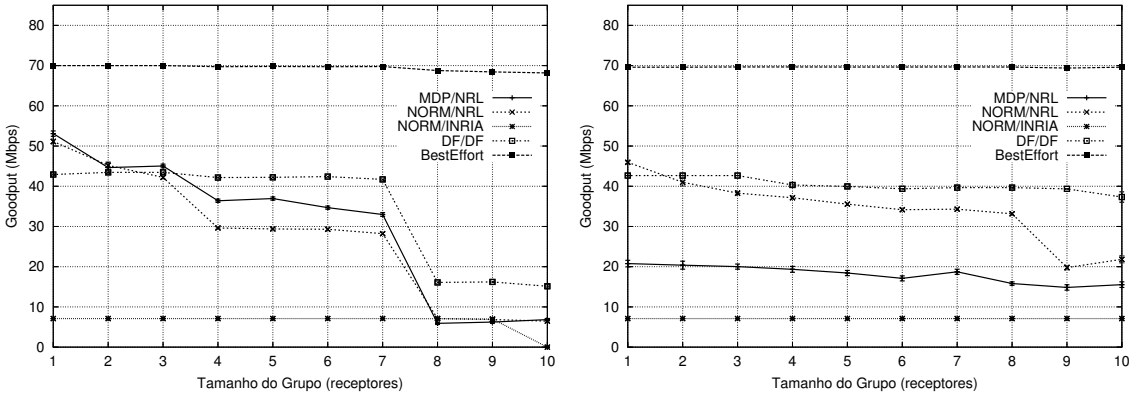
5. Resultados e Análise

Esta seção apresenta os resultados obtidos com os experimentos e faz uma análise dos mesmos. Os gráficos nem sempre apresentam todos os protocolos, pois ocorreram problemas específicos de implementação em determinados casos, a saber: o JRMS apresentou robustez praticamente nula no ambiente de WAN, com alto de falhas mesmo na LAN. O TCP-XM, por sua vez, não pôde ser executado na WAN em duas das máquinas, devido a questões de configuração local das mesmas, portanto limitando o tamanho do grupo a 8 receptores.

Em todos os gráficos mostrados a seguir, o eixo *x* representa o tamanho do grupo, enquanto o eixo *y* a métrica em questão (que pode ser *G* ou *N*). As medidas de *G* estão expressas em Mbps, as de *N* em MB (Megabytes), enquanto *S* é um percentual. GS é usado para denotar o tamanho de um grupo.

5.1. Cenários de Taxa Fixa

As Figuras 1(a) e (b) apresentam os gráficos de ***goodput*** para o cenário de taxa fixa em WAN e LAN, para os protocolos MDP/NRL, NORM/NRL, NORM/INRIA, DF/DF (ou simplesmente, DF) e BestEffort. O primeiro aspecto a se notar é a curva BestEffort, no topo das Figuras 1(a) e (b), que demonstra que a rede é capaz de entregar aproximadamente 70Mbps de IP multicast para todos os tamanhos de grupo (na WAN, 68,16Mbps para $GS = 10$).



(a) WAN (b) LAN

Figura 1: *Goodput* (*G*) para diferentes tamanhos de grupos em ambientes WAN e LAN com transmissão de taxa fixa.

No outro extremo da escala, o protocolo NORM/INRIA obtém valores de *G* que oscilam entre 7,11 e 6,99Mbps para $1 \leq GS \leq 9$ na WAN, e 7,11 e 7,10Mbps na LAN. Este desempenho, com baixa variabilidade, é explicado pelos seguintes fatos: (a) NORM/INRIA transmite com taxa fixa especificada, que é suficientemente baixa (10Mbps) para ser atingida/mantida; (b) da taxa de envio (10Mbps) é subtraída a sobrecarga advinda do uso de 25% de paridade. O mecanismo de recuperação é pobre e não lida adequadamente com perdas, levando à falhas. Por essa razão, com $GS = 10$, nenhuma das execuções completou na WAN e portanto $G = 0$. Taxas superiores foram tentadas, alterando-se a biblioteca MCL, mas experimentos mostraram que a implementação se tornou instável.

O melhor desempenho global, tanto na WAN como na LAN, foi obtido pelo DF, com G em torno de 40Mbps para todos os casos na LAN e $GS \leq 7$ na WAN. Neste ambiente, em $GS = 8$ o *goodput* decresce abruptamente para 16,08Mbps porque a oitava máquina (idem nona e décima), de menor capacidade, está sujeita a uma parcela (pequena) de perdas, é lenta no recebimento dos pacotes (mais tempo transcorre até que o receptor DF tenha recebido o número necessário de pacotes) e na decodificação dos "símbolos" FEC. Na LAN, não há perdas, e o receptor mais lento possui capacidade de processamento suficiente.

Similar fenômeno de queda de G na WAN para $GS \geq 8$ aparece tanto no MDP/NRL como no NORM/NRL (a principal diferença entre estes protocolos é no controle de congestionamento, desativado nesse caso). Os três receptores mais lentos não conseguem manter a taxa de 60Mbps, devido à sobrecarga imposta pelos protocolos no tratamento de pacotes (incluindo decodificação de FEC, gerência de *buffers* e temporizadores), levando à perdas de pacotes e ineficiências no funcionamento do mecanismo de NACK. Perdas precisam ser recuperadas, atrasam o "avanço normal" e dificultam a gerência de *buffers* no transmissor, que precisa manter os pacotes ou executar operações de *seek* em disco. Por outro lado, para $GS \leq 7$, ambos protocolos apresentam bom desempenho na WAN, na casa dos 30Mbps para $4 \leq GS \leq 7$ e dos 40Mbps para $GS \leq 3$. Na LAN, MDP/NRL atingiu um G bastante superior ao NORM/NRL, para todos os GS; esta diferença variou entre 25 ($GS = 1$) e 6Mbps ($GS = 10$).

Os fenômenos nos gráficos de desempenho de WAN com receptores lentos são percebidos também na **sobrecarga de rede**, conforme visto na Figura 2. Existe uma correlação entre os valores de G e N para MDP/NRL e NORM/NRL, o que leva a crer que retransmissões entram na fila em grande número e "roubam" *slots* dos pacotes de dados originais.

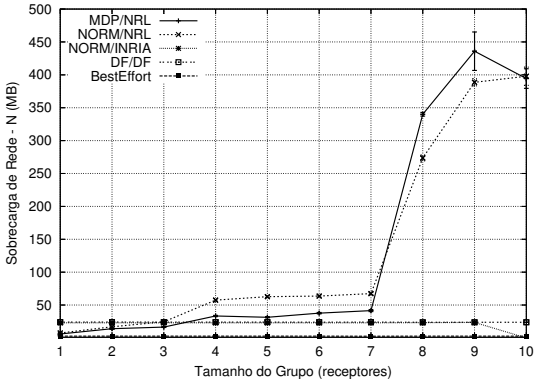


Figura 2: Sobrecarga de rede (*N*) para diferentes tamanhos de grupos em ambientes WAN com transmissão de taxa fixa.

Ilustrando, quando $GS \geq 8$, o valor de N sobe abruptamente e atinge mais de 400MB de sobrecarga; no caso do MDP/NRL, quando $4 \leq GS \leq 7$, os valores médios de N oscilam entre 31.43 e 41.55MB (uma sobrecarga aproximada de 50% ou mais do tamanho do arquivo), e menor do que isso para $GS \leq 3$. Valores levemente mais altos são observados com o NORM/NRL. Esta sobrecarga deriva da quantidade de perdas geradas (pacotes de NACK) e das retransmissões subseqüentes; outra causa para N tão alto é que cada NACK é enviado ao transmissor e potencialmente re-enviado ao grupo, pois MDP/NRL e NORM/NRL tiveram que operar em "modo nack-unicast" devido à falta de conectividade multicast simétrica completa.

As sobrecargas do NORM/INRIA e do DF são bastante constantes, 22,99$\sim$23,56MB e 23,79$\sim$24,13Mbps respectivamente, para todos os GS, e inferiores ao MDP/NRL e NORM/NRL para $GS \geq 4$. Isto se deve ao fato que esses protocolos empregam FEC como mecanismo único e usam a mesma taxa de redundância, 25%. Interessante notar que N deve ser neste caso pelo menos 16MB $(0,25 \times 64)$; devido aos cabeçalhos dos protocolos e das camadas inferiores, a sobrecarga é de aproximadamente 23MB.

BestEffort, por sua vez, não apresenta sobrecarga porque não possui *feedback* ou mecanismo de controle de erro.

A análise de **robustez** das implementações consiste em avaliar a proporção de experimentos falhos dentre o número total de experimentos. Assim, foi necessário executar uma quantidade significativa de experimentos para que fossem obtidos índices de robustez confiáveis. Ressalva-se, no entanto, que certas falhas poderão ocorrer devido ao mal funcionamento do IP multicast ou devido a uma limitação das estações-fim. Os resultados de robustez são apresentados na Tabela 2, considerando pelo menos 200 experimentos por protocolo na WAN e 100 na LAN, uniformemente distribuídos através dos diferentes tamanhos de grupo.

Tabela 2: Índices globais de robustez *S* para cenários de taxa fixa.

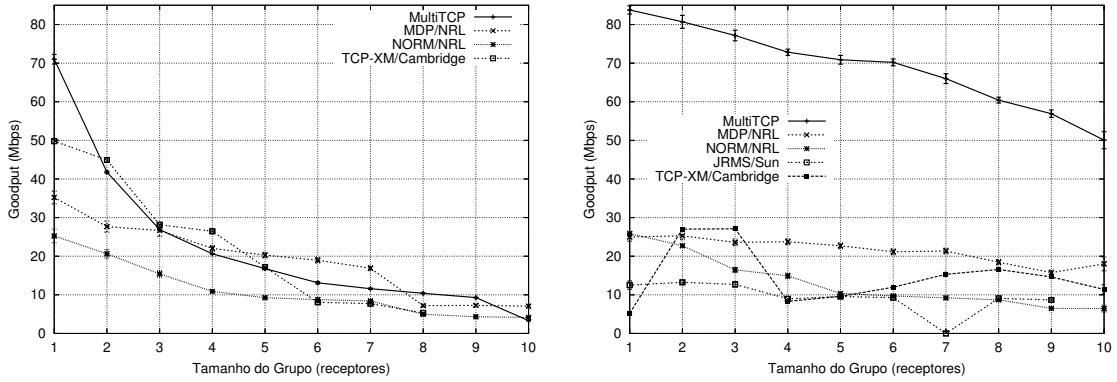
Protocolo	S em WAN	S em LAN
MDP/NRL	97,8	100
NORM/NRL	92,7	99,3
NORM/INRIA	67,1	100
DF/DF	93,7	100

Conforme esperado, WAN e LAN se mostram dois ambientes bastante distintos em

termos de robustez dos protocolos: na LAN, três dos quatro protocolos atingiram robustez perfeita, ou $S = 100$, enquanto este não foi o caso na WAN para nenhum protocolo. Na LAN, registrou-se uma única falha, que ocorreu em um dos 139 experimentos com o NORM/NRL. Na WAN, destaca-se negativamente a falta de robustez do NORM/INRIA, com $S = 67,1$, muitos deles com colapso do programa transmissor ou de um ou mais receptores.

5.2. Cenários Adaptativos

As Figuras 3(a) e (b) apresentam os gráficos de *goodput* para o cenário de taxa adaptativa, em WAN e LAN. São apresentadas curvas para os protocolos MultiTCP, MDP/NRL, NORM/NRL, TCP/XM e JRMS. O primeiro aspecto a se notar é que os gráficos são bem distintos para os ambientes WAN e LAN; contrário ao caso de taxa fixa, neste ambiente os protocolos fazem um esforço para "medir" a rede e determinar a melhor taxa de envio.



(a) WAN (b) LAN

Figura 3: *Goodput* (*G*) para diferentes tamanhos de grupos em ambientes WAN e LAN com transmissão de taxa adaptativa.

No caso da WAN, o G do MultiTCP decai de forma rápida mas previsível, devido às múltiplas transmissões redundantes (conforme demonstrado no gráfico de sobrecarga, na Figura 4): $G = 71$ para $GS = 1$ (unicast), e 3,3Mbps, para $GS = 10$. Em contraste, o fraco desempenho das implementações de multicast confiável não é nada previsível, e aponta para ineficiências no mecanismo de controle de erro e de congestionamento das mesmas. Na WAN, em termos gerais, não há um claro vencedor com melhor desempenho, alternando-se o melhor G entre os protocolos de acordo com GS. Destaca-se apenas a diferença entre MDP/NRL e NORM/NRL, consistente em favor do primeiro para todos os valores de GS, o que está de acordo com o fato que o algoritmo de controle de congestionamento do MDP/NRL é mais agressivo do que o do NORM/NRL.

Já no caso da LAN, o G do MultiTCP é muito superior a todos as demais implementações, variando entre 83,7 e 50Mbps; o segundo melhor resultado, pertencente ao MDP/NRL, vai de 25,2 a 15,7Mbps com aumento de GS. Assim como na WAN, MDP/NRL apresentou desempenho superior ao NORM/NRL. Bastante surpreendente é o comportamento exibido pelo TCP/XM, cujo desempenho varia sem uma lógica aparente. Foram executados pelo menos 22 experimentos com TCP/XM para cada ponto apresentado, e conforme demonstra a barra de erro, a variância foi bastante baixa. Ilustrando, o **maior** G obtido nas 22 execuções de TCP/XM para $GS = 1$ foi 6,65Mbps, enquanto o **menor** G obtido com as 22 execuções para $GS = 2$ foi 21,3Mbps. Ou seja, apesar de

aparentemente patológico, este comportamento foi exibido pelo protocolo de forma consistente através dos experimentos. Por fim, note-se o desempenho do JRMS, relativamente estável em torno de 10Mbps, exceto para $GS = 7$, quando nenhum experimento completou com sucesso.

A Figura 4 demonstra a sobrecarga em WAN. Conforme comentado anteriormente, o N do MultiTCP cresce rapidamente, de forma linear com GS, devido às múltiplas *streams* redundantes. Comparativamente, a sobrecarga dos demais protocolos é pequena: $4,16 \sim 46,74$ para MDP/NRL, $4,16 \sim 34,86$ para NORM/NRL, e $5,14 \sim 96,94$ para TCP/XM (note que o protocolo possui apenas 8 pontos devido a restrições em duas máquinas do conjunto). Relembrando, TCP/XM é capaz de usar multicast e unicast ao mesmo; uma troca no transmissor, de multicast para unicast, quando do oitavo receptor, explicaria o custo um pouco mais elevado para $GS = 8$.

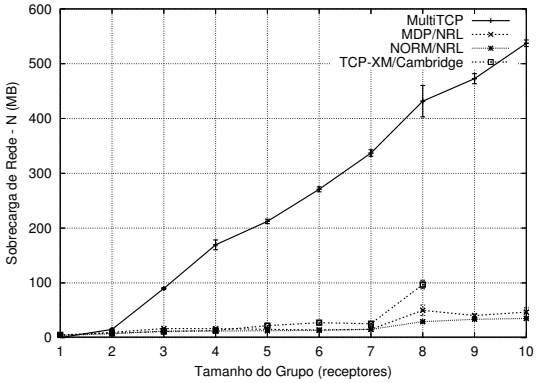


Figura 4: Sobrecarga de rede (*N*) para diferentes tamanhos de grupos em ambientes WAN com transmissão de taxa adaptativa.

Os resultados de robustez são apresentados na Tabela 3, considerando no mínimo 400 experimentos por protocolo na WAN e 200 na LAN (o dobro de experimentos que no caso fixo porque os resultados apresentaram maior variação). Tal como no caso do cenário de taxa fixa, existe uma grande diferença entre WAN e LAN. Conforme era de se esperar, o protocolo mais robusto foi o MultiTCP, com $S = 100$ em ambos os casos, uma vez que o mesmo usufrui da robustez do TCP. O segundo protocolo mais robusto foi o MDP/NRL, que não coincidentemente é o segundo protocolo com mais tempo de desenvolvimento. O destaque negativo é o JRMS, que apresentou robustez bastante baixa, no caso da WAN chegando a inviabilizar os experimentos com o mesmo.

Tabela 3: Índices globais de robustez *S* para cenários de taxa adaptativa.

Protocolo	S em WAN	S em LAN
MultiTCP	100	100
MDP/NRL	99,3	100
NORM/NRL	94,1	100
JRMS	4,0	26,0
TCP/XM	96,1	85,8

6. Considerações Finais

Este trabalho apresentou um estudo sobre desempenho, custo e robustez de protocolos multicast confiável, alicerçado em uma extensa avaliação experimental de implementações que representam o estado da arte na área. Analisando-se os resultados de forma global,

nos ambientes LAN e WAN, e cenários com taxa fixa e adaptativa, conclui-se que: (i) as implementações de multicast confiável tem o potencial de oferecer os ganhos previstos em termos de economia de recursos de rede; (ii) o custo e desempenho do protocolo dependem, fortemente, da "correta" configuração dos parâmetros do protocolo em relação às condições da rede, algo não trivial; (c) as implementações não atingem os níveis de desempenho esperados, considerando os recursos de rede disponíveis.

Os resultados servem como um "choque de realidade", pois esperava-se que, pelo menos em grupos pequenos, protocolos multicast atingiriam desempenho similar a uma única *stream* TCP direcionada o receptor mais lento do grupo. Caso assim fosse, o gráfico de desempenho dos protocolos multicast iniciaria em torno de 70Mbps, para $GS = 1$, e não cairia mais do que 10 ou 15% até $GS = 10$.

Muitas são as causas potenciais para o desempenho pobre, incluindo problemas de implementação nos protocolos ou de configuração em máquinas fim e/ou na rede. Com base nos resultados obtidos, acredita-se que os fatores limitantes sejam, para TCP/XM e JRMS, oriúndos de ineficiências na implementação. Já para DF, MDP/NRL, NORM/NRL e NORM/INRIA, os efeitos são (i) a sobrecarga de processamento nas estações fim, particularmente nos protocolos de FEC devido às tarefas de (de)codificação, e (ii) o *design* baseado em receptor, obtendo escalabilidade em detrimento de desempenho. Trabalhos futuros irão tratar essas questões mais a fundo.

Além desta, existem outras perspectivas para continuidade deste trabalho, incluindo: (i) a extensão do conjunto de protocolos, acrescentando-se p.ex., FLUTE [T. Paila et al., 2004]; (ii) a investigação aprofundada do NORM/NRL (em conjunto com os autores); e (iii) modificação do ambiente de rede, seja via expansão do conjunto de máquinas em uma configuração global (incluindo *sites* no Brasil, na Irlanda e nos Estados Unidos), seja via execução de experimentos em uma rede menor porém com características bem distintas. Em relação a esta última possibilidade, está em andamento uma avaliação similar na RNP, via um de seus Grupos de Trabalho [V. Roesler et al., 2005].

Referências

Adamson, B., Bormann, C., Handley, M., and Macker, J. (2004). NACK-Oriented Reliable Multicast (NORM) Building Blocks. RFC 3941. http://www.ietf.org/rfc/rfc3941.txt.

B. Adamson, C. Bormann, M. Handley, and J. Macker (2004). Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol. RFC 3940. http://www.ietf.org/rfc/rfc3940.txt.

B. N. Levine and J.J. Garcia-Luna-Aceves (1998). A Comparison of Reliable Multicast Protocols. *ACM Multimedia Systems Journal*, pages 334–348.

B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd, and M. Luby (2001). Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer. RFC 3048. http://www.ietf.org/rfc/rfc3048.txt.

Byers, J. W., Luby, M., Mitzenmacher, M., and Rege, A. (1998). A digital fountain approach to reliable distribution of bulk data. In *SIGCOMM*, pages 56–67.

C .Diot, B. N. Levine, and B. Liles (2000). Deployment issues for the IP multicast service and architecture. *IEEE Network*, pages 78–88.

C. Hänle and M. Hofmann (1998). A Comparison of Reliable Multicast Protocols using the Network Simulator ns-2. In IEEE, editor, *IEEE Conference on Local Computer Networks (LCN)*, pages 222 – 237, Boston, MA.

Chalmers, R. and Almeroth, K. (2001). Modeling the Branching Characteristics and Efficiency Gains of Global Multicast Trees. In *IEEE Infocom*, pages 77–86.

IETF Working Group on Reliable Multicast Transport. RMT Charter website. http://www.ietf.org/html.charters/rmt-charter.html.

J. P. Macker (1999). The Multicast Dissemination Protocol (MDP) Toolkit. In *IEEE MILCOM*, volume 1, pages 626–630.

K. Jeacle and J. Crowcroft (2003). Reliable high-speed Grid data delivery using IP multicast. In *All Hands Meeting*. http://www.allhands.org.uk/2004/.

M. Hofmann (1996). A Generic Concept for Large-Scale Multicast. In *IZS'96 - Intl. Zurich Seminar on Digital Communications*, pages 95–106.

M. Hofmann (2004). Local Group Concept website. http://lgmp.mhof.com/.

M. Luby and et all (2002a). Asynchronous Layered Coding (ALC) Protocol Instantiation. RFC 3450. http://www.ietf.org/rfc/rfc3450.txt.

M. Luby and et all (2002b). Forward Error Correction Building Block. RFC 3452. http://www.ietf.org/rfc/rfc3452.txt.

M. Luby and et all (2002c). Layered Coding Transport (LCT) Building Block. RFC 3451. http://www.ietf.org/rfc/rfc3451.txt.

M.W. Koyabe and G. Fairhurst (2001). Reliable Multicast via Satellite: a comparison survey and taxonomy. *International Journal of Satellite Communications (IJSC)*, 24(1):21–26.

NRL. Naval Research Laboratory (NRL) PROTocol Engineering Advanced Networking (PROTEAN) Research Group - NACK-Oriented Reliable Multicast (NORM) website. http://norm.pf.itd.nrl.navy.mil/.

P. Rosenzweig, M. Kadansky, and S. Hanna (1998). The Java Reliable Multicast Service: A Reliable Multicast Library. Technical report, Sun Microsystems, Inc.

S. Pingali, D. Towsley, and J. Kurose (1994). A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols. *ACM SIGMETRICS Conf. on Measurement and Modelling of Computer Systems*.

Sun Microsystems (2005). Java Reliable Multicast Service website. http://www.experimentalstuff.com/Technologies/JRMS/.

T. Paila, M. Luby, R. Lehtonen, V. Roca, and R. Walsh (2004). FLUTE - File Delivery over Unidirectional Transport. RFC 3926. http://www.ietf.org/rfc/rfc3926.txt.

UKERNA (2005). SuperJANET website. http://www.ja.net/.

V. Roca (2005). INRIA-Rhone-Alpes MCLv3 Project website. http://www.inrialpes.fr/planete/people/roca/mcl/mcl.html.

V. Roesler, M. Barcellos, T. Farias, G. Facchini, and G. Brandt (2005). GT Multicast Confiável. http://prav.unisinos.br/gtmc/.

V. Sander (Ed.) (2004). Networking Issues of Grid Infrastructures. GGF GFD.37. http://www.ggf.org.