Distribuição de Dados Massivos via Multicast Confiável na RNP: metodologia de avaliação, ferramentas e resultados

Gustavo Bervian Brand, Giovani Facchini, Evandro Dall'Agnol, Renato Costa, Marinho Barcellos, Valter Roesler

Universidade do Vale do Rio dos Sinos (Unisinos)
Av. Unisinos, 950 – São Leopoldo – RS – Brasil

{gugabrand, facchini, evandrocd}@gmail.com, renatof@turing.unisinos.br, marinho@acm.org, roesler@unisinos.br

Resumo. *A transmissão de grandes volumes de dados de um ponto para múltiplas estações de destino através de um País de tamanho continental como o Brasil é um desafio. Neste cenário, o artigo investiga o problema de distribuir antecipadamente a grade de programação das TVs Universitárias, em arquivos ocupando dezenas de Gigabytes, via Rede Nacional de Pesquisa (RNP). Para transmití-la em tempo hábil, é necessário considerar tanto a carga imposta à rede na transmissão dos dados como o tempo necessário para completar a transferência de forma confiável. O trabalho explora a viabilidade do uso de protocolos multicast confiáveis na transmissão de conteúdo multimídia offline através da RNP. Uma ferramenta baseada na Web foi projetada, implementada e disponibilizada para uso. Uma metodologia de teste foi estabelecida e utilizada, através de uma ferramenta de avaliação, para tomar medidas de desempenho e custo. Os resultados apontam multicast confiável como uma solução adequada aos objetivos propostos.*

Abstract. *It is a challenge to transmit large data volumes from one point to multiple destinations through a country of continental dimensions like Brazil. We investigate the problem of distributing in advance, through the RNP (Rede Nacional de Pesquisa), the entire program set created for the University TVs. As these files occupy dozens of Gigabytes, the feasibility of the transmission depends on both the traffic added to the network and the time required to complete the transfer reliably. This paper exploits the feasibility of using reliable multicast to transfer offline multimedia content through the RNP. A Web-based tool was designed, implemented and made available for general use. A test methodology was established and employed, by means of an evaluation tool, to obtain performance and cost measurements. The results indicate reliable multicast as an adequate solution for the proposed objectives.*

1. Introdução

Multicast, na sua forma mais simples, pode ser definido como a transmissão de um pacote de um remetente para múltiplos destinatários, reunidos em um grupo e referenciados através do mesmo. IP multicast surgiu no início dos anos 90 e, desde então, a presença de suporte a "multicast nativo" em equipamentos de rede é cada vez mais comum. Protocolos de multicast confiável visam aumentar as garantias ao usuário, acrescentando certo nível de confiabilidade na entrega de pacotes a destinatários. Em outras palavras, perdas de pacotes na rede são recuperadas através de um mecanismo de controle de erro, que pode variar bastante entre protocolos.

Até recentemente, multicast era relativamente pouco utilizado, pelo menos em relação ao que se imaginava inicialmente. Entretanto, a situação parece estar finalmente mudando. Na Europa, grandes provedores começam a habilitar multicast em suas redes em função de solicitações de clientes. Aplicações como *Access Grid* têm funcionado como propulsores da tecnologia, demonstrando que multicast é adequado e pode ser usado com pouco esforço

adicional de gerência. As aplicações potenciais de multicast, ou seja, que empregam comunicação *um-para-vários*, são muitas. Em [Diot, 2000], são identificadas quatro classes de aplicações multicast: distribuição de áudio e vídeo em tempo real; áudio/videoconferência e aplicações interativas de grupo; aplicações *push;* e transferência de arquivos.

Este artigo relaciona-se com o último tipo de aplicação, pois o problema considerado consiste em enviar dezenas de Gigabytes de uma estação para um grupo de estações espalhadas geograficamente no Brasil. Para transmitir em tempo hábil essa quantidade de dados, é necessário levar em conta tanto a carga imposta à rede na transmissão dos arquivos, como o tempo necessário para completar cada transferência de forma confiável. Uma das contribuições do artigo é avaliar diversos protocolos de multicast segundo estes dois fatores, investigando a viabilidade do uso de multicast confiável na Rede Nacional de Pesquisa, a RNP (ou em redes com características similares). Além dos resultados encontrados, são contribuições também a metodologia de avaliação empregada e a ferramenta de avaliação desenvolvida para tal fim, e um software para transmissões multicast com suporte a agendamento via Web.

O presente artigo estende o trabalho de avaliação experimental reportado em [Barcellos, 2005], mas difere-se do mesmo no cenário de avaliação. Contudo, não será abrangido avaliação Analítica e nem Simulação. No referido trabalho, foi empregada a rede acadêmica de alta-velocidade britânica, a SuperJanet, com grande largura de banda fim-a-fim. Em contraste, o presente trabalho foi desenvolvido com base na rede acadêmica brasileira em 2004/2005, com largura de banda significativamente menor, além de atrasos e taxas de perdas maiores.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta o problema tratado, enquanto a Seção 3 descreve os protocolos multicast confiável considerados. A metodologia de avaliação é apresentada na Seção 4, juntamente com experimentos preliminares na Seção 5. Tais experimentos foram necessários para a obtenção dos resultados apresentados na Seção 6. Na Seção 7 é mostrada uma aplicação implementada e disponibilizada para uso, a qual visa permitir o uso de protocolos multicast confiável de forma mais natural. Por fim, na Seção 8 encerra o artigo com comentários finais.

2. Detalhamento do Problema

Uma das motivações para desenvolver este trabalho foi uma demanda da RITU (Rede de Interconexão entre TVs Universitárias), que tem como uma de suas metas a implantação de uma rede de troca de programação entre as diversas emissoras de TV, permitindo a criação de uma grade de programação única, ou um canal Universitário, comum a todas as Universidades e de maior qualidade quando comparado a uma emissora em particular. Dessa forma, as TVs compartilharão parte de sua programação em prol do benefício de todos.

Nesse tipo de aplicação, é premente a necessidade de distribuição massiva de dados, pois os arquivos de vídeo gerados pelas emissoras de TV possuem tamanho significativo, considerando a média dos arquivos trocados na Internet. Mais precisamente, supondo codificação de 4Mbps, uma hora de programação corresponde a 1,8Gbytes. Para 12h de programação inéditas diárias, tem-se um total de 21,6 Gbytes sendo enviados diariamente pela estação cabeça de rede a cada um dos destinatários, que atualmente passam de 32 emissoras.

Este artigo avalia a viabilidade do uso de multicast confiável para efetuar as transmissões de arquivo de forma *offline* para as diferentes emissoras. O ambiente utilizado para efetuar os experimentos foi a rede da RNP, no âmbito do Grupo de Trabalho em Multicast Confiável (GTMC). Para tanto, obteve-se, junto à administração da rede da RNP, um ambiente de trabalho em diversos PoPs no Brasil. Dentre os PoPs disponíveis encontram-se RS, PR, DF, SP, AM, GO, MG, RJ, SP, PB e ES. As velocidades de conexão nominal de cada PoP são bastante variadas, indo desde 8Mbit/s com o PoP de AM, até 622 Mbit/s, entre os PoPs do Rio de Janeiro e São Paulo.

3. Implementações de Protocolos Multicast

Diversas implementações de protocolos multicast confiável foram analisadas em função do problema proposto. Existem diferentes tipos de abordagens possíveis na implementação de cada protocolo multicast confiável. Como exemplo, pode-se fazer uso de confirmações negativas de recebimento (NACKs), ou pode-se realizar uma transmissão em camadas, ou ainda fazer uso de modelos híbridos, os quais funcionam tanto com transmissão unicast como multicast. A seguir, descreve-se os protocolos empregados; maiores informações sobre os mesmos podem ser encontradas nas respectivas referências.

TCP-XM. O protocolo TCP-XM foi desenvolvido na Universidade de Cambrige [Jeacle, 2005]. Neste protocolo, ao contrário dos demais, o transmissor guarda informações sobre cada um dos receptores (por isso, diz-se que o mesmo é orientador a transmissor). O foco do protocolo são aplicações onde o grupo de receptores é pequeno (até quarenta). Sua principal vantagem é poder mesclar multicast e unicast na mesma comunicação, este último sendo empregado quando um ou mais destinatários não possui conectividade multicast. O protocolo foi desenvolvido em nível de usuário, utilizando lwIP (*Lightweight* IP).

MDP (*Multicast Dissemination Protocol*). O MDP é um protocolo que se baseia em confirmações negativas de recebimento (NACKs). Foram desenvolvidas duas versões do MDP. O MDPv1 [Macker, 1996] emprega o mecanismo de *backoff*, com o uso de temporizadores para supressão de NACKs repetidos. Já o MDPv2 [Macker, 1999] adicionou ao seu funcionamento a utilização de FEC (*Forward Error Correction*), onde as informações redundantes de paridade são adicionadas no pacote para transmissão dos dados aos receptores [Luby, 2002c]. Estas informações redundantes são geradas por um processo de codificação e podem auxiliar os receptores a recuperar possíveis informações perdidas na transmissão, evitando-se a necessidade de utilização de *feedback*.

NORM/NRL (*Nack-Oriented Reliable Multicast Protocol / Naval Research Laboratory*). O protocolo NORM (*NACK Oriented Reliable Multicast*) é derivado do MDP e compartilha os mesmos princípios: mecanismo seletivo de NACK auxiliado pelo uso de FEC (opcional à implementação). O objetivo principal do NORM é a transferência de um grande volume de dados, mas instanciações do protocolo podem ser criadas observando necessidades específicas de cada aplicação. NORM então pode ser considerado uma classe de protocolos, visto que a base de seu funcionamento foi especificada em blocos, deixando opcional na sua implementação o uso de FEC e de *feedback*. Este protocolo possui controle de congestionamento, que se caracteriza por selecionar o pior receptor para enviar informações do estado da transmissão, solicitando ao protocolo uma melhor adaptação [Adamson, 2004a], [Adamson, 2004b].

NORM/INRIA. Esta implementação também é baseada no bloco de construção NORM, mas sua implementação é parte de uma biblioteca de transmissão multicast, MCL, em desenvolvimento no INRIA [Roca, 2005]. Não há controle de congestionamento presente na versão atual: o protocolo transfere dados a uma taxa fixa, fornecida como parâmetro de entrada pelo usuário e por esse motivo, o protocolo aparece apenas nos gráficos de resultados com taxa fixa. Essa implementação é bem mais modesta em termos de parametrização, não permitindo, por exemplo, a variação do número de pacotes por bloco FEC. É possível fornecer a taxa de transferência (através de quatro perfis de velocidade) e o percentual de redundância (paridade) enviada na transmissão.

ALC/LCT-INRIA. O protocolo ALC/LCT (Asynchronous Layered Coding / Layered Congestion Control) [Luby, 2002a][Luby, 2002b] é um protocolo multicast confiável padronizado pelo IETF que se baseia em grande parte no protocolo RLC [Vicisanno, 1998]. Dessa forma, cada grupo ou nodo receptor pode especificar a taxa mais adequada para si, seja em função de limitações de banda ou do poder de processamento. Dentre as principais características do protocolo está a ausência de *feedback* dos receptores aos transmissores, ou seja, não deve existir um canal de retorno. Para se obter confiabilidade nas transmissões, a taxa de paridade de

dados utilizada é fundamental frente ao meio. Através da manipulação destas taxas de FEC, mesmo sobre um meio parcialmente instável, é possível se conseguir transmissões confiáveis através do protocolo ALC. Esse protocolo somente permite envio de dados através de taxa fixa, portanto somente está presente no gráfico de taxa fixa.

MultiTCP. Este não é um protocolo multicast, mas seu emprego visa fornecer uma base de comparação entre os protocolos multicast e a abordagem "tradicional", de transferir os dados individualmente para cada receptor usando TCP. O transmissor estabelece uma conexão TCP individual com cada um dos receptores (protocolo multi-unicast). A ferramenta foi implementada em linguagem C, e desenvolvida dentro do Projeto MUST [Nekovee, 2005].

Nas implementações analisadas, existem protocolos que suportam taxa de transmissão *fixa,* aqueles que suportam *taxa adaptativa*, e aqueles que suportam ambas. A Tabela 1 ilustra o suporte à taxa fixa ou variável nos protocolos. Os campos que estão preenchidos com cor escura, indicam o tipo de taxa que o protocolo suporta. Quando o campo estiver com '-', indica que o protocolo não tem suporte ao tipo de taxa.

Tabela 1: Classificação dos protocolos de acordo com o modo de transmissão.

Taxa	***Protocolos***					
	ALC	NORM/INRIA	TCPXM	MDP/NRL	NORM/NRL	MultiTCP
Fixa			-			-
Adaptativa	-	-				

4. Metodologia de Avaliação

Para avaliação de viabilidade no uso de multicast confiável, foram analisadas diversas implementações destes protocolos. Para comparação, além da robustez das implementações, foram considerados o custo, em termos de ocupação na rede, e o desempenho das transmissões (em função do tempo de envio necessário).

Para avaliação de desempenho, define-se a métrica *goodput* como a vazão média de dados. O valor do *goodput* é obtido dividindo-se o tamanho do arquivo transmitido pelo tempo de transmissão. Assim, se o arquivo possuir 10Mbytes e demorar 80 segundos para ser transmitido, o *goodput* será 1 Mbps. Essa informação é útil para determinar qual protocolo envia os arquivos mais rapidamente e se o tempo necessário para envio não excede o tempo disponível (por exemplo, não é possível demorar mais do que 24 horas para se transmitir uma grade de 24 horas).

A segunda métrica é a sobrecarga na rede (*overhead*), a qual reflete o custo, ou seja, o volume extra de dados enviados através da rede para que a transmissão seja efetuada corretamente, incluindo informações de controle ou pacotes adicionais. Assim, por exemplo, se para transferir um arquivo de 10Mbytes para dois destinatários o remetente envia pela rede 20Mbytes, a sobrecarga causada pelo protocolo é de 100%. A sobrecarga na rede é afetada pelo número de destinatários e pelas condições da rede.

A terceira métrica reflete a robustez do protocolo, pois mede a taxa de falhas no mesmo, sendo definida como o número de transmissões completadas com sucesso dividido pelo número total de transmissões.

Para execução dos experimentos, foi desenvolvido um conjunto de ferramentas de avaliação, composto pelo MURMET(*Multiple Reliable Multicast Experimental Toolkit*), Mpoll e Mplot. O Mpoll foi utilizado para verificar a conectividade multicast entre pontos da rede. Ele recebe e envia pacotes no endereço multicast selecionado, e monta os grupos de recepção. De uma maneira simplista, pode-se olhar essa ferramenta como um "ping" multicast com controle de recepção. O Mplot, por sua vez, é uma ferramenta baseada no gnuplot, mas com ajuste interativo de plots via interface gráfica e com funcionalidade para geração automática de diversos

tipos de gráficos relacionados aos experimentos. A seguir, descreve-se em maior detalhe a ferramenta mais importante do conjunto, o MURMET.

O MURMET foi desenvolvido pelo grupo visando a automatização dos experimentos. As dificuldades enfrentadas incluem a heterogeneidade das implementações dos protocolos (ferramentas, linguagens e estilos), dos sistemas operacionais e do hardware. Além disso, existe a necessidade de configuração, monitoramento e o término da execução dos protocolos, incluindo a detecção e recuperação de falhas. Outra facilidade desenvolvida nesta ferramenta foi a capacidade de efetuar medição e coleta de dados sobre desempenho, de forma remota e em tempo real.

Em sua essência, a aplicação realiza diversos experimentos individuais, variando seus parâmetros. Um único experimento é concluído executando a ferramenta de transferência de arquivos específica de cada protocolo. Os requisitos mínimos de parâmetros a serem utilizados para o receptor e transmissor são IP Multicast e porta, ficando os demais parâmetros dependentes do próprio protocolo. Por exemplo, o transmissor (e receptor) podem receber parâmetros como TTL *(Time to Live)* e tamanho dos *buffers* assim como os arquivos para serem transmitidos ou a localização onde os arquivos recebidos serão salvos. Alguns protocolos recebem dois ou três parâmetros (NORM/INRIA), enquanto outros recebem um número bem maior (NORM/NRL). Cada ferramenta tem sintaxe própria e forma de recebimento de parâmetros diferenciada. O uso incorreto dos parâmetros pode ocasionar uma falha na ferramenta ou torná-la muito ineficiente.

A aplicação distribuída acomoda elegantemente essa variação no número de parâmetros, significado e estilo de sintaxe. Ela utiliza um *agente origem* na máquina origem e um *agente remoto* em cada máquina destino. Os agentes remotos são ativados e esperam pela conexão do agente origem; o agente origem é iniciado e conecta em cada agente destino, os instrui sobre a configuração do experimento, sincroniza-se com eles e então inicia a execução de um experimento. Depois da execução, o agente origem coleta informações dos agentes remotos e determina se a transferência foi bem sucedida, e os valores para as métricas monitoradas. Ele também formata e grava as informações dentro de um arquivo de log chamado “result”. Nesse meio tempo, os receptores aguardam a chegada da próxima mensagem do agente origem, a qual irá indicar se um novo experimento deve ser executado, e se esse for o caso, quais os seus parâmetros.

O processo é ilustrado na Figura 1. Um conjunto de experimentos começa com o agente origem estabelecendo uma conexão TCP com cada um dos agentes receptores. Cada rodada começará com uma mensagem do tipo INFO enviada pelo agente origem para os agentes receptores. Esta mensagem irá instruir os receptores sobre aspectos específicos daquela rodada, definindo protocolo, tamanho do arquivo, parâmetros do protocolo, tempo máximo para transmissão, etc. Cada receptor cria uma *thread*, denotado por “Thread” na figura. Essa *thread* inicia a ferramenta de transmissão de arquivo (“Recvr”) e o processo criador da thread envia para o agente origem uma mensagem de READY informando que o receptor está pronto. Quando o transmissor receber a mensagem de READY de todos os receptores, ele envia para cada receptor uma mensagem START, inicia a contagem do tempo para a transmissão e cria um processo filho que iniciará imediatamente a transmissão multicast.

O que acontece a partir desse ponto depende do protocolo multicast em questão. Pode haver sincronização entre as máquinas, antes de iniciar a transmissão efetiva; em outros, o transmissor começa a transmitir assumindo que em algum ponto os receptores passarão a receber os dados. Essa fase em que dados são enviados corresponde ao “Tempo de transferência” na Figura 1. Nesse intervalo, cada receptor envia periodicamente uma mensagem informando o progresso da transmissão. A obtenção do progresso acontece de forma diferenciada para cada protocolo (dependendo da implementação do mesmo).

No lado dos receptores, antes de executar a ferramenta, um semáforo é adquirido e então a mesma é executada por uma *thread*. A ferramenta de arquivos recebe os pacotes de dados e

grava um arquivo como resultado. Quando terminada sua execução, o semáforo é liberado e a *thread* principal verifica a integridade do arquivo recebido utilizando o algoritmo do MD5 (*Message Digest version* 5).

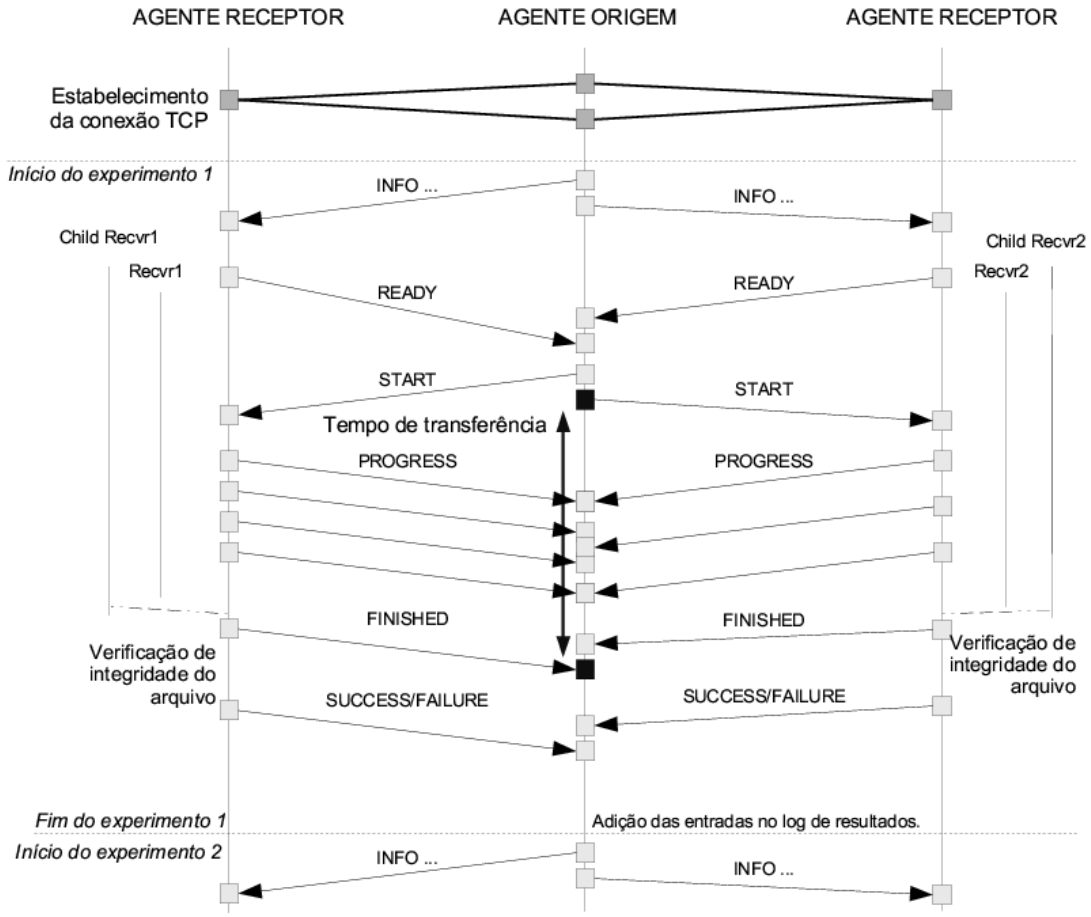

Figura 1: Visão geral de um experimento utilizando o MURMET

O agente origem fica aguardando por uma mensagem FINISH de cada receptor (utilizando uma *thread* por receptor). Quando uma *thread* recebe a mensagem FINISH ela marca o tempo de recebimento, inserindo tal informação na tabela de resultados finais. O tempo final considerado para a transmissão é o mais alto entre os receptores (pois foi quando a transmissão realmente terminou).

Se por alguma razão a transferência não for concluída, o temporizador do agente receptor irá anunciar o tempo limite da transmissão esgotado (*timeout*). Com isso a ferramenta de transmissão será finalizada e uma mensagem de FINISH seguida de uma de FAILURE será enviada ao agente origem. Existem muitas razões que podem fazer com que a transmissão não ocorra com sucesso, inclusive devido a problemas de implementação dos protocolos: *livelocks*, *deadlocks*, falha no receptor ou no transmissor (*segmentation faults*), falhas de desempenho (taxa média de transmissão fica abaixo de um limiar pré-configurado). Isso pode ocorrer devido à complexidade dos protocolos multicast confiável, pois quanto mais complexos, maior a probabilidade de falhas na implementação.

Após a execução dos experimentos, o agente origem recebe todos os *logs* dos receptores, gerando os gráficos de resultados.

5. Experimentos Preliminares

Visando obter resultados cientificamente corretos, foi necessário efetuar uma série de experimentos preliminares no ambiente alvo. A preparação do ambiente consistiu na definição das velocidades de interconexão entre os PoPs, no ordenamento das máquinas de acordo com a velocidade obtida, na determinação do tamanho do arquivo e dos parâmetros de FEC. Este processo, parte importante da metodologia considerada neste artigo, está detalhado nos itens a seguir.

5.1. Tamanho do arquivo

Existe um *trade-off* quanto ao tamanho do arquivo a ser usado nos experimentos. Por um lado, um arquivo de tamanho grande resulta em um experimento mais estável, pois em função da maior quantidade de tempo, a transferência sofre menos influência de picos no comportamento da rede. Além disso, com o passar do tempo, os protocolos se adaptam à rede, atingindo seu melhor desempenho. Transferências com arquivos pequenos sofrem muito mais influência de picos de rede, não permitindo uma estabilização tão eficiente dos protocolos. Dessa forma, seus resultados geram um desvio padrão alto e não correspondem ao mesmo comportamento das transmissões longas, foco principal dos protocolos multicast.

No entanto, a desvantagem de se utilizar arquivos de tamanho grande é o tempo excessivo consumido na duração do experimento, muitas vezes não sendo possível a realização da quantidade desejada de experimentos no tempo hábil disponível devido a baixas taxas de transmissão. Como existem muitos parâmetros que precisam ser avaliados, são necessárias várias repetições devido à validação estatística dos dados.

Enfim, são necessárias transmissões com arquivos de tamanho suficientemente grande para que não ocorram flutuações de resultados em função de picos eventuais na rede, que ocorram por curtos períodos ou de pequena intensidade. Dessa forma, foram realizadas avaliações variando-se apenas o tamanho do arquivo, objetivando um melhor desempenho dos protocolos multicast confiável.

O resultado obtido está ilustrado na Figura 2, que apresenta no eixo x o tamanho do arquivo, e no eixo y o *goodput*, em Mbytes e Mbps, respectivamente. Foram avaliados quatro tamanhos: 1, 10, 25 e 100Mbytes, para quatro protocolos. Fica claro, pelas curvas de desempenho do gráfico, que a partir de um tamanho de arquivo em torno de 25Mbytes existe uma estabilidade em termos de *goodput* mesmo com o aumento significativo do tamanho do arquivo. A partir desse resultado especificou-se em 32Mbytes o tamanho de arquivo mínimo a ser transmitido, tendo o mesmo efeito de tamanhos maiores.

O cenário avaliado foi considerado apenas para descobrir qual seria o tamanho mínimo de arquivo que gerasse a menor quantidade de flutuações devido a picos na rede. Com isso, a configuração de rede apontava para uma combinação onde o TCP obtinha vantagens pois o número de clientes de recebimento era baixo. Sob este prisma, percebe-se que o tamanho do arquivo identificado foi para o melhor caso e para casos piores os arquivos poderiam ser menores, pois o tempo de transmissão seria maior. Esse gráfico não deve ser considerado como medida de desempenho entre protocolos e sim como medida de desempenho do tamanho do arquivo.

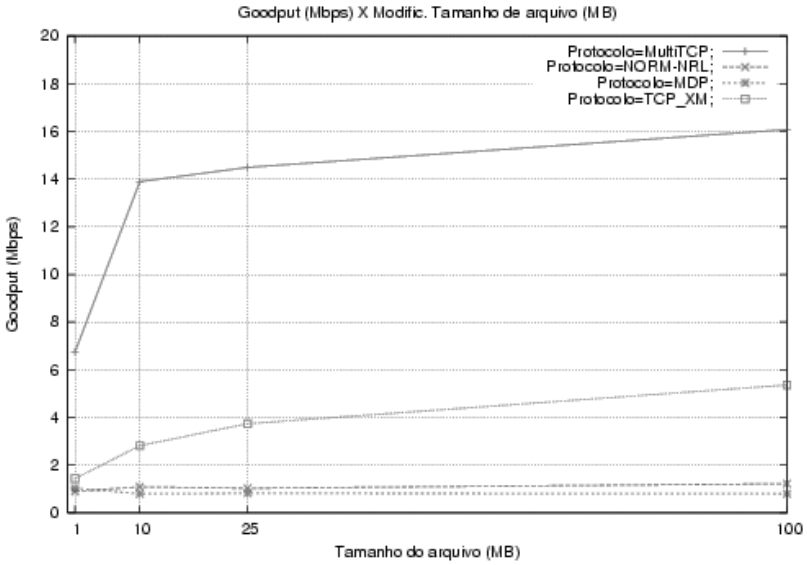

Figura 2: Experimento de tamanho de arquivo

5.2. Ordem das máquinas e velocidade de interconexão

Nos protocolos multicast confiável analisados, a transmissão adapta-se sempre ao receptor mais lento do grupo. Os experimentos foram realizados aumentando-se gradativamente a quantidade de receptores. Dessa forma, a ordem de inclusão dos receptores utilizada é crucial no desempenho resultante do protocolo. Assim, por exemplo, supondo que o PoP-A é o mais rápido, o PoP-B o segundo mais rápido, e assim por diante, o ideal é que o primeiro experimento seja feito somente com o PoP-A. O segundo experimento deve ser feito com o grupo formado pelo PoP-A+PoP-B. O terceiro com PoP-A+PoP-B+PoP-C, e assim por diante.

Para tanto, com cada PoP, individualmente, foram executadas 30 repetições de uma transferência de um arquivo de 32 Mbytes utilizando o protocolo TCP. Tais experimentos foram efetuados entre os PoPs utilizados, dois a dois. Assumiu-se que a velocidade encontrada através deste teste refletiria o máximo de banda utilizável no enlace para multicast.

Foram feitos experimentos com origem em diversos PoPs da RNP, incluindo PB, MG e RS. A Figura 3 apresenta os resultados de *goodput* na transmissão TCP a partir do PoP-PB. Nota-se estabilidade na transmissão apesar da baixa taxa obtida. Este baixo *goodput* também era esperado dada a capacidade do enlace de saída do PoP-PB na ocasião dos experimentos. Como pode-se observar, a saída do PoP é limitada em 8Mbit/s, limitando a transmissão para todos os outros PoPs.

Segundo a Figura 3, o enlace de saída do PoP-PB é o fator limitante, desconsiderando-se o PoP-AM, pois mostra a capacidade TCP do enlace do PoP para com cada um dos PoPs presentes no gráfico. Pode-se notar que somente o PoP-AM possui um enlace inferior. Com isso, a linha apresentada no grádico representa uma 'linha superior' de desempeho que, teoricamente, não pode ser ultrapassada, pois para transmissões em grupo multicast, o *goodput* final sempre fica preso ao menor enlace disponível no sistema.

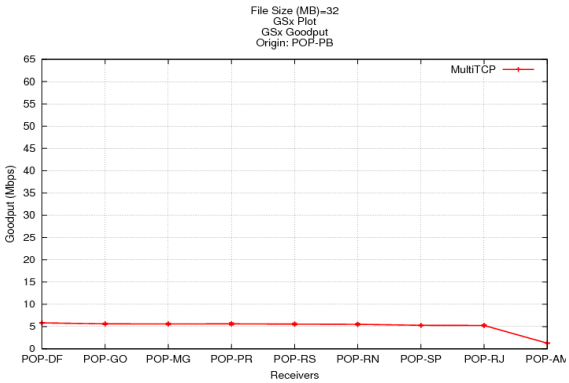

Figura 3. Transmissão unicast de arquivo de 32Mbytes a partir do PoP-PB.

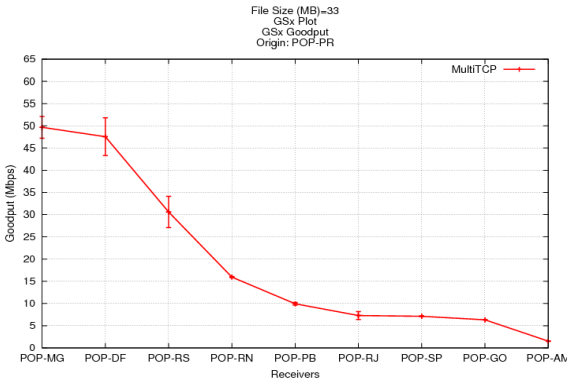

Figura 4. Transmissão unicast de arquivo de 32Mbytes a partir do PoP-PR.

Já para o transmissor localizado no PoP-PR, a realidade é diferente, como ilustrado na Figura 4. Neste gráfico, pode-se observar que o enlace do PoP-PR pode suportar no mínimo 50Mbps. Os dois pontos mais contrastantes do gráfico representam os enlaces disponibilizados pela RNP nos PoPs de RJ e SP. O primeiro tem enlaces de cerca de 622Mbps, no entanto, a máquina estava conectada em um switch de 10Mbps. Já o baixo *goodput* do PoP-SP está ligado a existência de filtros. Contudo, esse gráfico mostra que o PoP-PR não possui um enlace limitante no sistema e, com isso, tem um bom desempenho (*goodput*) com TCP. Já em termos de multicast confiável, será vantajoso sobre o TCP apenas, em termos de *goodput*, quando o grupo possuir um número razoável de componentes.

5.3. Parâmetros de FEC

Quatro, dos seis protocolos analisados, utilizam FEC como técnica de controle de erro, com o objetivo de diminuirem o número de retransmissões e minimizar os pacotes de *feedback* na rede. Tais mecanismos variam quanto aos seus parâmetros de entrada. Com o objetivo de descobrir quais parâmetros de FEC resultam em um melhor desempenho (em termos de *goodput*), foi efetuado um estudo variando-se exclusivamente métricas relacionadas ao FEC para descobrir a melhor combinação para cada um dos protocolos. Os demais parâmetros permanecem inalterados, pois a investigação em mente foca-se apenas em FEC visando diminuir ao máximo as variações causadas por outros fatores não relacionados.

Para a realização do estudo de FEC, dividiu-se os receptores em dois grupos, para mostrar um ambiente de tráfego rápido e outro lento, baseando-se em medições realizadas com TCP para avaliação da capacidade dos enlaces envolvidos. Dentre os principais parâmetros que permanecem fixos encontram-se o tamanho do arquivo em 32 Mbytes, tamanho de *buffer*, controle de congestionamento (ativado), *Time to Live* (em 10), NACKs enviados apenas para o transmissor (ativado). Nesse estudo não fornece-se uma taxa de envio, pois com o uso de controle de congestionamento a taxa é adaptativa.

Cada protocolo que disponibiliza a utilização de FEC para transmissão tem seus parâmetros específicos. TCP-XM e MultiTCP não fazem uso de FEC. Para o protocolo NORM desenvolvido pelo INRIA o único parâmetro disponível é o “--fec” que informa a quantidade de transmissão pró-ativa que o protocolo utilizará. Os protocolos NORM e MDP desenvolvidos pelo NRL compartilham alguns dos parâmetros. Os parâmetros em comum são: “-segmentSize” que indica a quantidade de dados úteis por pacote; “-block” que indica a quantidade de pacotes por bloco de transmissão; “-parity” que indica a quantidade de pacotes de paridade gerado por bloco (apenas gerado e não transmitidos); “-auto” indica a quantidade de pacotes de paridade enviados pró-ativamente por bloco (esse número deve ser menor ou igual a “-parity”). O NORM ainda possui um parâmetro chamado “-extra” que indica a quantidade de pacotes de paridade enviada aos transmissores quando ocorre uma perda.

Uma vez determinados os parâmetros fixos, buscou-se a variação dos parâmetros de utilização de FEC. Dentro desse cenário ainda existem dois casos: um onde se utiliza FEC e outro onde o mesmo não é utilizado. Assim, variou-se os parâmetros de tamanho do bloco de dados utilizados para cálculo de paridade, a quantidade de pacotes de paridade gerados, a quantidade de pacotes de paridade enviados pró-ativamente e a quantidade de pacotes de paridade enviados em caso de perdas (NACK).

A partir daí se realizou-se uma gama de experimentos com os grupos no intuito de descobrir a melhor alternativa de utilização dos parâmetros de FEC com os protocolos disponíveis.

6. Resultados dos Experimentos

A seguir são detalhados os resultados obtidos através das medições. São apresentados os resultados tendo-se o PoP-PB como transmissor, pois este foi um dos PoPs mais estáveis durante o período de avaliação. Inicialmente, são visualizados os resultados de sobrecarga na rede. Posteriormente, é analisado o resultado obtido em termos de *goodput* para taxa fixa e, logo depois, para taxa adaptativa. Por fim, é apresentado o resultado de robustez dos protocolos.

6.1. Sobrecarga da rede

O conceito de sobrecarga de rede adotado é a respeito da quantidade de dados a mais que o desejado trafegam na rede a fim de completar a transmissão. Estes dados podem ser de cabeçalhos, pacotes de controle, retransmissão de tráfego perdido ou múltiplas transferências do mesmo arquivo no caso do MultiTCP. A Figura 5 mostra, conforme esperado, a sobrecarga do tráfego TCP para diversos destinos em comparação à sobrecarga apresentada pelos protocolos de multicast confiável. O MultiTCP cria tantos tráfegos independentes quanto forem o número de receptores, ou seja, há controles individuais para cada receptor. Os dados (um arquivo, por exemplo) serão então transmitidos o mesmo número de vezes que a quantidade de receptores. Já no caso do tráfego multicast os dados desejados serão transmitidos idealmente somente uma vez e a retransmissão de tráfego perdido pode ser feita somente para um destino específico, dependendo do protocolo.

O resultado obtido foi consistente para todos os PoPs analisados, e todos mostraram o mesmo padrão de saída.

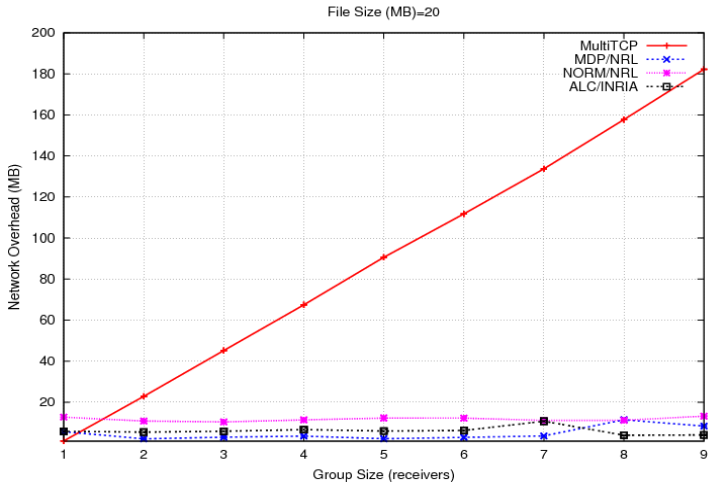

Figura 5: Sobrecarga de rede a partir do PoP-PB

6.2. Goodput com taxa fixa

O experimento demonstrado na Figura 6 foi realizado a partir do PoP-PB com destinos em SP, PR, RN, GO, MG, RJ, AM, RS e DF. A queda no *goodput* verificada nesta figura quando

o grupo de receptores chegou a sete deve-se à limitação física no enlace do PoP-AM. Este PoP possui quatro enlaces de 2 Mbps, sendo que a capacidade de transmissão real verificada foi de 1,78 Mbps.

Observa-se na Figura 6 que os protocolos multicast superaram o MultiTCP, mesmo tendo somente um receptor. Isso ocorreu devido ao fato do protocolo MultiTCP fazer uso de controle de congestionamento e, como a taxa de utilização do enlace era considerável no momento de realização dos experimentos, a taxa de transmissão com controle de congestionamento limitou-se a 3Mbps. Já fazendo uso dos protocolos multicast confiável, o envio é realizado através de múltiplos pacotes UDP sem controle de congestionamento, impondo seu tráfego na rede.

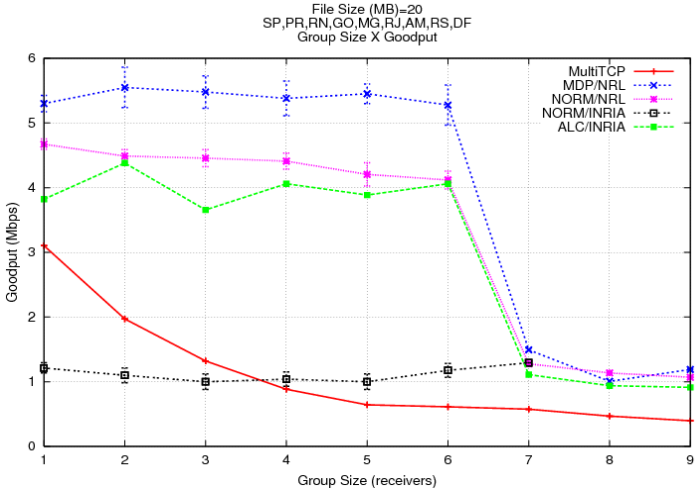

Figura 6: Experimento de taxa fixa a partir do PoP-PB

6.3. Goodput com taxa adaptativa

Nos experimentos com taxa adaptativa, configurou-se os parâmetros dos protocolos de forma que os mesmos modificassem sua taxa de transmissão de acordo com o congestionamento da rede. As Figuras 7 e 8 mostram o *goodput* obtido a partir do PoP-PB em horários de alto tráfego e baixo tráfego, respectivamente, com destinos em DF, PR, MG, RS, RJ, GO, AM e SP. Observa-se aqui que, à medida que aumenta o número de receptores, os protocolos baseados em multicast passam a ser mais vantajosos que os protocolos baseados em TCP unicast. Quanto maior o número de receptores, maior a vantagem observada nos protocolos multicast.

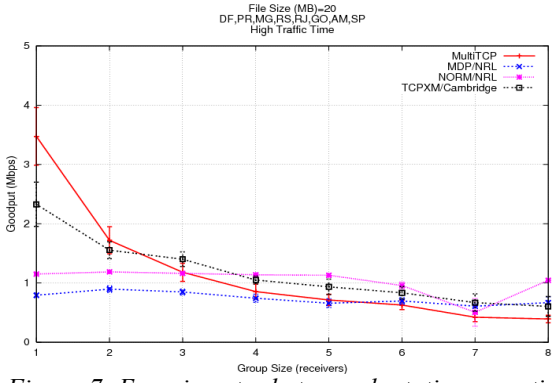

Figura 7: Experimento de taxa adaptativa a partir do PoP-PB em horário de alto tráfego.

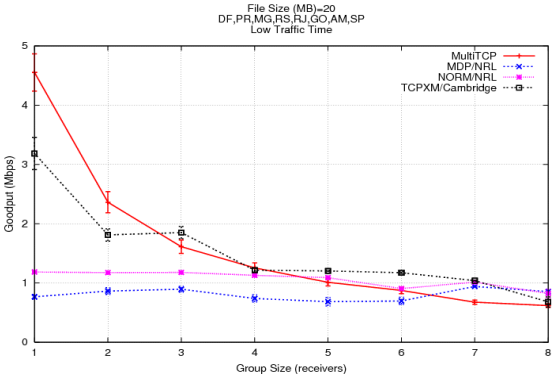

Figura 8. Experimento de taxa adaptativa a partir do PoP-PB em horário de baixo tráfego.

6.4. Robustez

A medição da robustez visa demonstrar o quão confiável é um protocolo. É importante contrastar esses resultados com o número de experimentos realizados, pois algumas vezes realizam-se poucos experimentos com um determinado cenário, acarretando uma diminuição drástica dessa métrica mesmo com o acontecimento de apenas uma falha.

Para o cenário fixo apenas o PoP-PB atuou como transmissor. A Tabela 2 demonstra a variação do sucesso dos protocolos MDP e NORM/INRIA (MCL). Esse acontecimentos podem ser causados por falhas temporárias na rede ou uma falha no protocolo em questão. Foram realizados 30 testes por protocolo (para cada número de receptores), totalizando aproximadamente 1100 experimentos.

Tabela 2. Robustez atingida nos experimentos.

Quantidade de	**Taxa Fixa - percentual sobre 30 experimentos**			
Receptores	**MPD/NRL**	**NORM/NRL**	**NORM/INRIA**	**ALC/INRIA**
1	100	100	100	100
2	100	100	100	98,5
3	100	100	100	100
4	100	100	100	100
5	100	100	85	100
6	100	100	100	100
7	100	100	82,5	100
8	63,5	100	100	100
9	100	100	40	100
Média	**95,94**	**100**	**89,72**	**99,83**
Quantidade de	**Taxa Adaptativa - percentual sobre 30 experimentos**			
Receptores	**MPD/NRL**	**NORM/NRL**	**TCPXM**	**MultiTCP**
1	100	100	100	100
2	100	90	100	100
3	100	85	100	100
4	100	80,5	95,5	100
5	100	95	91	100
6	100	90	95,5	100
7	100	90	82	100
8	91	80,5	83	100
Média	**98,88**	**88,88**	**93,38**	**100**

Como pode-se notar com os dados apresentados na Tabela 2, os níveis percentuais de confiabilidade foram altos, na maior parte acima de 90%, tendo como foco principal os protocolos multicast MDP e NORM.

7. A Interface FATMC

O objetivo dos experimentos efetuados foi investigar a aplicabilidade de multicast com a solução de transferência de arquivos na RNP. Em tal estudo foi empregada a ferramenta MURMET, destinada para realizar a avaliação dos protocolos. Embora tenha atingido seus objetivos como ferramenta de avaliação, a mesma não é destinada para o uso junto a usuários finais. Visando então uma aplicação para esse tipo de público, e com foco na tranmissão de arquivos com protocolos de multicast confiável de forma mais simplificada, foi desenvolvida a interface FATMC (Ferramenta de Agendamento de Transferência com Multicast Confiável).

A Figura 9 ilustra a arquitetura da aplicação desenvolvida. A interface gráfica é composta por duas partes: um *applet* (interface gráfica propriamente dita) e um *daemon* gerenciador do mesmo, que está rodando em um servidor Web.

Inicialmente, o usuário acessa um servidor Web e executa um *applet* em seu próprio navegador. Através do mesmo, o usuário configura diversos parâmetros da transmissão, como protocolo escolhido, arquivos a enviar e velocidade da transmissão. Após a configuração, o usuário inicia a transferência de arquivos. A partir desse momento, o *applet* se comunica com o transmissor (que foi configurado na interface), e este último inicia a transmissão em multicast confiável, fazendo uso do protocolo escolhido pelo usuário.

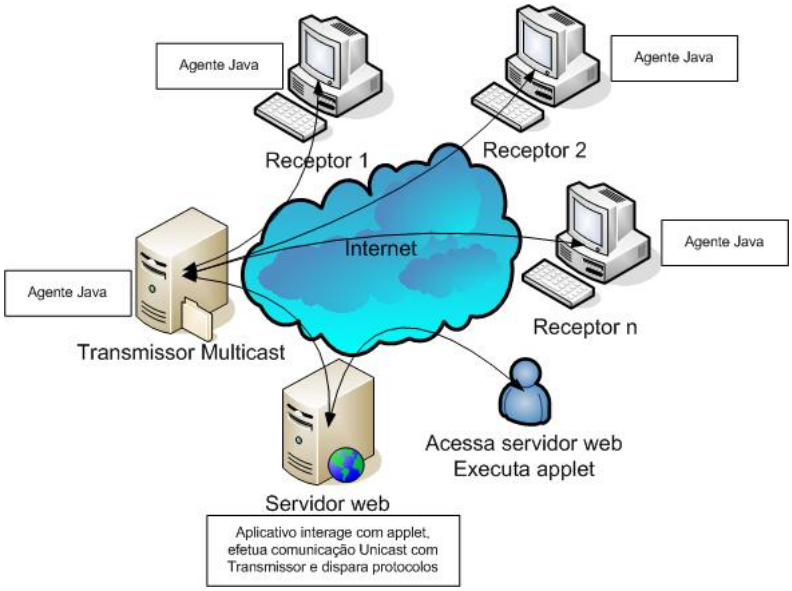

Figura 9: Arquitetura da interface desenvolvida

Visualmente, o aplicativo é dividido em duas partes (abas): agendamento de transferência e gerenciamento. Através da primeira aba, conforme a Figura 10, é feito o agendamento de transferências. Na área 1 dessa figura, o usuário seleciona o PoP transmissor, o arquivo a ser transmitido a partir do mesmo e o diretório de destino, comum a todos os receptores, onde o arquivo de origem será salvo.

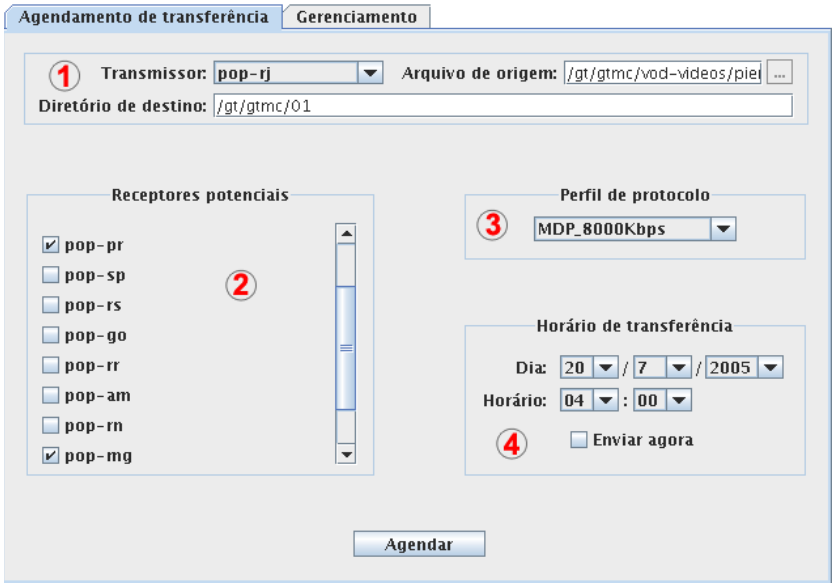

Figura 10: Interface com o usuário do mecanismo de transmissão multicast confiável

Em um segundo momento, na área 2, o usuário seleciona os receptores (pelo menos um), a partir da lista de PoPs restantes. Então, na terceira área, o perfil de protocolo a ser utilizado na

transferência é escolhido dentre os disponíveis (TCP-XM, MDP, NORM, MultiTCP). O último passo a ser efetuado é optar por iniciar a transferência imediatamente ou agendá-la para uma data posterior, conforme pode ser observado na área 4. Concluída esta etapa, a transferência pode ser acompanhada na tela de gerenciamento.

Na aba de gerenciamento, é possível visualizar o histórico das transferências executadas, em execução e aquelas aguardando para iniciar (agendadas para alguma data futura). Além disso, é possível obter, para cada transferência, as seguintes informações: data de início, duração, perfil de protocolo utilizado, IP do transmissor, lista de IPs dos receptores e informações sobre o status da mesma. Enquanto uma transferência não termina, o estado da mesma é atualizado no applet, de acordo com o intervalo de tempo determinado pelo usuário. O usuário pode também, nessa mesma aba, abortar uma ou mais transferências que ainda não tenham terminado.

8. Considerações Finais

Pelos resultados atingidos, os protocolos NORM-NRL e MDP demonstraram ser capazes de realizar transmissões contínuas com alto volume de dados em uma rede sem qualquer reserva de banda ou QoS para tal. Tais protocolos apresentaram taxas de sobrecarga na rede significativamente menores que os protocolos TCP-XM e MultiTCP, mesmo na transmissão para grupos pequenos, de somente dois receptores. Para grupos maiores, a diferença ficou mais evidente, principalmente no gráfico apresentado na Seção 6.1.

É importante ressaltar a inexistência de estudos práticos a respeito de protocolos multicast confiável em âmbito nacional. As implementações existentes ainda têm muito a evoluir em termos de desenvolvimento, mas para fins de transmissão de arquivos em uma rede proprietária, onde os administradores têm acesso a todos roteadores para configuração de multicast, o uso de transmissões multicast confiável pode seguramente ser realizado de forma contínua, gerando ganhos significativos em termos de economia do uso de largura de banda da rede como um todo.

Mesmo em situações com dificuldades externas de operação, os protocolos de multicast confiável NORM-NRL e MDP demonstraram maior capacidade de transferência de dados com menor sobrecarga em relação aos outros protocolos avaliados. A rede utilizada apresentou diversas dificuldades durante a execução dos experimentos. Destaca-se inconstâncias na taxa de transmissão verificadas ao longo de diversos dias, máquinas com capacidade de saída de tráfego diferentes entre comunicações TCP e UDP com prejuízo para UDP, utilizado pelos protocolos multicast, e enlaces fora de operação com certa freqüência. A configuração da rede em si, contemplando a utilização de maior tráfego UDP, é indispensável para a obtenção de melhores resultados.

As ferramentas desenvolvidas, FATMC e MURMET, possuem características de configuração e modularização que possibilitam seu uso em diferentes ambientes de rede, sendo uma contribuição direta deste trabalho. Outros ambientes podem ter sua capacidade multicast avaliada pelo MURMET apenas adequando seu arquivo de configuração. Determinado o resultado desta avaliação, o FATMC pode ser configurado a fim de aumentar a eficiência na transferência de arquivos nestes ambientes.

9. Referências Bibliográficas

Adamson, B. et al (2004a). **Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol.** IETF RFC 3940

Adamson, B. et al (2004b). **Negative-Acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Building Blocks.** IETF RFC 3941.

Barcellos, M. P. (2005) **Avaliação Experimental de Protocolos Multicast Confiável**. Anais SBRC 2005. Fortaleza.

Diot, C., et. al. (2000) **Deployment Issues for the IP Multicast Service and Architecture**. IEEE Network magazine. Special issue on Multicasting. January/February.

Iperf (2005). **Iperf website**. Disponível em http://dast.nlanr.net/projects/iperf. Último acesso em 15/07/2005.

Jeacle, K.; Crowcroft, Jon; Barcellos, M.; Pettini, S. (2005) **Hybrid Reliable Multicast with TCP-XM**. Proceedings ACM CoNEXT 2005, p.177-187.

Lemos, Guido; Leite, Luiz Eduardo; Batista, Thais Vasconcelos (2001). **DynaVideo - Um Serviço de Distrib. de Vídeo baseado em config. dinâmica**. Anais SBRC 2001. Florianópolis.

Luby, M.; Gemmell, J.; Vicisano, L.; Rizzo, L.; J. Crowcroft (2002a). **Asynchronous Layered Coding (ALC) Protocol Instantiation.** IETF RFC 3450.

Luby, M.; Gemmell, J.; Vicisano, L.; Rizzo, L.; Handley, M.; J. Crowcroft (2002b). **Layered Coding Transport (LCT) Building Block.** IETF RFC 3451 .

Luby, M.; Vicisano, L.; Gemmell, J.; Rizzo, L.; Handley, M.; J. Crowcroft (2002c). **Forward Error Correction (FEC) Building Block.** IETF RFC 3452.

Luby, M.; Vicisano, L.; Gemmell, J.; Rizzo, L.; Handley, M.; J. Crowcroft (2002d). **The Use of Forward Error Correction (FEC) in Reliable Multicast.** IETF RFC 3453.

Macker, Joseph P. (1999) **The Multicast Dissemination Protocol (MDP) Toolkit**. Proc. IEEE. MILCOM. pp. 626-630.

Macker, Joseph; Dang, W. **The Multicast Dissemination Protocol version 1 (MDPv1) Framework**. Naval Research Laboratory. Technical white paper. 1996.

Mankin, A.; Romanow, A.; Bradner, S; V. Paxson. **IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols.** IETF RFC 2357, June 1998.

NEKOVEE, Maziar ; BARCELLOS, M. P. ; DAW, Michael . **Reliable Multicast for the Grid: A Case Study in Experimental Computer Science**. Phylosophical Transactions Of The Royal Society Of London, London, v. 363, n. 1833, p. 1775-1791, 2005.

Rangan, P. V.; Vin, H.M.; Ramanathan, S. (1992) **Designing an On-Demand Multimedia Service**. IEEE Communications Magazine, July.

Roca, V. (2005) **INRIA-Rhone-Allpes MCLv3 Project website**. Disponível em http://www.inrialpes.fr/planete/people/roca/mcl/mcl.html. Último acesso em 07/07/2005.

Vicisanno, L.; Rizzo, L.; Crowcroft, J. (1998) **TCP-like congestion control for layered multicast data transfer**. Proceedings of IEEE INFOCOM, San Francisco.