

# Distribuição de Conteúdo Multimídia em Tempo Real com Transporte de Fluxos Controlados e Não Confiáveis entre Pares

Leandro M. de Sales<sup>1</sup>, Hyggo O. de Almeida<sup>2</sup>, Angelo Perkusich<sup>2</sup>, Rafael A. Silva<sup>1</sup>

<sup>1</sup>Instituto de Computação  
Universidade Federal de Alagoas (UFAL)  
57.072-970 – Maceió – AL – Brasil

<sup>2</sup>Centro de Engenharia Elétrica e Informática  
Universidade Federal de Campina Grande (UFCG)  
58.429-140 – Campina Grande – PB – Brasil

{leandro,rafaelsilva}@ic.ufal.br, hyggo@dsc.ufcg.edu.br, perkusic@dee.ufcg.edu.br

**Resumo.** *A transmissão de conteúdo multimídia em tempo real através da Internet é de fundamental importância para as diversas aplicações atuais como voz sobre IP, videoconferência, jogos e TV pela Web. Os protocolos de transporte mais conhecidos - TCP e UDP - não são efetivos quando se deseja transmitir dados destas aplicações. Neste sentido, a IETF tem trabalhado em novos protocolos de transporte que aumentem a qualidade destas aplicações multimídia. Dentre estes novos protocolos, o DCCP (RFC 4340) é efetivo em transmissões de conteúdo multimídia através da Internet. Todavia, o DCCP não é efetivo em cenários onde existam muitos nós receptores e apenas um nó transmissor. Portanto, este artigo propõe o MU-DCCP, uma extensão do protocolo DCCP capaz de permitir transmissões de dados multimídia de 1 para muitos nós e controle de congestionamento de tráfego não confiável. O MU-DCCP utiliza multicast quando disponível ou um fluxo unicast por cada rede de destino e compartilhamento de dados entre os nós receptores. Com o uso do MU-DCCP, constata-se uma redução considerável de congestionamento na rede.*

## 1. Introdução

A transmissão de conteúdos multimídia em tempo real através da Internet tornou-se uma necessidade em aplicações como Voz sobre IP (VoIP), Videoconferência, Jogos e WebTV. Aplicações deste tipo implementam mecanismos sofisticados para melhorar a qualidade dos fluxos de dados e utilizar de forma eficiente os recursos disponíveis da rede. Na prática, tais mecanismos são implementados em forma de protocolos de rede com o intuito de: (i) reduzir altos níveis de congestionamento na rede; (ii) manter a equidade entre diferentes fluxos transmitidos por diferentes sistemas finais conectados à rede; e (iii) garantir níveis mínimos de qualidade do conteúdo multimídia sendo transmitido.

Aplicações como as supracitadas estão ganhando cada vez mais espaço no que diz respeito a adeptos em suas utilizações, principalmente na Internet. Para se ter uma idéia deste crescimento, recentemente a empresa Cisco publicou um artigo [Leavitt 2010] onde apresentou uma previsão de que em 2014 o tráfego de vídeo será maior do que o tráfego P2P em 2009, o que correspondeu a 39 % do tráfego naquele ano. A empresa prevê também que em 2014, o tráfego de VoIP, vídeo e jogos na Internet alcançará a marca de 40 Exabytes/mês, quase 50 % do tráfego de dados total na Internet previsto para 2014. Embora já se saiba o potencial do modelo de serviço P2P, esta previsão demonstra o nível de aceitação dos usuários em compartilhar seus arquivos e obter mais dados disponibilizados por outros usuários. Recentemente, a Paramount Pictures publicou uma nota [Freak 2011] informando que passará a transmitir filmes através de grandes redes P2P, tais como o BitTorrent. A própria empresa BitTorrent anunciou que está trabalhando em uma aplicação P2P para transmissão de conteúdos multimídia em tempo real. Por

último, no final de março/2011, a Amazon anunciou [Amazon 2011] que também entrará na disputa por uma fatia deste tipo de serviço. A empresa disponibilizará um reprodutor de conteúdo multimídia completamente *online*, onde as pessoas poderão comprar músicas e vídeos *online* e em seguida reproduzi-los em tempo real, com os dados armazenados na infra-estrutura de computação nas nuvens da empresa.

Diante deste cenário, desenvolver protocolos de rede para dar suporte a estas novas formas de fornecer serviços multimídia através da Internet tem se tornado uma tarefa ainda mais complexa, pois exige-se uma harmonia entre os requisitos mencionados anteriormente em meio às mudanças no consumo e disponibilidade de recursos da rede. Os protocolos para a Internet precisam ser projetados para inferir o estado da rede e ao mesmo tempo tomar decisões de acordo com as mudanças detectadas de forma rápida. Do ponto de vista da camada de transporte da pilha TCP/IP, protocolos tradicionais de transporte como o UDP e o TCP não foram projetados para tal, sendo deixado a cargo dos desenvolvedores das aplicações implementarem seus próprios mecanismos para controle de congestionamento e de fluxos de dados sem garantia de entrega, particularmente no caso do uso do UDP, adotado na maior parte das aplicações multimídia. Nos casos em que se tentam utilizar TCP em aplicações multimídia em tempo real, o mesmo não apresenta um desempenho satisfatório porque se implementa controle de congestionamento e garantia de entrega de uma forma não adequada para aplicações multimídia com transmissão de dados em tempo real.

Visando melhorar este cenário, novos protocolos de transporte de dados para a Internet foram padronizados pela IETF, onde os principais foram o protocolo DCCP (*Data-gram Congestion Control Protocol*) [Kohler et al. 2006b] [Kohler et al. 2006c] e o SCTP (*Stream Control Transmission Protocol*) [Jungmaier 2003]. No contexto deste trabalho, o foco da pesquisa está no protocolo DCCP e, por este motivo, discussões acerca de outros novos protocolos de transporte, como o SCTP, serão omitidas.

### 1.1. Cenário de aplicação e escopo do trabalho

Neste trabalho, o cenário de aplicação analisado envolve a transmissão de dados multimídia em tempo real e o uso de algoritmos de controle de congestionamento durante a transmissão de dados visando compartilhar o canal de transmissão de forma equilibrada entre todos os nós presentes na rede. Este cenário geralmente envolve diversos desafios, tais como: (i) permitir que fluxos multimídia convivam com fluxos de dados de aplicações elásticas sem que estes últimos sejam degradados pelos primeiros – vasta utilização do protocolo TCP; e (ii) evitar perdas excessivas de dados por parte das aplicações multimídia em questão, pois, neste caso, não faz sentido retransmitir dados quando estes são perdidos.

Existem diversos exemplos de aplicações que podem ser aplicados no cenário investigado, principalmente quando se utiliza o modelo de serviço P2P, tais como:

- Aplicações para transmissão de conteúdo multimídia em tempo real de um nó da rede para um outro, ou para um conjunto de outros nós. Por exemplo, sites durante a transmissão de dados, *online* como o *livestream.tv* e *streamtheworld.com* permitem que um usuário transmita conteúdos multimídia do seu computador para milhares de outros usuários conectados à Internet;
- Aplicações de telefonia IP, tais como o Skype, principalmente considerando o modo de conversa em grupo;
- TV *Online*, por exemplo, transmissões através da Internet de jogos da copa do mundo ou do campeonato brasileiro de futebol; e
- Jogos, videoconferência 1-para-muitos e rádios *online*. Para estes exemplos, existem muitos usuários interessados por um mesmo conteúdo, porém poucos dados individualizados. Ou seja, nada muda em termos dos dados que cada nó receptor receberá do nó transmissor.

Uma característica fundamental compartilhada por todos os exemplos supracitados é que na transmissão de dados sempre existe um nó gerador do conteúdo e milhares de nós receptores ( $1 \rightarrow N$ ). Quando se projeta um protocolo de rede para este fim, deve-se levar em consideração esta característica da aplicação para que se possa obter resultados significativos na qualidade do fluxo de dados multimídia sendo transmitido e na efetiva utilização dos recursos da rede.

Neste contexto, as duas grandes questões que surgem são: (i) como fazer isto de forma padronizada e assim evitar que cada desenvolvedor de aplicação tenha que implementar este tipo de mecanismo na sua própria aplicação, o que aumentaria a sua complexidade e o seu gerenciamento do ciclo de vida; e (ii) o que pode ser aproveitado do que está disponível para atingir o objetivo da primeira pergunta.

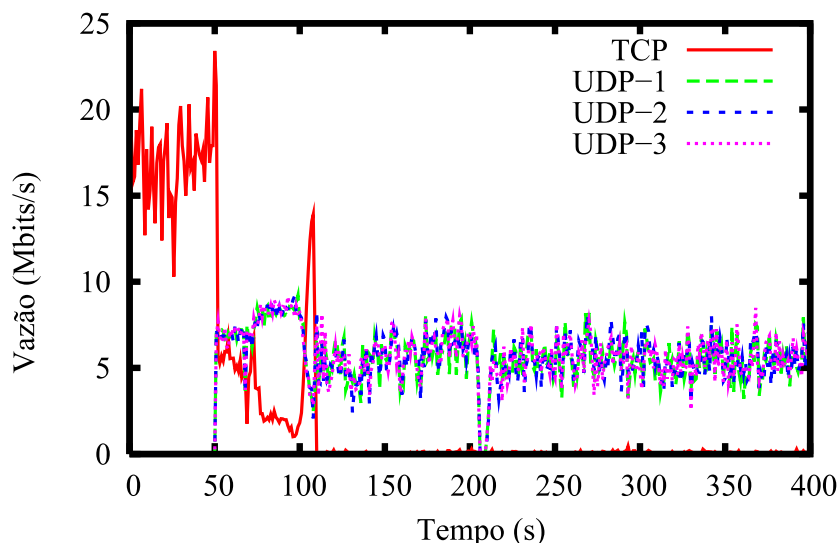
A seguir, será abordada uma discussão geral acerca da utilização dos protocolos de transportes da pilha TCP/IP quanto utilizados para transmissão de fluxos multimídia em tempo real, assim como detalhes e problemas do protocolo DCCP em transmissões multimídia em tempo real. Na Seção 3, será apresentada discussões sobre o problema tratado neste trabalho. Na Seção 4, será apresentado o MU-DCCP, uma versão multi(uni)cast do protocolo DCCP para distribuição de conteúdos multimídia em tempo real e com suporte a controle de congestionamento e compartilhamento de fluxos multimídia entre pares. Na Seção 5 será apresentada a metodologia utilizada nesta pesquisa e os resultados obtidos com discussões relacionadas. Por fim, na Seção 6 serão apresentadas as conclusões desta pesquisa.

## 2. Visão geral do TCP, UDP e DCCP em cenários de aplicações multimídia em tempo real

O protocolo UDP tem sido largamente utilizado em aplicações multimídia em tempo real por ser um protocolo leve, fazendo uso apenas do serviço de melhor esforço do IP para transmitir dados na Internet. Com o passar dos anos e antes do DCCP, o UDP se tornou a primeira e única opção para transmissão de dados multimídia em tempo real, porém gerando diversos efeitos colaterais nas grandes redes, os quais são vastamente discutidos na literatura[Floyd et al. 2006, de Sales et al. 2008b, de Sales et al. 2008a].

Para se ter uma idéia dos efeitos colaterais gerados na rede com o uso do UDP, observa-se o gráfico *vazão*  $\times$  *tempo* ilustrado na Figura 1. Este gráfico corresponde a um experimento realizado com a transmissão de 1 fluxo TCP competindo com 3 fluxos de áudio UDP em uma rede *Ethernet* de 100 *Mbps*. Observe que o UDP sempre ocupa o máximo da largura de banda disponível na rede ao passo que não oferece chances para outros fluxos utilizarem o canal, como é o caso do TCP. Por este motivo, o UDP sempre se apresenta com altas taxas de perda de pacotes, sobretudo quando há congestionamento na rede. No caso deste experimento, nos primeiros 50 s, quando não disputava com nenhum outro fluxo, o fluxo TCP utilizou a rede de forma satisfatória, alcançando uma vazão em torno de 20 *Mbps*. Entretanto, após esta fase, quando os três fluxos UDP foram transmitidos na rede, a vazão do fluxo TCP reduziu praticamente para 0, permanecendo assim até o final do experimento.

O protocolo TCP, por sua vez, atende de forma satisfatória as aplicações que toleram atrasos na entrega de dados e que exigem que estes sejam todos entregues corretamente e em ordem (aplicações elásticas). Porém, em se tratando de transporte de dados multimídia em tempo real, o TCP se torna o protocolo menos apropriado para este fim, pelo menos comparando-o com o UDP e o DCCP. Nas aplicações de fluxo multimídia em tempo real, é preferível manter o fluxo de dados e reproduzir o conteúdo que chega a esperar do que a informação perdida ser retransmitida, mesmo diante do fato de que parte dos dados da aplicação terem sido perdidos. Ao utilizar o TCP, isto não é possível. O principal motivo é que o TCP implementa uma entrega confiável de dados adotando a abordagem de retransmitir qualquer pacote perdido quando: (i) nenhuma confirmação de recepção (ACK) de pacote for recebida dentro do intervalo de tempo de um temporiza-



**Figura 1. TCP × UDP. Após 50 s do início do experimento, o fluxo UDP ocupa toda a largura de banda disponível na rede.**

dor de retransmissão (RTO); e (ii) pela recepção de três pacotes contendo a confirmação de recepção do último pacote recebido corretamente. Esta estratégia acarreta em atrasos indesejáveis quando se trata de transmissão de dados multimídia em tempo real.

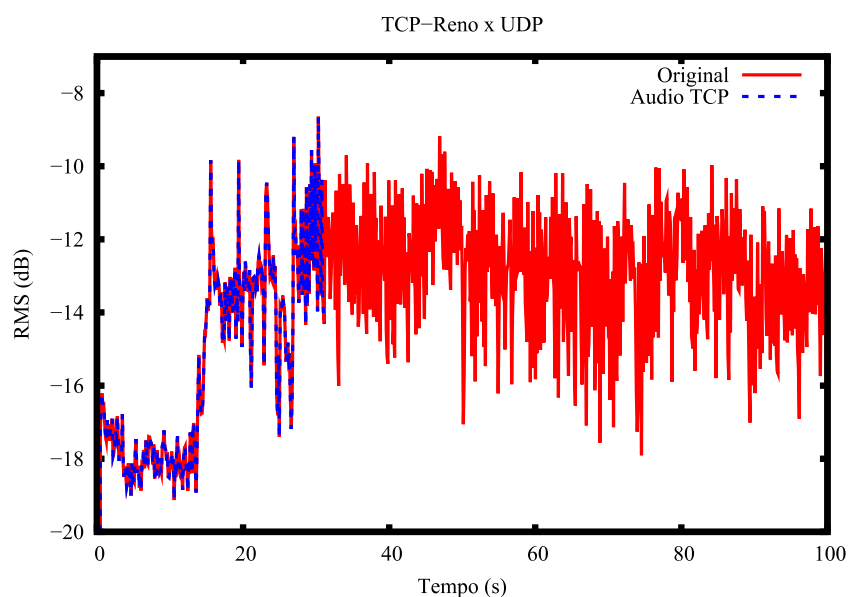
Em condições de congestionamento na rede, o atraso fim a fim aumenta e conseqüentemente degrada a qualidade do conteúdo multimídia sendo transmitido. Esta situação é agravada com a retransmissão de pacotes perdidos e que podem não fazer mais sentido à aplicação receptora devido ao comportamento transiente dos fluxos multimídias. Neste caso, se os pacotes de dados retransmitidos não alcançarem o receptor até um determinado instante, estes serão descartados. A consequência disso é o desperdício no uso dos recursos da rede, pois *buffers* dos roteadores são alocados para processar e repassar pacotes que, nestes casos, terminam sendo inúteis às aplicações.

O comportamento do TCP para situações como a mencionada anteriormente pode ser observado no gráfico ilustrado na Figura 2. No experimento realizado, transmitiu-se um áudio com duração de 100 s, armazenado no destino e em seguida comparado com o original. Neste caso, constatou-se que apenas 30 % do áudio alcançou o destino, fato ocorrido devido ao excesso de retransmissões de pacotes que foram perdidos na rede quando considerado fluxos TCP disputando com fluxos UDP.

## 2.1. O Protocolo DCCP

Com apenas essas duas opções para transporte de dados na Internet e objetivando promover melhorias nos serviços oferecidos pelas aplicações multimídia, a IETF aprovou a especificação do protocolo DCCP para transporte de dados multimídia para Internet. Trata-se de um protocolo de rede localizado na camada de transporte da pilha TCP/IP, tal como o TCP e o UDP. É um protocolo orientado à conexão, não garante entrega e nem ordenação dos dados transmitidos. Todavia implementa controle de congestionamento para transmissão não-confiável de fluxo de dados [de Sales et al. 2008b].

Sendo assim, o DCCP herda do TCP as características de ser orientado à conexão e fornecer controle de congestionamento. Do UDP, o DCCP herda as características de não garantir entrega e nem ordenação dos dados transmitidos. Além destas características, o DCCP também adiciona dois conceitos novos: a escolha tardia de dados e um arcabouço para gerenciamento dos algoritmos de controle de congestionamento de forma modular. A escolha tardia de dados permite a mudança de dados de um pacote mesmo depois que o mesmo já tenha sido enviado para a camada de transporte e ainda não tenha sido



**Figura 2. TCP Reno  $\times$  UDP, sendo o TCP enviando um arquivo de áudio.**

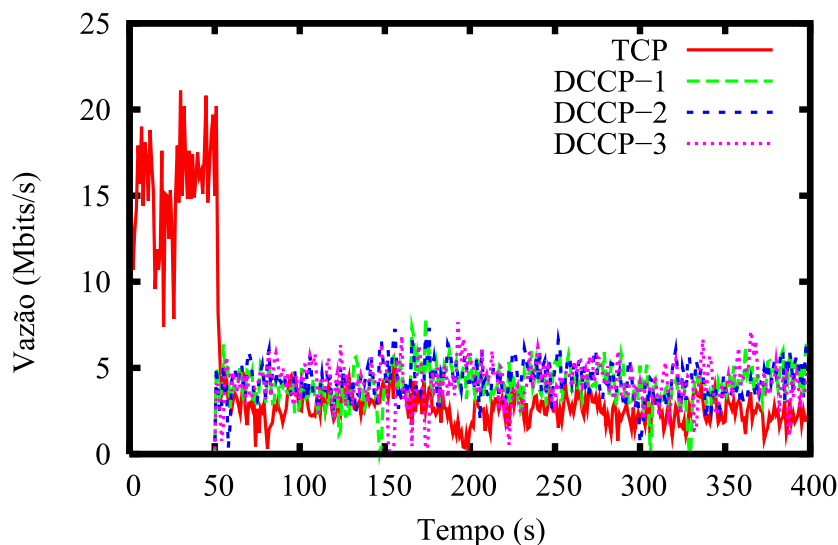
enviado através da rede. Já o arcabouço de gerenciamento de algoritmos de controle de congestionamento permite adicionar novos algoritmos de controle de congestionamento à aplicação e utilizá-los mesmo que uma conexão DCCP já tenha sido estabelecida.

Para entender as melhorias providas pelo protocolo DCCP, considere o gráfico *vazão  $\times$  tempo* apresentado na Figura 3. Neste gráfico, ilustra-se os comportamentos dos protocolos TCP e DCCP quando utilizados para transmissão de um arquivo e de um conteúdo multimídia, respectivamente. A partir do gráfico, é possível constatar que os protocolos TCP e DCCP compartilham entre si a largura de banda disponível, onde cada fluxo consegue transmitir dados na rede. Note que o comportamento do protocolo TCP para os 50 s iniciais foi similar ao confronto TCP  $\times$  UDP (Figura 1). Porém, diferentemente do que ocorreu naquele caso, após os primeiros 50 s dos confrontos TCP  $\times$  DCCP, a vazão do protocolo TCP continuou sendo satisfatória.

### 2.1.1. Arcabouço Modularizado de Algoritmos de Controle de Congestionamento

O DCCP oferece três algoritmos de controle de congestionamento, chamados CCIDs (*Congestion Control Identifiers*). Os CCIDs são os componentes responsáveis por oferecer o controle de congestionamento em conexões DCCP. No Linux, os CCIDs são módulos do *kernel* que funcionam sobre o núcleo da implementação do DCCP. Como tal, podem ser carregados e descarregados a qualquer momento, e as aplicações podem selecionar um CCID adequado para um determinado padrão de fluxo multimídia. Por exemplo, aplicações VoIP são caracterizadas pela transmissão de rajadas de pequenos pacotes seguidas de períodos de silêncio, enquanto aplicações de vídeo sob demanda geralmente transmitem conteúdo multimídia a taxas constantes. Nesse caso, para uma aplicação VoIP é melhor usar uma técnica de controle de congestionamento criada para VoIP. Esta idéia não se aplica em protocolos como o TCP e o UDP.

Os CCIDs padronizados são: CCID-2 [Kohler et al. 2006a], CCID-3 [Kohler and Handley 2006] e o CCID-4 [Kohler et al. 2007]. O CCID-2 (RFC



**Figura 3. TCP × DCCP. Ambos os protocolos conseguem transmitir dados na rede.**

4341) é melhor para aplicações que usam toda a banda de rede disponível e se adaptam a alterações súbitas de banda. Assemelha-se ao controle de congestionamento do TCP, que se baseia no conceito de janela de congestionamento. O tamanho dessa janela determina quantos pacotes o remetente tem permissão para enviar pela rede. Isso significa que quanto maior for a janela de congestionamento, mais pacotes o DCCP está autorizado a enviar dados pela rede.

Quando o CCID-2 detecta que um pacote foi perdido, ele diminui pela metade a janela de congestionamento. Tal ação caracteriza uma mudança abrupta na taxa de transmissão, principalmente para aplicações multimídia. No estado inicial de transmissão, a janela de congestionamento aumenta de forma exponencial conforme os pacotes enviados são confirmados, até alcançar a fase de contenção do congestionamento. Na fase de contenção, a taxa de transmissão aumenta linearmente até que aconteça o primeiro evento de perda de pacote.

O CCID-3 (RFC 4342) implementa um algoritmo de controle de congestionamento baseado no receptor, que controla a taxa do remetente. Periodicamente, o receptor envia ao remetente pacotes de informação relatando eventos de perda e outras estatísticas de conexão que são inseridas na equação do TFRC (*TCP-Friendly Rate Control*) (RFC 3448). O resultado desta equação corresponde à taxa de transmissão que o DCCP deverá utilizar para os próximos envios de pacotes.

Por último, a IETF recentemente publicou a RFC do CCID-4 (RFC 5622), própria para aplicações que enviam rajadas de pacotes pequenos intercalados por intervalos de silêncio. A escrita da especificação do CCID-4 também foi uma contribuição no contexto deste projeto, sendo as principais: melhoria do texto e implementação da especificação no kernel do Linux.

Para transmissões de dados multimídia em redes de computadores, onde satisfazer os requisitos de tempo pode definir o nível de qualidade da transmissão multimídia, o DCCP pode melhorar a qualidade do fluxo multimídia e ainda resolver diversos problemas de congestionamento da rede, como os causados por retransmissões desnecessárias de pacotes feitas pelo protocolo TCP ou por problemas de excessiva perda de pacotes quando se utiliza o protocolo UDP.

Sendo assim, a motivação do DCCP está relacionada com as características intrínsecas das aplicações com restrição de tempo de resposta e no fato de que grande parte

desse tipo de aplicação utiliza o UDP. Considerando os problemas e limitações dos protocolos TCP e UDP discutidos até aqui, o DCCP surgiu como uma das possíveis soluções para o seguinte problema-chave: de um lado apresenta-se o protocolo UDP, adotado por diversas aplicações com restrição de tempo e que não realizam controle de congestionamento. Por outro lado apresenta-se o protocolo TCP, cujas aplicações podem se tornar inutilizáveis devido ao congestionamento causado pelo UDP, além de realizarem retransmissões para garantir a entrega de dados.

Neste trabalho, apresenta-se o protocolo DCCP como uma opção efetiva a ser adotada para transmissão de dados multimídia, principalmente por ter um comportamento similar ao protocolo TCP e por ser um protocolo padronizado. Acredita-se que sua utilização trará diversas vantagens, pois melhora o uso dos recursos da rede, como é apontado nos gráficos apresentados. Todavia, ainda possui falhas críticas e que precisam ser corrigidas quando utilizados em larga escala.

### 3. O Problema do DCCP em Transmissões de Dados Multimídia em Tempo Real

Como foi possível observar, com o DCCP, obtém-se resultados animadores no ponto de vista de transmissão de dados multimídia e melhor utilização dos recursos de rede, resolvendo a maior parte dos problemas aparentes do TCP e do UDP nesse contexto. O problema se apresenta quando o protocolo DCCP é utilizado para distribuição de conteúdo multimídia sendo transmitido a partir de um nó da rede para vários nós localizados em redes distintas ( $1 \rightarrow N$ ).

Não é possível utilizar facilmente o protocolo DCCP para realizar esse tipo de transmissão, pois o DCCP é um protocolo orientado à conexão e portanto para cada novo usuário interessado em receber um fluxo multimídia transmitido com DCCP, uma nova conexão se faz necessária. As conseqüências desta limitação do DCCP são desastrosas, tornando-o um protocolo paradoxal para o que ele se propõe: resolver os problemas de congestionamento de rede gerados pelo protocolo UDP em cenários de aplicações multimídia. Entretanto, quando o protocolo é utilizado em larga escala, simplesmente não é efetivo. Este fato pode ser explicado pelos seguintes motivos:

1. **Excessivo consumo de recurso computacional:** para cada nova conexão, o nó transmissor deve alocar recursos computacionais (memória e processamento) para poder tratar cada nova conexão. Em cenários como os considerados nesta pesquisa, se muitos nós estão conectados em um único servidor, então isto elevará sobremaneira o consumo de recurso computacional do nó transmissor proporcionalmente a quantidade de nós receptores interessados pelo fluxo multimídia transmitido por um único nó na rede. Além disso, embora o conteúdo transmitido por um nó seja de interesse de muitos outros nós, os fluxos são enviados independentemente uns dos outros, o que gera duplicações desnecessárias e conseqüentemente desperdício de recursos de rede;
2. **A taxa de transmissão individualmente tenderá a 0:** O protocolo DCCP realiza controle de congestionamento utilizando uma equação matemática para definir a taxa de transmissão de uma conexão. À medida que mais nós se conectam a um nó transmissor, menor será a taxa de transmissão do nó transmissor para cada um dos nós receptores conectados a ele. Para a rede, esta estratégia é justa e evita que a mesma entre em colapso de congestionamento, mas para cada fluxo de dados isto é ruim. Este fenômeno é vastamente descrito na literatura e denominado de tragédia dos comuns [Hardin 1968]. A tragédia dos comuns ocorre neste caso porque se a taxa de transmissão de um determinado fluxo for muito pequena, pode ser que ela não seja suficiente para a recepção de um conteúdo multimídia em tempo real, que geralmente exige uma taxa de transmissão mínima para que a transmissão ocorra efetivamente. Estatisticamente, a taxa de transmissão de cada fluxo DCCP convergirá para um ponto de equilíbrio. Todavia, este ponto será mínimo, não

suficiente em aplicações consideradas neste trabalho, muito embora todos terão direitos iguais sobre o uso do canal. Este problema pode ser descrito matematicamente utilizando como base a Equação 1, que define cada taxa de transmissão  $X_i$  calculada pelo DCCP CCID-3. Nesta equação,  $X_i$  é a taxa de transmissão em bytes/segundo,  $s$  é o tamanho do pacote em bytes,  $R$  é o RTT em segundos,  $p$  é a taxa de ocorrência de perdas, entre 0 e 1,  $RTO$  é o valor do *timeout* de retransmissão do TCP em segundos e  $b$  é igual a 1 e representa o número máximo de pacotes confirmados por um único ACK. Considere-se o problema descrito anteriormente e que o uso total do canal para uma quantidade  $N$  de fluxos DCCP é definido por  $B = \sum_{i=1}^N X_i$ . Em condições severas de congestionamentos na rede, o valor de  $B$  é equivalente à largura de banda do canal de transmissão. Quando isto ocorre, tem-se que  $N$  atingiu um valor maior do que a rede suporta, fazendo com que os valores de  $p$  e  $R$  na Equação 1 também aumentem, e portanto tem-se que  $\lim_{N \rightarrow \infty} \frac{B}{N} = 0$ .

$$X_i = \frac{s}{R \times \sqrt{2 \times b \times \frac{p}{3}} + (RTO \times 3 \sqrt{3 \times b \times \frac{p}{8}} \times p \times (1 + 32 \times p^2))} \quad (1)$$

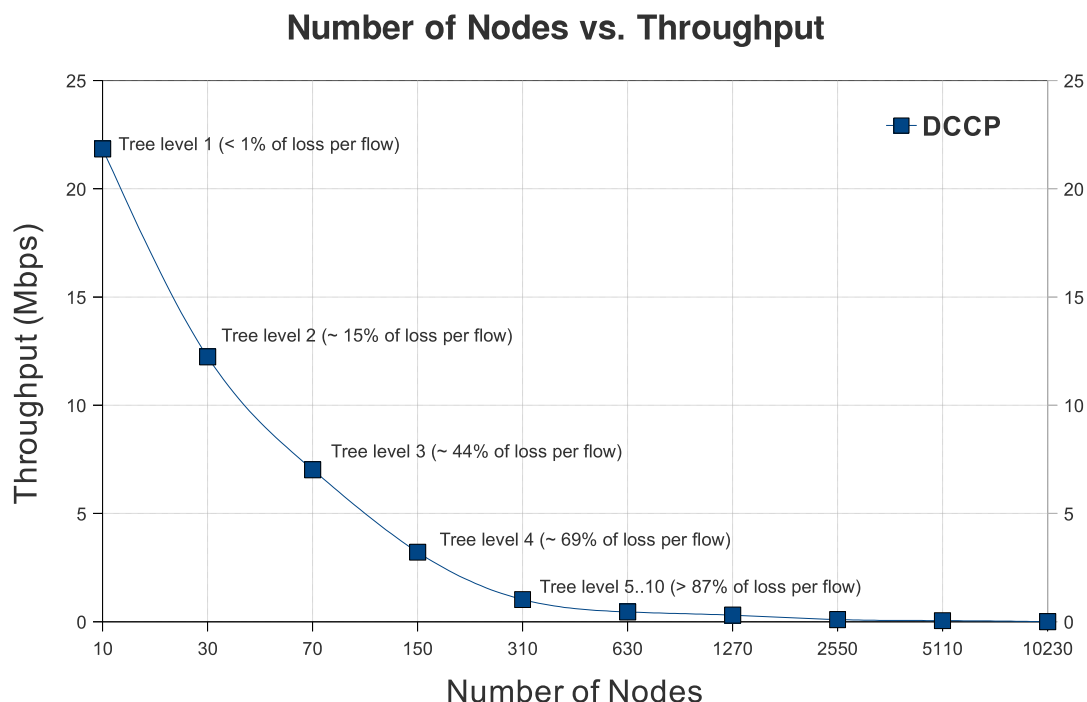
Neste contexto, foram realizadas pesquisas em busca de evidências mais contundentes de que este fato ocorre na prática. Por limitações de recursos computacionais para a realização de experimentos acerca do problema descrito, realizou-se simulações utilizando o NS-2 a fim de explorar os cenários estudados aqui. Foram executados 10 cenários de simulações de rede cuja topologia foi definida como uma árvore binária, onde cada ramo da árvore representava um roteador e cada roteador tinha 10 nós DCCP conectados a ele. Cada um dos 10 cenários foram repetidos  $n$  vezes até ser alcançado uma média com nível de confiança de 95%. Em cada cenário, aumentava-se o nível da árvore binária que definia a topologia da rede em 1. Por exemplo, o primeiro cenário tinha 10 nós receptores e 1 roteador, no cenário seguinte 30 nós e três roteadores, no seguinte 70 nós e 7 roteadores e assim por diante. O resultado das simulações estão resumidos no gráfico ilustrado na Figura 4. O eixo X do gráfico representa o número de nós receptores à medida que as simulações eram executadas, ao passo que o eixo Y é a taxa de transmissão média conseguida por cada conexão DCCP. A transmissão ocorreu da seguinte forma: um nó localizado na raiz árvore transmitiu o mesmo conteúdo multimídia para todos os outros nós conectados à rede, simulando uma típica transmissão multimídia  $1 \rightarrow N$  e um tráfego de comportamento equivalente a VoIP.

É possível observar no gráfico da Figura 4 que a vazão média de cada fluxo DCCP transmitido aos receptores tende a zero à medida que o número de receptores aumenta, sendo possível concluir que o protocolo DCCP não escala quando utilizado para transmissão de dados multimídia em cenários de aplicações com um transmissor transmitindo para vários receptores ( $1 \rightarrow N$ ).

Uma questão intrigante neste aspecto é que o protocolo DCCP funciona perfeitamente em cenários simplórios, mas sofre claramente de um problema de escalabilidade, o que é crítico para aplicações consideradas neste trabalho. Isto torna o protocolo DCCP inútil para cenários de distribuição de conteúdo multimídia, fazendo com que os desenvolvedores continuem sem motivações para efetivamente utilizar o protocolo DCCP em suas aplicações.

Sendo assim, o protocolo DCCP apresenta-se como uma alternativa ao UDP em cenários simplórios de transmissão multimídia, mas em situações de cenários reais de aplicações multimídia como costuma-se encontrar na Internet, o protocolo DCCP, pelo menos da forma que atualmente é empregado, não apresenta qualquer possibilidade de ser utilizado em cenários de aplicações multimídia mais complexos. Obviamente, é possível observar através do gráfico apresentado que quanto mais nós receptores forem adicionados na rede, mais acentuada será a perda de dados, causando a diminuição da vazão





**Figura 4.** Gráfico de uma transmissão DCCP com um transmissor enviando dados de áudio VoIP utilizando o protocolo DCCP para X receptores. A vazão média de cada fluxo tende a zero à medida que o número de receptores aumenta.

média alcançada por cada fluxo DCCP. Note que apenas fluxos DCCP foram transmitidos nesta simulação, os quais já foram necessários para causar o problema apresentado. Em situações mais realistas, o problema se torna ainda mais grave, pois o protocolo DCCP disputará o canal não apenas com outros fluxos DCCP, mas também com fluxos TCP e ainda com fluxos UDP.

Sendo assim, para cenários como este, os desenvolvedores de aplicações multimídia não tem outra opção a não ser continuar utilizando o protocolo UDP, porém ao custo do que já foi discutido na seção anterior; além disso, TCP não é aplicável neste cenário por diversas questões também já discutidas. Quanto aos desenvolvedores de aplicações multimídia em tempo real que decidem utilizar UDP e que desejam implementar um mecanismo de controle de congestionamento, estes atualmente se deparam com dois problemas:

1. Desenvolvimento de pelo menos um algoritmo de controle de congestionamento na camada de aplicação, o que aumenta na complexidade da aplicação;
2. Falta de padronização, cada desenvolvedor implementará seu algoritmo da forma que desejar.

Diante deste problema, utilizar UDP não parece ser a solução apropriada, porém é a que se utiliza atualmente. Uma abordagem neste sentido e vastamente considerada pelos pesquisadores para atender aos requisitos das aplicações consideradas neste trabalho é utilizar um modo de transmissão *multicast*. Nestes casos, todos os usuários passam a receber o mesmo conteúdo do fluxo de dados sendo transmitido ao custo de ser necessário apenas uma única transmissão de dados por parte do nó transmissor. Porém, a complexidade aumenta, no ponto de vista do desenvolvimento da aplicação, quando se implementa mecanismos de controle de congestionamento na camada de aplicação, além de quebrar a filosofia dos protocolos em camadas considerada pelo modelo de serviços TCP/IP. Neste caso, cada camada deve oferecer serviços para sua camada superior e, conseqüentemente,

usufruir de serviços da camada inferior. Sendo assim, o serviço de controle de congestionamento deve ser implementado na camada de transporte e não na camada de aplicação.

Portanto, embora o DCCP seja um protocolo promissor para transporte de fluxos multimídia, os desenvolvedores não têm motivos para passarem a utilizá-lo devido a falta de qualquer função deste protocolo que permita o desenvolvimento de uma solução sub-ótima para distribuição de conteúdos multimídia em tempo real não proporcional a transmissão de um novo fluxo para cada novo nó interessado em recebê-lo.

## 4. O Multi(Uni)cast DCCP

O *Multi(Uni)cast Datagram Congestion Control Protocol* (MU-DCCP) é uma extensão do protocolo DCCP para transmissão de fluxos de dados multimídia em cenários de um nó transmissor com muitos nós receptores. O MU-DCCP permite a transmissão de pacotes de dados com suporte ao controle de congestionamento de fluxos não confiáveis. O MU-DCCP pode operar em dois modos de transmissão: (i) multicast; e (ii) multi-unicast. Para cada um dos modos de transmissão providos pelo MU-DCCP, a aplicação primeiramente seleciona o modo desejado e um algoritmo específico é utilizado para realizar o estabelecimento de conexão e transmissão de dados entre os receptores envolvidos. O modo multicast é utilizado geralmente em redes locais, e o modo unicast é utilizado para que um nó, localizado na rede local, estabeleça uma conexão com um nó transmissor e distribua o conteúdo na sua rede local.

### 4.1. Visão Geral do MU-DCCP

Quando uma aplicação inicia um *socket* DCCP correspondente a conexão desejada, a mesma envia um pacote do tipo DCCP-MREQUEST com o campo TTL da camada de rede igual a 1. Este pacote é enviado em modo multicast para o endereço 239.255.255.251 e a porta 1900 (este *socket* é chamado de canal de controle do MU-DCCP). No cabeçalho do pacote DCCP-MREQUEST existem dois campos, um para especificar o endereço IP (32bits) do nó transmissor e o outro para especificar a porta (16bits) do nó transmissor, do qual o nó receptor deseja receber o conteúdo multimídia. Como o pacote é transmitido na rede local com TTL=1, isto significa que o pacote não será roteado para a rede externa e apenas o nós da sua rede local o receberá. Para facilitar a explicação, considere-se o endereço 200.200.211.5 e porta 8900 como sendo o *socket* do nó transmissor o qual o nó receptor tem interesse de se conectar. Neste caso, o nó receptor envia uma mensagem *multicast* do pacote DCCP-MREQUEST para o endereço *multicast* especificado anteriormente e com os campos IP e porta do pacote DCCP-MREQUEST preenchidos com os dados do *socket* do nó transmissor.

Por outro lado, todos os outros nós DCCP existentes na rede local devem, ao implementarem a extensão MU-DCCP, iniciar um *socket multicast* no endereço IP 239.255.255.251 e na porta 1900. Isto permitirá a recepção de pacotes DCCP-MREQUEST transmitidos por qualquer nó na rede. Sendo assim, caso um nó na rede local já esteja recebendo um fluxo multimídia DCCP vindo de 200.200.211.5 na porta 8900, quando receber um DCCP-MREQUEST responderá para o nó interessado um pacote do tipo DCCP-MRESPONSE, o que deve ocorrer em modo *unicast*. Ao responder com o DCCP-MRESPONSE, o nó receptor do fluxo original, passa a funcionar como um nó *relay* (DCCP Relay), retransmitindo na rede local os pacotes que estão recebendo do nó transmissor. No cabeçalho do DCCP-MRESPONSE, o nó especificará em qual modo ele repassará os pacotes para a rede local, se no modo *multicast* ou no modo *unicast*.

Note que, caso não exista nenhum nó do tipo DCCP Relay na rede local, o nó interessado em receber o fluxo pode enviar um pacote do tipo DCCP-MREQUEST, porém com TTL=2. Caso o roteador da rede local esteja roteando pacotes *multicast* no endereço IP e porta do canal de controle do MU-DCCP, é possível que um DCCP Relay responda com um pacote do tipo DCCP-MRESPONSE da forma que foi explicado anteriormente.

Caso o nó que enviar o DCCP-MREQUEST não receba nenhum pacote do tipo DCCP-MRESPONSE, o mesmo poderá tomar duas decisões, configurável via parâmetro de configuração da aplicação (setado via *setsockopt()*, por exemplo): (1) Enviar outro pacote do tipo DCCP-MREQUEST com o valor de TTL incrementado em 1; ou (2) iniciar uma conexão *unicast* com o nó transmissor. Caso ocorra a segunda opção, o nó MU-DCCP que iniciar uma conexão *unicast* com um nó transmissor qualquer deve se auto eleger para ser um DCCP *Relay* em pedidos de conexão através do pacote DCCP-MREQUEST que este venha a receber via o canal de controle do MU-DCCP.

#### 4.2. Modos de Transmissão do MU-DCCP

A mensagem de resposta de conexão enviada por um nó MU-DCCP, contém três campos: (i) o campo *multicast* (1 bit), se ativado o nó transmitirá os dados em modo *multicast*; (ii) o campo endereço IP, que especificará qual endereço IP (32 bits) o DCCP *Relay* passará a transmitir os dados; e (iii) o campo porta (16 bits), que especificará a porta que o DCCP *Relay* passará a transmitir os dados.

Note que um DCCP *Relay* pode decidir alterar o modo de sua transmissão a qualquer momento, bastando para isso enviar um pacote do tipo DCCP-MSYNC, via *multicast*, para o canal de controle do MU-DCCP, contendo a modificação desejada. Ao receber um DCCP-MSYNC, os nós envolvidos na recepção dos dados devem realizar as devidas modificações em seu modo de recepção. O formato do DCCP-MSYNC é igual ao formato do pacote DCCP-MRESPONSE.

#### 4.3. Controle de congestionamento com DCCP em modo multicast

O MU-DCCP suporta transmissão de dados tanto *unicast* quanto em modo *multicast*. Algoritmos de controle de congestionamento em modo *multicast* são mais complexos porque geralmente estes utilizam o *feedback* dos receptores para tomar decisões na definição da sua taxa de transmissão. Neste caso, nenhum CCID existente para DCCP foi projetado para funcionar neste modo, sendo necessário trabalhar em um novo CCID capaz de tratar todas as características de transmissão de dados de fluxo não confiável em modo *multicast*. No contexto deste trabalho, definiu-se o CCID-5, um algoritmo para controle de congestionamento de fluxos de dados não confiável em transmissões *multicast* utilizando o protocolo DCCP.

O foco principalmente de funcionamento do CCID-5 é fazê-lo capaz de ser executado em um DCCP *Relay* que esteja transmitindo em modo *multicast*. O CCID-5 determina uma nova taxa de transmissão baseado em relatórios enviados pelos nós receptores da sua rede local. Note que, como estão sendo tratados fluxos de dados não confiáveis, a solução para o CCID-5 pode fazer uso deste tipo de transmissão para reduzir a necessidade de confirmação de todos os pacotes recebidos pelos nós receptores. Como estão sendo considerados cenários onde existem apenas um nó transmissor dos dados e diversos nós receptores, receber confirmação de recepção de todos os nós receptores poderia inundar o nó transmissor de pacotes de controle, com confirmação de recepção de pacotes de dados por parte dos receptores.

Assim, foi possível desenvolver o CCID-5 como um algoritmo de controle de congestionamento que não precisa necessariamente receber um relatório de recepção de todos os nós receptores, mas sim de um sub-grupo de nós receptores, reduzindo de tal forma a quantidade de dados de controle transmitidos na rede em direção ao nó transmissor dos dados. É importante salientar que nem sempre o nó transmissor dos dados será aquele que gera o conteúdo, mas sim um nó transmissor que poderá executar o CCID-5 pode ser um nó do tipo DCCP *Relay*. Dado isto, desenvolveu-se um algoritmo para permitir a eleição de nós DCCP *Relays* e nós DCCP *Reporters*, responsáveis por relatar a um nó transmissor qualquer sua taxa de recepção de dados via *multicast*. Ao receber relatórios enviados pelos DCCP *Reporters*, os nós DCCP *Relays* ajustarão sua taxa de transmissão de acordo com estes relatórios, o que pode ser feito utilizando uma equação de cálculo de taxa de transmissão, como por exemplo a própria Equação 1.

#### 4.4. Eleição de Nós DCCP Relays e de DCCP Reporters

Os nós DCCP *Relays* são selecionados de duas formas: (i) serão DCCP *Relays* aqueles que iniciarem a primeira conexão *unicast* com um nó DCCP gerador dos dados, ou seja, o nó transmissor original; (ii) serão DCCP *Relays* aqueles que negociarem com algum outro nó DCCP *Relay* sua promoção para nó DCCP *Relay*. Note-se que este caso o nó que conceder a promoção de um nó DCCP *Relay* para outro deverá se rebaixar para um nó MU-DCCP normal. Além disso, quando um DCCP *Relay* conceder este *status*, o mesmo poderá se desconectar do nó DCCP gerador dos dados enviando um DCCP-MRESET, que conterà o endereço do novo nó DCCP *Relay*. É possível também que um nó DCCP *Relay* eleja outros nós DCCP *Relays* secundários, localizados na sua própria rede local. Esta funcionalidade é importante porque caso o atual DCCP *Relay* perca sua conexão ou desconecte do nó transmissor gerador dos dados, qualquer DCCP *Relay* secundário poderá assumir o papel de DCCP *Relay* primário. Neste caso, o nó que passar a assumir este papel deverá enviar um pacote do tipo DCCP-MELECT informando que assumirá a transmissão de dados outrora provida pelo nó DCCP *Relay* antigo.

Com relação aos nós DCCP *Reporters*, o processo de eleição funciona de forma similar, porém qualquer nó pode se tornar um DCCP *Reporter*. O processo funciona da seguinte forma: à medida que um DCCP *Relay* receber pacotes do tipo DCCP-MREQUEST, no pacote DCCP-MRESPONSE o nó DCCP *Relay* pode ativar a *flag* desse pacote informando que aquele nó deverá enviar relatórios de confirmação de pacotes. Desta forma, o DCCP *Relay* poderá limitar a quantidade de DCCP *Reporters* e assim receber relatórios apenas de um sub-conjunto de nós da rede. Como a transmissão é *multicast* e local, é possível ter poucos nós DCCP *Reporters* e continuar sendo efetivo no ponto de vista de determinar a melhor taxa de transmissão. No contexto deste trabalho, não realizou-se muitos experimentos na tentativa de se é possível determinar qual é a quantidade necessária de DCCP *Reporters* para se obter taxas de transmissões justas.

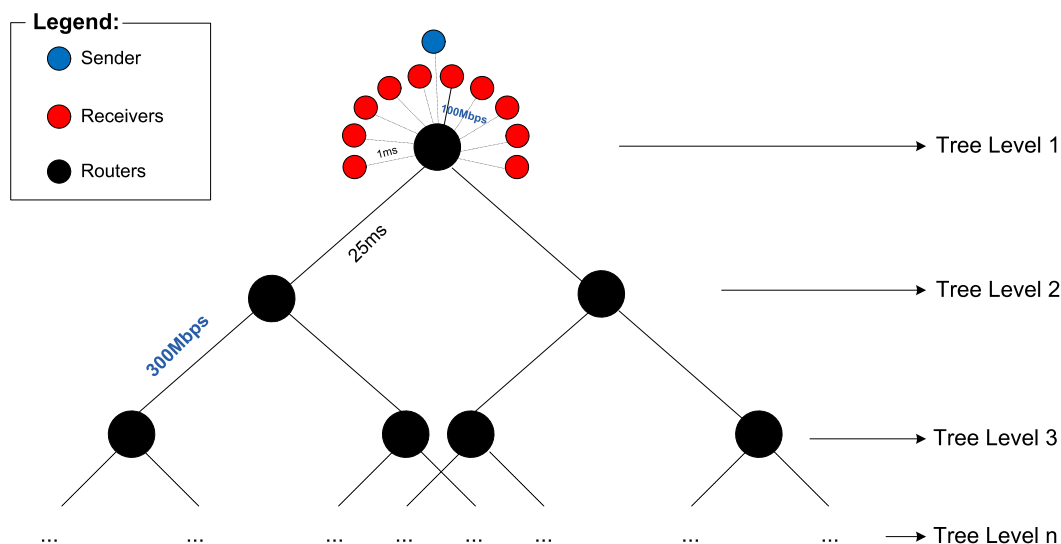
#### 4.5. Adaptação de Fluxo Multimídia

Uma funcionalidade peculiar do MU-DCCP é a capacidade que ele tem de fazer adaptação de fluxos multimídia de forma distribuída. A maioria das soluções para transmissão de dados multimídia, além de realizarem controle de congestionamento no nível de aplicação, realizam adaptação de fluxo multimídia na fonte geradora dos dados. Uma solução para adaptação de fluxo multimídia é transmitir os dados em diferentes canais, sendo que em cada canal transmite-se fluxos multimídia com uma determinada qualidade. Dependendo da qualidade desejada pelo nó receptor, ele solicita a transmissão em um determinado canal. O problema dessa abordagem é que o nó transmissor, necessariamente deve transmitir os dados em múltiplos canais, o que aumenta a complexidade da aplicação e a quantidade de fluxos de dados sendo transmitidos a partir do servidor. No MU-DCCP, é possível realizar a adaptação de fluxo de dados de forma distribuída, na prática, em cada DCCP *Relay*. Suponhe-se que existem duas redes adjacentes, rede 1 e rede 2. Considere que existe um nó DCCP *Relay* na rede 1 e entre esta e o nó transmissor a largura de banda disponível é de 100 *Mbps*. Caso a largura de banda disponível na rede 2 seja de no máximo 10 *Mbps*, um nó receptor na rede 2 teria que solicitar um fluxo multimídia em um canal diferente, considerando a solução supracitada. No caso do MU-DCCP é possível que um nó na rede 2 obtenha o fluxo multimídia através do DCCP *Relay* presente na rede 1, bastante, neste caso, que o DCCP *Relay* presente na rede 1 adapte o fluxo para o nó que esteja na rede 2. Desta forma, pode-se diminuir o tráfego na rede do nó transmissor e ainda sim permitir que nós em redes com baixa largura de banda consigam obter o fluxo multimídia adaptado.

### 5. Resultados e Discussões

Nesta seção são apresentados os resultados e discussões sobre um conjunto de simulações realizadas no contexto desta pesquisa utilizando o MU-DCCP. Nas simulações realizadas,

utilizou-se uma topologia de rede ilustrada na Figura 5 e a metodologia de execução das simulações igual a que foi descrita na Seção 3. A topologia da rede simulada segue a estrutura de uma árvore binária, onde as folhas de cada nó pai representam outras redes e assim sucessivamente. Os parâmetros de configuração de toda a rede foram definidos da seguinte forma:



**Figura 5. Topologia da rede definida para as simulações realizadas. Cada rede é representada por um roteador e com 10 nós em cada rede.**

- Número de computadores receptores por rede: 10
- Largura de banda da rede local: 100Mbps
- Latência da rede local: 1ms
- Largura de banda do backbone: 300Mbps
- Latência do backbone: 25ms
- Tamanho da fila dos roteadores do backbone: 3000 pacotes
- Duração da simulação: 900s

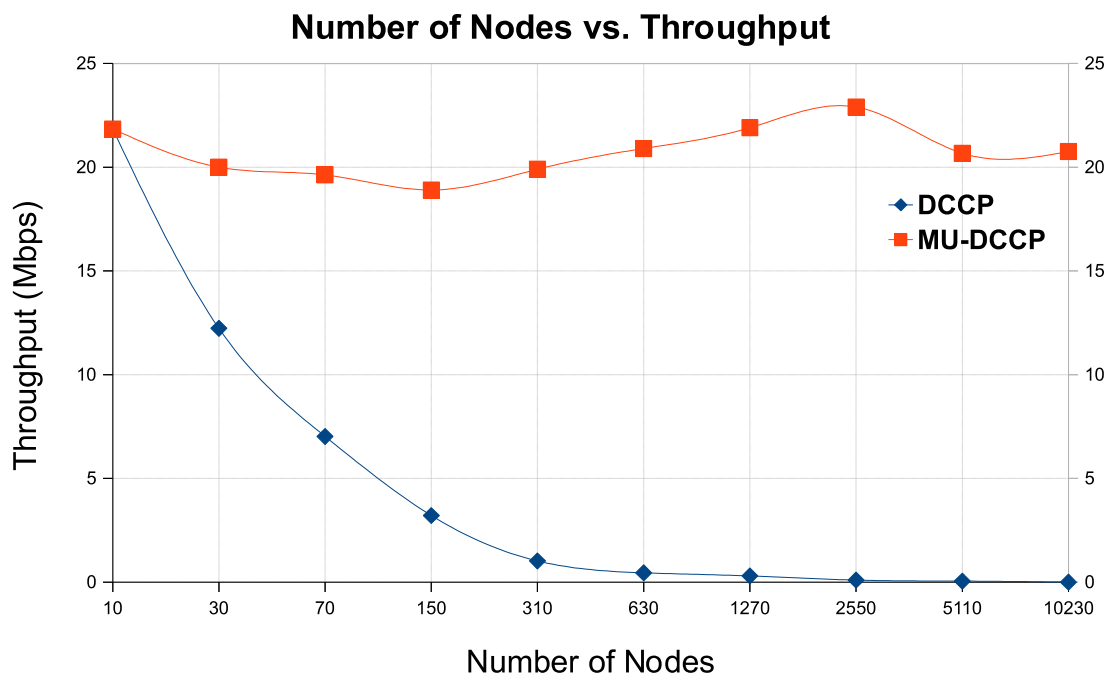
### 5.1. Discussões sobre os resultados

No gráfico ilustrado na Figura 6, apresenta-se o resultado das transmissões simuladas utilizando o NS-2 e o protocolo DCCP com a extensão MU-DCCP habilitada. Transmitiu-se o mesmo padrão de tráfego VoIP utilizado na Seção 3. É possível observar claramente que, quando se compara o gráfico da Figura 4 com o gráfico da Figura 6, o uso do MU-DCCP melhorou sobremaneira a taxa efetiva de recepção de dados por parte dos nós receptores.

Como está sendo considerada a taxa efetiva de recepção, constatou-se uma redução de mais de 90 % da taxa média de perda de pacotes por cada nó receptor, além de permitir uma distribuição adequada de fluxos de dados multimídia. É possível observar também que, mesmo aumentando a quantidade de usuários receptores, por exemplo, entre  $N=150$  e  $N=2550$ , a taxa média de recepção continuou sendo satisfatória, sendo crescente à medida que o valor de  $N$  aumentou. Isso pode ser justificado pelo fato de que quanto mais nós DCCP na rede, aumentam-se as chances de se tornarem nós DCCP Relays, o que conseqüentemente aumentam as chances de existirem mais nós receptores recebendo fluxos de dados via *multicast* e não somente via *unicast*.

## 6. Considerações Finais e Trabalhos Futuros

Neste artigo, apresentou-se os resultados da pesquisa sobre o MU-DCCP, uma versão modificada do protocolo DCCP com suporte a transmissão de dados multimídia em modo



**Figura 6.** Gráfico de uma transmissão DCCP com um transmissor enviando dados de áudio VoIP utilizando o protocolo DCCP e a extensão MU-DCCP para X receptores.

*multicast* quando possível ou *multi-unicast* quando *multicast* não for possível. Após os estudos realizados durante o desenvolvimento deste trabalho, algumas conclusões puderam ser obtidas acerca do uso do protocolo DCCP em transmissão de dados em cenários  $1 \rightarrow N$ : (i) o protocolo DCCP não se apresenta de forma satisfatória quando utilizado em cenários de aplicações em larga escala, por exemplo, em aplicações de distribuição de conteúdo multimídia partindo de um nó transmissor para diversos nós receptores; (ii) Fez-se necessárias soluções a fim de resolver este problema do protocolo DCCP e o MU-DCCP foi a resposta para isto. Após a definição e implementação do MU-DCCP, diversas simulações foram realizadas com o MU-DCCP, sendo os principais resultados e discussões apresentados neste artigo. No contexto do MU-DCCP também foi desenvolvido o CCID-5, um algoritmo de controle de congestionamento para transmissão de dados multimídia com suporte a *multicast*. De acordo com os resultados obtidos, observou-se uma melhoria significativa do uso do MU-DCCP em cenários de aplicações multimídia em tempo real.

Com estes resultados obtidos, o atual trabalho está sendo a escrita de um *Internet Draft* para posterior submissão como RFC promovida pela IETF. Como os resultados tem sido bastante animadores, espera-se que o MU-DCCP venha a ser adotado pelas principais aplicações multimídia no futuro. A proposta do MU-DCCP é importante porque ela abstrai toda a complexidade de transmissão de dados multimídia com suporte a controle de congestionamento e com suporte ao modo *multicast*, característica ausente na versão original do protocolo DCCP.

Como trabalhos futuros, pretende-se evoluir o MU-DCCP no sentido de melhorar o algoritmo de eleição de DCCP *Relays* e de DCCP *Reporters*. A melhoria desses algoritmos esta vinculada a execução de mais simulações em cenários diferentes do que o estudado e apresentado neste trabalho. Além disso, pretende-se executar simulações do MU-DCCP em transmissões de dados com outros padrões de tráfego, tais como vídeo e jogos e realizar um estudo mais aprofundado a respeito do atraso gerado à medida que os

nós receptores estão mais distantes do nó transmissor. Pretende-se ainda implementar o MU-DCCP no *kernel* do Linux e executar experimentos a fim de estudar o comportamento do MU-DCCP em cenários reais de rede.

## Referências

- Amazon (2011). Introducing Amazon Cloud Player for Web & Android. Online publication in Amazon.com. <http://www.amazon.com/b?ie=UTF8&node=2658409011>.
- de Sales, L. M., Almeida, H. O., Perkusich, A., and Jr., M. S. (2008a). An Experimental Evaluation of DCCP Transport Protocol: A Focus on the Fairness and Hand-off over 802.11g Networks. In *In Proceedings of the 5th IEEE Consumer Communications and Networking Conference*, pages 1149–1153.
- de Sales, L. M., Almeida, H. O., Perkusich, A., and Jr., M. S. (2008b). On the Performance of TCP, UDP and DCCP over 802.11g Networks. In *In Proceedings of the SAC 2008 23rd ACM Symposium on Applied Computing Fortaleza, CE*, pages 2074–2080.
- Floyd, S., Handley, M., and Kohler, E. (2006). Problem Statement for the datagram congestion control protocol (DCCP). <http://www.ietf.org/rfc/rfc4336.txt>. Último acesso: 12/04/2011.
- Freak, T. (2011). Paramount Pictures to Release Film on Bittorrent. Online publication in the Torrent Freak Website. <http://torrentfreak.com/paramount-pictures-partner-with-bittorrent-release-movie-110317/>.
- Hardin, G. (1968). The Tragedy of the Commons. *Science*, 162(3859):1243 – 1248.
- Jungmaier, A. (2003). *SCTP for Beginners*. Computer Network Technology Group, 1 edition.
- Kohler, E. and Handley, M. (2006). Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC). <http://www.ietf.org/rfc/rfc4342.txt>. Último acesso: 12/04/2011.
- Kohler, E., Handley, M., and Floyd (2006a). Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control. In *IETF Online RFC*. <http://www.ietf.org/rfc/rfc4341.txt>. Último acesso: 12/04/2011.
- Kohler, E., Handley, M., and Floyd, S. (2006b). Datagram Congestion Control Protocol (DCCP). <http://www.ietf.org/rfc/rfc4340.txt>. Último acesso: 12/04/2011.
- Kohler, E., Handley, M., and Floyd, S. (2006c). Designing DCCP: Congestion Control Without Reliability. In *SIGCOMM '06: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 27–38, New York, NY, USA. ACM Press.
- Kohler, E., Handley, M., and Floyd, S. (2007). Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 4: TCP-Friendly Rate Control for Small Packets (TFRC-SP). <http://www.ietf.org/rfc/rfc5622.txt>. Último acesso: 6 de maio de 2011.
- Leavitt, N. (2010). Network-Usage Changes Push Internet Traffic to the Edge. *Computer*, 43(10):13 –15.

