

Global Media Transmission Protocol (GMTP)

Leandro M. de Sales¹, Wendell S. Soares¹, Thiago B. M. de Sales¹,
Hyggo O. de Almeida², Angelo Perkusich²

¹Instituto de Computação (IC)
Universidade Federal de Alagoas (UFAL) – Maceió, AL – Brasil

²Centro de Engenharia Elétrica e Informática (CEEI)
Universidade Federal de Campina Grande (UFCG) – Campina Grande, PB – Brasil

leandro@ic.ufal.br, hyggo@dsc.ufcg.edu.br, perkusich@dee.ufcg.edu.br

Abstract. *Live streaming systems, over the Internet, execute various algorithms that have been implemented to compensate the lack of network services for this purpose. As a result, it is difficult to define mechanisms for interoperability among such systems, increasing the consumption of network resources and impacting in the metrics that determine user satisfaction when watching a live event. This work introduces the Global Media Transmission Protocol (GMTP), a cross-layer network protocol for live media distribution through the use of P2P sockets (transport) in order to constitute a network of favors among cooperative routers, managed by the streaming servers, which orchestrate the partnerships based on the capacity of each transmission channel established by the routers to forward the media datagrams to the end systems.*

Resumo. *Os sistemas para transmissão de mídias ao vivo na Internet executam diversas funções que foram implementadas para suprir a ausência de serviços de rede para este fim. Esta prática dificulta a interoperabilidade desse tipo de sistema e, conseqüentemente, no consumo exacerbado dos recursos de rede, impactando nas métricas que determinam a satisfação do usuário ao assistir a um evento ao vivo. Neste trabalho, introduz-se o Global Media Transmission Protocol (GMTP), um protocolo de rede cross-layer para distribuição de mídias ao vivo pelo uso de sockets P2P (transporte) e formação de parcerias entre roteadores (rede), gerenciadas pelos servidores da mídia, que orquestram as parcerias com base na capacidade de cada canal de transmissão formado pelos roteadores para repassar os datagramas multimídia aos sistemas finais.*

1. Introdução

Com a evolução da *World Wide Web* para a Web 2.0, os usuários passaram a ter um papel de destaque no processo de prover alguns serviços. Por exemplo, o serviço de distribuição de conteúdos multimídia ao vivo, como o CoolStreaming™, PPLive™ e o UStream.tv™, permitem a transmissão de conteúdos ao vivo gerados a partir do computador de um usuário e transmitidos para milhares de outros usuários conectados à Internet [Silva et al., 2011]. Nesse contexto, há evidências contundentes de que os sistemas proeminentes definem uma arquitetura de rede híbrida P2P/CDN, ou seja, par-a-par e cliente-servidor, com suporte de uma rede de distribuição de conteúdos. Como resultado, consegue-se escalabilidade do número de usuários e redução de custos com infraestrutura de rede, por meio das redes P2P; ao passo que facilita-se o gerenciamento e obtém-se maior estabilidade de disponibilização dos serviços, por meio das CDNs (qualidade) [Meskovic et al., 2012].

A disseminação de diferentes sistemas e protocolos de rede com este propósito tem levado à pulverização de soluções para transporte de dados, gerando uma falta de

padronização na forma como tais sistemas transportam seus dados na Internet, principalmente por tais soluções serem implementadas na camada de aplicação [Liu et al., 2008]. Isto não seria um problema para a infraestrutura de rede e para os sistemas finais se essa pulverização não potencializasse duplicações de fluxos de dados da transmissão de um evento destinado a um sub-conjunto de redes contendo nós receptores interessados em tal evento. Como consequência, exponencia-se o consumo desnecessário de recursos de rede, com a impossibilidade de que dois ou mais nós, conectados em sistemas distintos, cooperarem entre si, compartilhando o mesmo fluxo de dados transmitido por um servidor. Idealmente, os nós deveriam compartilhar o mesmo fluxo de dados e cada um apresentar o conteúdo da forma definida pela aplicação, desacoplando a forma de transportar os dados da forma como estes são apresentados pela aplicação aos seus respectivos usuários, como no caso do serviço Web. No ponto de vista de protocolo de comunicação, o sucesso da Web ocorre devido ao desacoplamento do conteúdo a ser transmitido e a forma de transportá-lo entre o cliente (navegador) e o servidor (web). Independente do modelo e versão dos navegadores e dos servidores web, todos transportam dados utilizando o protocolo TCP, evitando-se assim a pulverização entre diferentes formas de transportar os dados de tal serviço – fica a cargo de cada aplicação decidir como exibir as informações recebidas. Contudo, observa-se a inexistência de um protocolo equivalente ao TCP adequado para distribuição de mídias ao vivo, apesar da grande demanda de utilização dos mais variados sistemas – a Internet não foi primariamente projetada para este fim.

Sendo assim, neste trabalho, apresenta-se resumidamente o projeto e a avaliação de um protocolo de rede denominado *Global Media Transmission Protocol* (GMTP), que considera uma arquitetura híbrida P2P/CDN para distribuição de mídias ao vivo com base na constituição de uma rede de favores entre roteadores. Mais especificamente, busca-se melhorar as métricas que determinam a satisfação dos usuários ao reproduzirem uma mídia ao vivo, através de um protocolo que explora a relação entre o escalonamento de recursos computacionais em redes P2P e a estabilidade das redes CDN, combinando-se técnicas para otimizar a formação de parcerias, que passa a ser entre os roteadores, quando seus respectivos clientes estão interessados em uma mídia ao vivo.

Este documento está organizado como segue. Na Seção 2, apresenta-se brevemente o funcionamento do GMTP. Na Seção 3, apresentam-se uma análise de projeto do GMTP em comparação ao estado da prática e da arte. Na Seção 4, apresentam-se a metodologia e os resultados obtidos sobre o GMTP em um cenário de rede simulado. Por fim, na Seção 5, apresentam-se as considerações finais, como conclusões e trabalhos futuros.

2. Global Media Transmission Protocol (GMTP)

O *Global Media Transmission Protocol* (GMTP) é um protocolo que atua nas camadas de transporte e de rede (*cross-layer*) da pilha TCP/IP, projetado para operar na Internet em sistemas de distribuição de mídias ao vivo. Trata-se de um protocolo baseado em uma arquitetura de rede híbrida P2P/CDN, através da qual transmitem-se e compartilham-se datagramas de um ou mais sistemas independente do controle da aplicação. Para isto, constituem-se redes de favores formadas por roteadores, que cooperam entre si a fim de entregarem o conteúdo multimídia de interesse comum aos seus clientes. Nesse cenário, os servidores atuam como super nós para os nós da rede P2P, auxiliando-os no envio e recebimento dos fluxos de dados. Uma aplicação cliente reproduz o conteúdo multimídia ao usuário final à medida que recebe pacotes de dados contendo partes da mídia, geradas por um ou mais servidores. Os clientes não necessariamente recebem os pacotes de dados diretamente dos servidores, mas podem recebê-los de roteadores já envolvidos na transmissão da mídia. Os servidores instruem os roteadores a efetivarem as parcerias com outros roteadores que possuem clientes também interessados no mesmo conteúdo multimídia, o que ocorre com base na capacidade de transmissão das rotas já utilizadas

para disseminar a referida mídia. As informações das capacidades das rotas são obtidas e atualizadas periodicamente, por meio de um algoritmo de controle de congestionamento assistido pela rede chamado *Rate Control Protocol* (RCP) [Dukkipati, 2008].

As trocas de dados entre os nós GMTP ocorrem por meio do envio e recebimento de partes de uma mídia (*chunks*), transmitidas por diferentes nós da rede. Os roteadores GMTP transmitem os fluxos de dados para outros roteadores em modo *unicast*, ao passo que distribuem os pacotes de dados a seus clientes em modo *multicast*, caracterizando um modo de transmissão *multi-unicast*. Em ambos os modos de transmissão, o GMTP realiza controle de congestionamento em transmissões sem garantia de entrega e a escolha do modo de transmissão para disseminar um fluxo de dados ocorre sem a influência da aplicação. Nesse contexto, a aplicação precisa simplesmente “sintonizar” sua conexão em um determinado canal *multicast* configurado automaticamente pelo roteador, de acordo com a demanda de seus clientes e informado como resposta ao pedido de conexão transmitido pelo cliente, porém interceptado por um roteador no trajeto ao servidor. Tal abstração para a camada de aplicação ocorre de modo que os processos em execução utilizam o GMTP através de uma API compatível com as especificações de *socket* BSD e POSIX, facilitando-se seu uso nos atuais e futuros sistemas, onde os clientes se tornam compatíveis para reproduzir um conteúdo independente do servidor que o transmite.

O princípio básico do GMTP é a introdução do conceito de *socket* P2P, onde a aplicação envia um pedido de conexão ao servidor, similar ao que ocorre com protocolos de transporte tradicionais. A diferença é que, se um roteador, presente no trajeto entre um cliente requisitante e o servidor, já estiver repassando uma determinada mídia sendo requisitada, este pode interceptar o referido pacote de requisição e responder ao cliente, como se fosse o próprio servidor.

2.1. Visão Geral

Define-se o GMTP pela execução de quatro grandes etapas (Figura 1): *Constituição da rede de favores*: descobrir, definir, efetivar e desfazer parcerias entre os roteadores de acordo com o evento ao vivo a ser transmitido; *Distribuição de fluxos de dados através de uma camada de socket*: conectar os clientes interessados em receber um fluxo de dados de um evento ao vivo, bem como transmitir tal fluxo de dados através das redes de favores; *Controle de congestionamento*: controlar a taxa de transmissão dos fluxos de dados distribuídos e utilizar as informações sobre a capacidade de transmissão de um canal para sugerir favores entre roteadores; *Autenticidade do conteúdo*: verificar a autenticidade do fluxo de dados antes de repassá-los aos clientes, evitando-se ataques de poluição.

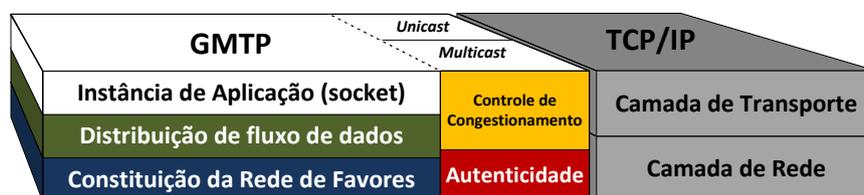
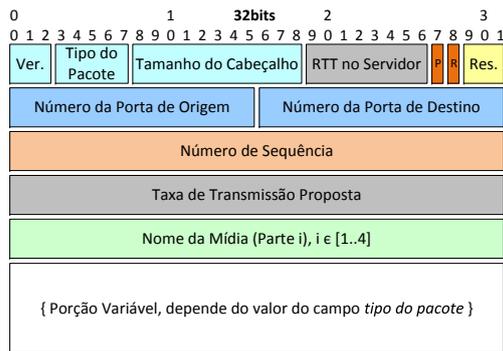


Figura 1. Blocos funcionais do GMTP e as relações com a pilha de protocolos TCP/IP.

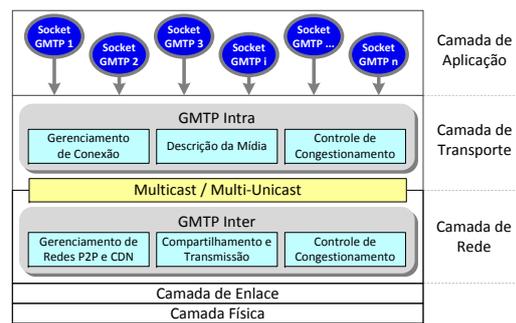
Toda comunicação entre dois ou mais nós GMTP ocorre através da troca de datagramas com cabeçalhos organizados por uma porção fixa e outra porção variável de campos de dados, os quais carregam sinalizações de controle necessárias para executar as ações do protocolo. Na Figura 2(a), ilustra-se a porção fixa do cabeçalho GMTP, com seus respectivos campos e tamanhos de cada campo. As partes variáveis do cabeçalho

GMTP é definida por um *codenome*, determinado na porção fixa, no campo *Tipo de Pacote*, compreendendo 18 tipos diferentes de pacotes que permitem o estabelecimento de conexão, compartilhamento de conteúdo, formação da rede de favores, descrição da mídia e verificação de autenticidade do conteúdo sendo transportado. Por limitações de espaço, não será possível descrever todos os campos do cabeçalho, tampouco suas partes variáveis, mas é muito importante entender que para a adoção do GMTP na Internet, será necessário registrar o uso de um novo número para o campo *Protocolo*, no caso do IPv4, e para o campo *Próximo Cabeçalho*, no caso do IPv6.

O protocolo GMTP é composto por dois módulos chamados de *GMTP-Intra* e *GMTP-Inter*, que operam nas camadas de transporte e de rede, respectivamente, definindo assim sua arquitetura, ilustrada na Figura 2(b). A principal responsabilidades do GMTP-Intra é fornecer serviços às aplicações de rede a fim de abstrair a complexidade na execução de tarefas comuns a qualquer sistema final, tais como conexão multi-ponto, multiplexação/demultiplexação de segmentos IP entre as camadas de transporte/rede/aplicação e controle de congestionamento. Este módulo compreende a instância do GMTP em execução no sistema operacional do cliente, acessível através de uma API de *socket* GMTP. Já a responsabilidade do GMTP-Inter é de constituir as redes de favores P2P compostas por roteadores, os quais funcionam como pontes de acesso aos servidores de uma rede CDN.



(a) Porção fixa do cabeçalho de pacotes GMTP.



(b) Arquitetura do Protocolo GMTP.

Figura 2. Arquitetura do protocolo GMTP e a porção fixa do seu cabeçalho de pacotes.

No contexto dos tipos de nós do GMTP, definem-se 4 tipos, definidos a seguir. *Cliente*, sistema final capaz de reproduzir e gerar conteúdos multimídia ao vivo, executa um processo em nível de sistema operacional, representando uma aplicação manipulada pelo usuário final. Um cliente recebe os pacotes de dados e os entrega ao processo de aplicação em execução. *Relator*, um cliente com habilidades de enviar relatórios periódicos ao nó *Repassador* sobre o estado da transmissão. *Repassador*, roteadores que participam efetivamente da rede de favores, com a responsabilidade de repassar os datagramas, originados em um ou mais *Servidores*, para outros repassadores até que os datagramas alcancem os clientes. *E*, *Servidor*, sistema final que participa de uma rede CDN e obtém a mídia a ser transmitida. Os servidores recebem sinalizações de controle contendo requisições dos repassadores que, ao receberem uma resposta correspondente à sua requisição, atendem à demanda de um ou mais clientes.

2.2. Fluxo básico de comunicação

Como ilustra-se no fluxograma da Figura 3, o GMTP executa 6 grandes passos (marcados em verde): *Registro de Participação*, *Transmissão de Dados*, formação da rede de Fa-

mite a mídia correspondente, já que o pedido de conexão é intencionalmente endereçado a este. Em seguida, o servidor decide se aceita o pedido de conexão ou se delega tal requisição a um repassador, com base no conhecimento de quais repassadores já estão encaminhando o conteúdo de interessado para alguma rede. No primeiro caso, o servidor simplesmente responde ao cliente e inicia a transmissão do fluxo de dados correspondente, o que resulta em conhecer a nova rota utilizada para transmitir o conteúdo, definida pelos roteadores existentes do servidor até ao cliente, nessa direção e ordem (Passo 2). No segundo caso, em vez de aceitar uma nova conexão, o servidor a recusa e determina que um repassador comece a servir o repassador (de origem) do cliente, determinando-se uma parceria entre um repassador já em uso para disseminar o conteúdo de interesse e o repassador que solicitou o registro de participação (Passo 3). Em ambos os casos, sempre o repassador de origem assumirá o controle de uma requisição do cliente, habilitando-se como candidato a parceiro para outros repassadores, quando motivados por requisições originadas pelos seus respectivos clientes. Nesse ínterim, outros passos são executados, como distribuição dos pacotes de dados nas redes locais e verificação da autenticidade do conteúdo no momento do repasse (Passo 4). Especificamente, a verificação da autenticidade se baseia na assinatura digital dos dados transportados nos pacotes e na capacidade que os repassadores têm de validar se o conteúdo foi intencionalmente alterado por usuários maliciosos no trajeto até o cliente, com base no certificado digital emitido pelo servidor – esta ação é opcional e decidida pela aplicação no início da transmissão.

Por fim, nos Passos 5 e 6, executam-se os algoritmos de controle de congestionamento. No Passo 5, executa-se um algoritmo de controle de congestionamento em cada repassador, que calcula sua capacidade de transmissão atual com base apenas no tamanho da fila de repasse e no atraso fim-a-fim marcado em pacote de dados a ser roteado. Em seguida, o repassador compara sua capacidade de transmissão com a menor capacidade de transmissão de todos os repassadores anteriores, transportada em cada pacote de dados. Se sua capacidade de transmissão for menor que a capacidade de transmissão informada no pacote de dados, o repassador a altera informando sua capacidade de transmissão, caso contrário, transmite-o para o próximo repassador. Ao final desse processo, o servidor regulará sua taxa de transmissão com base no repassador com menor capacidade de transmissão, informação contida nos pacotes de dados transmitidos ao servidor. Essa estratégia foi baseada no protocolo RCP, porém adaptada no GMTP para suportar o conceito de sub-fluxos. O importante é que além de regular sua taxa de transmissão, o servidor também sugere que os repassadores realizem parcerias entre si com base na capacidade de transmissão de cada repassador, propondo-se portanto um mecanismo explícito de seleção de nós com base na capacidade de transmissão das rotas utilizadas para distribuir o conteúdo. Já no Passo 6, executa-se um algoritmo de controle de congestionamento adaptado do *TCP-Friendly Multicast Congestion Control* (TFMCC) [Widmer and Handley, 2006] para fluxo de dados *multicast*, regulando-se a taxa de transmissão na rede local com base nos relatórios transmitidos pelos relatores.

Sendo assim, o posicionamento dos repassadores e suas habilidades permitem a redução do número de fluxos de dados na rede correspondentes a um mesmo evento, ao tempo que maximiza a quantidade de clientes interessados em receber o mesmo fluxo (escalabilidade). Isto porque qualquer roteador pode fazer cache temporário de datagramas referente a uma determinada mídia. Por este mesmo motivo, o protocolo GMTP é flexível para permitir que um repassador atue somente encaminhando conteúdos multimídia entre duas ou mais redes distintas, mesmo que este não tenha demandas explícitas dos seus clientes por tal conteúdo. Desta forma, maximiza-se o uso dos canais de transmissão ociosos, em particular das redes residenciais, principalmente quando seus usuários estão ausentes e portanto sem fazer uso dos recursos disponíveis de rede.

3. Análise do Projeto GMTTP

Os sistemas de distribuição de mídias ao vivo utilizam *middlewares* que implementam as principais funções necessárias para distribuição de mídias ao vivo, mas devido à existência de diferentes opções, cada sistema faz uso de um *middleware* específico (quando o fazem), causando um problema de interoperabilidade entre os sistemas. A falta de interoperabilidade tem importância crítica para o caso dos atuais sistemas de transmissão de mídias ao vivo: delega-se muitas responsabilidades às aplicações, que passam a ter que tratar problemas que não existiriam se a rede oferecesse um melhor suporte para essa classe de aplicação. O fato é que, apesar do suporte dos *middlewares*, que encapsulam e compartilham recursos computacionais complexos para desenvolver aplicações de rede, estes não encapsulam e tampouco compartilham a forma que os nós deveriam se comunicar. Em essência, argumenta-se que as minúcias de comunicar diferentes nós em uma rede é responsabilidade da infraestrutura de rede e não das aplicações, caso contrário, as aplicações se tornam soluções limitadas às estimativas sub-ótimas sobre a real disponibilidade dos recursos de rede.

Cenários como estes têm sido observado nos últimos anos, onde os pesquisadores têm apresentado diversas estratégias para otimizar a distribuição de mídias ao vivo na camada de aplicação, quando na verdade o problema é de infraestrutura. As aplicações e/ou *middlewares* implementam funções para tratar aspectos de conexão multi-ponto e diferentes topologias [Garropo et al., 2012]; seleção de nós [Liu et al., 2012]; tolerância às desconexões e outros problemas causados pelo *churn* [Sakata et al., 2012]; disponibilização de informação de contexto sobre rede para dar suporte à execução dos serviços das aplicações, como localização da rede, medições e custos entre redes para melhorar a qualidade de serviços dos sistemas finais [Shibuya et al., 2011]; estratégias para incentivos a cooperação entre nós [Wang et al., 2012]; algoritmos para inibir a participação de nós *free-riders* [Shin et al., 2009]; adaptação de fluxo baseado na capacidade de recepção dos nós [Thang et al., 2013]; e segurança [Vieira and Campos, 2010]. O fato é que o uso de alguns desses recursos se tornariam desnecessários se existisse um protocolo nas camadas de transporte e rede da pilha TCP/IP que tratasse *distribuição de mídias ao vivo* de forma peculiar, observando-se a relação entre o compartilhamento explícito dos canais de transmissão e a interoperabilidade entre os sistemas finais.

Sendo assim, o projeto do GMTTP objetiva não apenas um protocolo que reduz a complexidade das aplicação e promove a interoperabilidade entre os sistemas, mas também suscita que um conjunto mínimo de serviços, quando organizado de forma mais apropriada, promove melhorias substanciais no projeto das aplicações e principalmente no uso dos recursos de rede. Os desenvolvedores de sistemas de distribuição de mídias ao vivo enfrentam problemas que deveriam ser solucionados nas camadas inferiores da pilha TCP/IP e, por tentarem resolver na camada de aplicação, geram problemas adicionais. Por exemplo, ao tentarem implementar na aplicação recursos para conexão multi-ponto (em geral P2P) e tratamentos de diferentes topologias, passa a ser necessário selecionar nós parceiros e tolerar falhas, o que por si só já geram outros problemas e que impactam negativamente na qualidade de serviço das aplicações. A necessidade de selecionar nós parceiros acarreta na necessidade de definir estratégias para atenuar o *churn* da rede, o que requer que as aplicações definam estratégias utilizem o histórico de disponibilidade e oferta de serviço dos candidatos a parceiros, o que acarreta na necessidade de ter que aferir e tomar decisões para ordenar os nós com base em critérios pré-definidos, mas sem garantias de obter uma melhor qualidade de serviço.

Como consequência das ações anteriores, demanda-se a obtenção de informação de contexto sobre os nós (localização, perfil do usuário, estimativas e medições de custos entre nós, etc). Devido ao dinamismo das redes, gerado pela tentativa de implementar conexão multi-ponto, faz-se necessário incentivar à cooperação entre nós a fim de mantê-los

mais tempo conectados. Como consequência, deve-se explorar e diversas técnicas para este fim, como as baseadas em modelos econômicos, cada uma com seus próprios desafios. Pelo modelo cooperativista implementado nas aplicações P2P, tornou-se necessário identificar e inibir a participação de nós *free-riders*, o que remete à necessidade de incentivos. Devido à complexidade de estimar a real capacidade de transmissão dos nós, tornou-se também necessário o uso de técnicas para monitorar a capacidade de recepção dos nós clientes a fim de adaptar os fluxos de dados com base em tal informação, sendo também necessário ofertar canais de transmissão com diferentes níveis de qualidade da mídia. Por fim, devido à cooperação direta entre os clientes, facilita-se a implementação de ataques de poluição e, portanto, fez-se necessário o emprego de soluções para impedi-los, ou ao menos validar o conteúdo antes de sua reprodução ao usuário final.

Ao observar os problemas resultantes da tentativa suprir as limitações da rede, a seguir, enumeram-se cinco principais benefícios que podem ser constatados ao utilizar o GMTP, tanto no contexto dos sistemas de distribuição de mídias ao vivo quanto no contexto do consumo dos recursos de rede, relacionando-os com os recursos atualmente empregados nos sistemas de distribuição de mídias ao vivo (Figura 4).



Figura 4. Estado das funções de aplicação após o uso do protocolo GMTP.

1. *Otimização arquitetural:* antes do GMTP, os desenvolvedores de sistemas multimídia eram obrigados a considerar limitações de arquitetura de rede em seus projetos de software. Por exemplo, a lógica para distribuição da mídia era toda embutida na aplicação, aumentando sua complexidade de manutenção. O GMTP abstrai a complexidade de distribuir mídias ao vivo com base, por exemplo, no gerenciamento transparente de canais *multicast* (adicionar e remover) sem a influência da aplicação, inclusive entre diferentes domínios administrativos. Além disso, abstraem-se outros aspectos importantes para a aplicação, como o controle de congestionamento, antes embutidos na aplicação por falta de uma proposta mais adequada.
2. *Interoperabilidade entre sistemas:* O GMTP unifica a forma que os sistemas finais transmitem e recebem mídias ao vivo, uma vez que todo esse processo ocorre na camada de transporte e sem qualquer influência da aplicação. Isto significa que os sistemas podem cooperar entre si sem necessariamente deter conhecimentos arquiteturais e de interfaces de aplicação (API) uns dos outros. O plano é entregar as partes da mídia o mais rápido quanto possível, utilizando-se os melhores pares e independente dos sistemas. Para tornar isto possível, estabeleceram-se no GMTP funções como descrição da mídia e mecanismos que promovem a cooperação entre os nós, sem necessitar manter controle sobre os nós que mais contribuem e aqueles que apenas se aproveitam dos recursos dos outros (*free-riders*). Nesse ínterim, realizam-se outras funcionalidades importantes tanto para melhorar o serviço oferecido às aplicações quanto ao consumo dos recursos de rede. Por exemplo, o mecanismo de controle de congestionamento empregado no GMTP funciona de forma colaborativa, tanto em *unicast* quanto *multicast*,

sem a influência da aplicação, que não pode injetar informações falsas sobre sua percepção do estado da rede.

3. *Facilidade na integração*: os atuais e novos sistemas de distribuição de mídias ao vivo poderão integrar o GMTP de forma simples, uma vez que se manteve o uso da API tradicional de *sockets*. Para os sistemas existentes, muitas das funcionalidades, como as citadas no início desta seção não serão mais necessárias, podendo-se reduzir sobremaneira a complexidade da aplicação e consequentemente de manutenção. Para os sistemas novos, permite-se uma prototipação mais rápida e com menor chances de erros, evitando-se a super utilização dos recursos de rede devido ao emprego de boas práticas de engenharia de software para distribuição de mídias ao vivo e desacopladas da aplicação.
4. *Desacoplamento e Extensibilidade*: a organização do GMTP em dois módulos, GMTP-Inter e GMTP-Intra, permite o desacoplamento entre o que é função de rede e de aplicação, respectivamente. Isto evita que as funções de rede, como controle de congestionamento e distribuição de pacotes de dados sofram influência da aplicação. Apesar dessa separação, ainda sim permite-se a troca de informações entre os sistemas finais e a rede, mas que sejam somente pertinentes à distribuição de mídias (*Cross-Layer*). Além disso, possibilita-se a extensão do GMTP em casos de requisitos mais avançados da aplicação, quando não disponível no GMTP. Por exemplo, um desenvolvedor poderá promover alterações no GMTP-Intra e/ou GMTP-Inter a fim de atender suas necessidades sem afetar o GMTP-Inter e a aplicação em execução no sistema final. Uma fabricante de um roteador pode promover mudanças no GMTP-Inter (por exemplo, melhorar a estratégia de manipulação do *cache* ou de verificação de pacotes) e disponibilizar uma nova versão da *firmware* sem necessariamente ter que atualizar o GMTP-Intra em todos os sistemas finais. As decisões tomadas principalmente na definição do cabeçalho do GMTP oferecem esse tipo de flexibilidade. Nesse contexto, é possível, por exemplo alterar o formato de descrição de uma mídia (função do GMTP-Intra) sem precisar alterar o GMTP-Inter, permitindo-se atualizações gradativas dos sistemas finais para uma nova versão do GMTP-Intra, sem afetar o funcionamento da aplicação, devido à abstração da API de *sockets*. O fato é que a proposta do GMTP é uma infraestrutura que permitirá a evolução do protocolo sem afetar a camada de aplicação, permitindo-se a adição de outras estratégias de distribuição de mídias ao vivo que possam surgir futuramente e o principal, que estarão disponíveis automaticamente para uso em qualquer aplicação.
5. *Eliminação de recursos paliativos e consequentes*: quando se delega muitas responsabilidades à aplicação, há uma tendência de desordem, de fragmentação de funcionalidades e consequentemente de um baixo aproveitamento dos recursos de rede, devido à liberdade que as aplicações passam a ter de promover seus próprios recursos da forma que seu desenvolvedor desejar. Por exemplo, em redes puramente P2P é sempre complexo manter uma estrutura sem nós *free-riders*, apesar de existirem algoritmos para reduzir o efeito dessa prática na rede. Com o uso do GMTP pode-se reduzir drasticamente os nós *free-riders*, pois quando um nó requisita um fluxo de dados automaticamente o GMTP poderá compartilhá-lo com outros nós parceiros. Um outro exemplo é o tratamento para conexão multi-ponto e tolerância às falhas. O mecanismo de conexão multi-ponto é resolvido no GMTP pela interceptação de requisições e sugestão de parcerias entre nós roteadores. Com relação às falhas, mesmo que um nó roteador seja desconectado da rede seus parceiros não se tornarão órfãos, pois o nó repassador GMTP pode obter o conteúdo multimídia a partir de múltiplos nós repassadores, o que reduz drasticamente problemas causados pelo *churn*. Enfim, atualmente outros problemas são tratados na camada de aplicação e que, com o uso do GMTP, tornam-se

obsoletos são: a necessidade de obter informações de contexto sobre os nós, tais como localização, perfil de usuário e custos; uso de mecanismos de incentivos para permitir a cooperação; adaptação de fluxos de dados com base na capacidade dos nós receptores; controle de congestionamento, detecção de nós *free-riders* e de ataques de poluição.

4. Avaliação de Desempenho

Nesta seção, apresenta-se a avaliação dos confrontos do GMTP contra duas proeminentes proposta de distribuição de mídias chamadas de Denacast/CoolStreaming [Seyyed and Akbari, 2011] e o CCN-TV [Ciancaglini et al., 2013]. Para realizar a comparação entre os sistemas supracitados, definiu-se a modalidade experimental em um ambiente de simulação de rede, sendo a topologia da rede a versão 2010, adaptada da atual rede de pesquisa e educação GÉANT¹, composta por 27 roteadores e uma quantidade de nós clientes entre 500 e 80.000, dependendo do tratamento executado. Para a conectividade dos nós clientes, utilizaram-se 297 roteadores de acesso, formando redes de diferentes domínios administrativos com, no máximo, 12 roteadores. Um roteador de cada uma dessas redes foi conectado aos roteadores do *backbone* da rede, estabelecendo-se enlaces de *download* de 1 *Mbps*. Além disso, os nós servidores foram conectados aos roteadores LV, CY e IL.

Na Tabela 1, apresentam-se os parâmetros de simulação utilizadas no experimento, com base na topologia da rede apresentada anteriormente. Salienta-se que foram alocadas baixas larguras de banda de *upload* e heterogêneas para os nós clientes, comparadas à taxa média de bits da mídia transmitida nos ensaios, que foi de 512 *Kbps*.

Tabela 1. Variáveis independentes utilizadas no experimento.

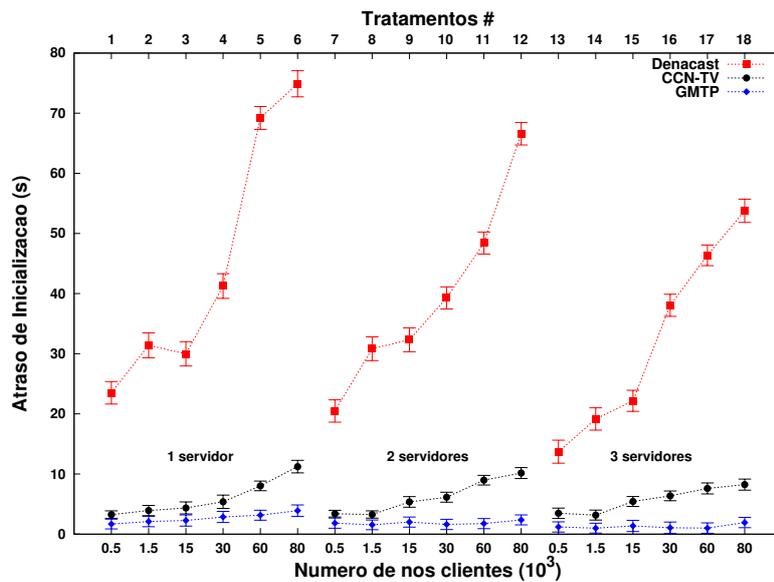
Variáveis	Valores
Dinâmica da rede (<i>churn</i>)	RandomChurn
Largura de banda entre LANs	2 <i>Mbps</i>
Atraso de propagação das redes locais	1 <i>s</i>
Atraso máximo aceitável para reprodução	5 <i>s</i>
Tempo de simulação de cada ensaio	900 <i>s</i>
Tamanho do <i>buffer</i> (roteadores)	100 pacotes (em média, 7s da mídia)
Tamanho máximo do pacote de dados	1500 <i>Bytes</i>
Taxa de upload dos roteadores das LANs	512 <i>Kbps</i> , 1 <i>Mbps</i> , 2 <i>Mbps</i> , 3 <i>Mbps</i>
Perda de pacote nos roteadores ¹	3 % de probabilidade de perder 5 % da fila

¹ Em caso de perda dos 5 % dos pacotes na fila do roteador, os pacotes são escolhidos aleatoriamente, função acionada com o *churn*.

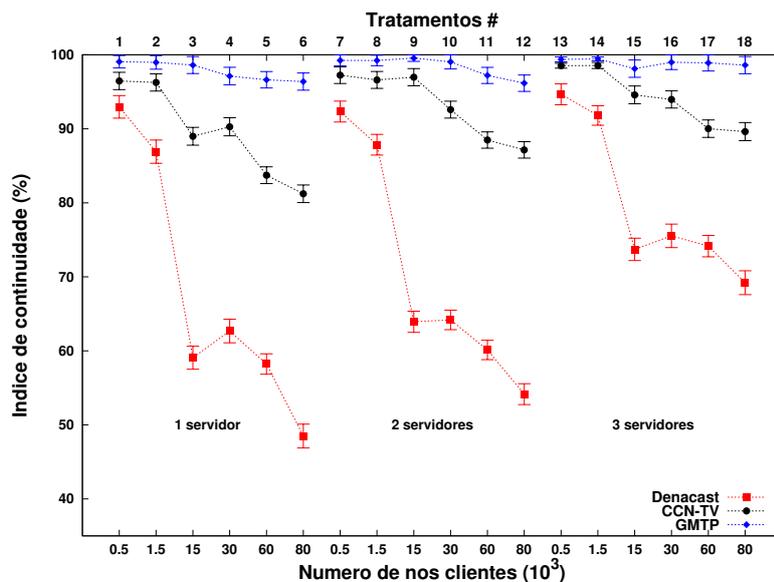
Nos gráficos das Figuras 5 e 6, apresentam-se os resultados de acordo com as métricas *Atraso de Inicialização*, *Índice de Continuidade*, *Distorção da Mídia* recebida por cada nó cliente e *Sobrecarga de Controle*. Em geral, observa-se superioridade de desempenho do GMTP frente aos outros sistemas estudados, apesar de empates técnicos entre o GMTP e o CCN-TV em alguns tratamentos, especialmente até 30.000 nós clientes e 1 e 3 nós servidores.

No caso do Denacast, observa-se que este obteve um desempenho inferior em comparação ao GMTP e ao CCN-TV. Nesse contexto, constatou-se que o Denacast se apresentou como linha base no experimento executado, sendo interessante para entender os limites de um exemplo do estado da prática em distribuição de mídias ao vivo, uma vez

¹ Site rede GÉANT: <http://www.geant.net/>



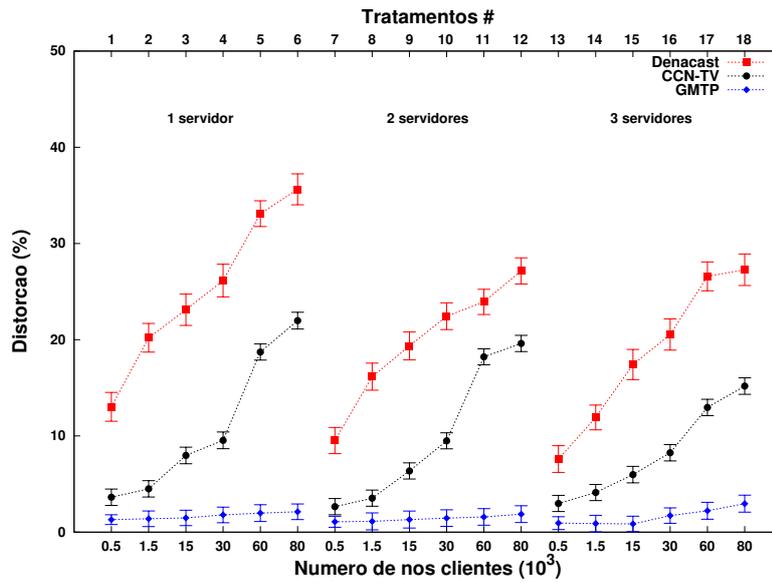
(a) Atraso de Inicialização x Número de clientes.



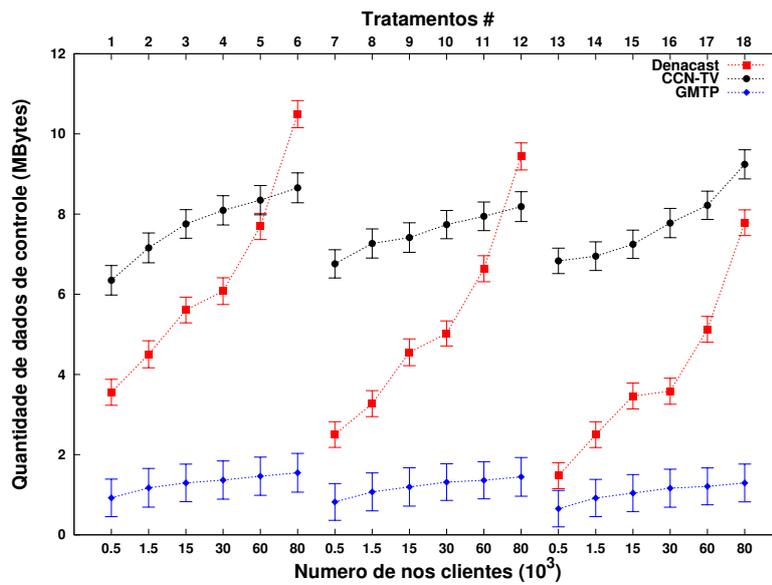
(b) Índice de Continuidade x Número de clientes.

Figura 5. Resultado dos experimentos GMTP vs. Denacast e GMTP vs. CCN-TV para as métricas de atraso de inicialização e índice de continuidade.

que se trata de uma proposta bastante difundida na Internet. De fato, parte dos resultados apresentados sobre o Denacast neste trabalho confirmam os resultados apresentados por seus autores em [Seyyedi and Akbari, 2011]. Nota-se que as disputas mais acirradas ocorreram entre o GMTP e o CCN-TV, evidenciando-se que a arquitetura e decisões de projeto, como o modelo de serviço utilizado, são aspectos que podem fazer diferenças significativas no desempenho dos sistemas para distribuir mídias ao vivo. Considerando-se os resultados para cada uma das métricas, com base em simulação de um cenário real de rede, reservando-se suas devidas proporções, pode-se afirmar que o GMTP apresenta um melhor desempenho se comparado às soluções do estado da prática e uma proposta do estado da arte.



(a) Qualidade da média x Número de clientes.



(b) Sobrecarga de controle x Número de clientes.

Figura 6. Resultado dos experimentos GMTP vs. Denacast e GMTP vs. CCN-TV para as métricas de qualidade da mídia e sobrecarga de controle.

Nesse contexto, resumam-se na Tabela 2, os ganhos do GMTP sobre o Denacast e o CCN-TV, com base nas médias obtidas para as métricas estudadas e com a quantidade de repetições necessárias para atingir 95 % de nível de confiança. Sendo assim, pode-se afirmar que, o uso do GMTP resulta na distribuição mais eficiente de mídias ao vivo em comparação a duas proeminentes propostas, uma considerada do estado da prática (Denacast) e a outra do estado da arte (CCN-TV). Especificamente, melhora-se o desempenho dos sistemas de distribuição de mídias ao vivo em 61,44 % em comparação às técnicas de aplicações baseadas em uma arquitetura P2P/CDN, ao passo que melhora-se 36,27 % em comparação às técnicas de distribuição de conteúdo com suporte da rede, instanciadas com base nos conceitos da rede CCN.

Tabela 2. Sumário dos ganhos do GMTP sobre o Denacast e o CCN-TV.

Métricas	Ganhos do GMTP por Confrontos	
	GMTP vs. Denacast	GMTP vs. CCN-TV
Atraso de Inicialização	93,22 %	47,17 %
Índice de Continuidade	37,88 %	9,01 %
Distorção da Mídia	25,26 %	13,03 %
Sobrecarga de Controle	177,05 %	165,94 %
Média aritmética	83,36 %	58,70 %
Média ponderada¹	61,44 %	36,27 %

¹ Para obter os referidos valores, calculou-se a média ponderada com pesos (2, 3, 3, 1) para as métricas atraso de inicialização, índice de continuidade, distorção e sobrecarga de controle, respectivamente. Esses pesos foram definidos com base na importância de cada métrica no processo de distribuição de mídias ao vivo.

5. Considerações Finais

Neste trabalho, apresentou-se um novo protocolo de distribuição de mídias ao vivo chamado *Global Media Transmission Protocol* (GMTP). Com base nos resultados de projeto e análise de desempenho do GMTP, a seguir, apresentam-se as principais conclusões deste trabalho.

O problema da interoperabilidade entre os sistemas de distribuição de mídias ao vivo pode ser resolvido através da abstração e da otimização das principais técnicas para distribuição de mídias ao vivo. Para isto, conclui-se que os novos métodos empregados no GMTP de conectividade multi-ponto e compartilhamento de conteúdo, tudo disponibilizado na camada de transporte e rede, sem sofrer a influência da camada de aplicação, permite o uso mais eficiente dos recursos de rede. Por isto, conclui-se também que o conceito de *socket P2P* empregado no GMTP permite uma abstração de distribuição P2P com suporte transparente ao uso de *multicast* por parte da aplicação, compartilhando-se os pacotes de dados entre múltiplos sistemas interessados por um mesmo conteúdo, o que reduz o consumo de recursos de rede e melhora a qualidade de serviço oferecida às aplicações. O desenvolvimento de sistemas se torna mais eficiente (mais rápido e com menores chances de erros) devido ao emprego de funções de software relacionadas à distribuição de mídias ao vivo desacopladas da aplicação. Esta decisão evita o uso de diferentes bibliotecas de software que, apesar de facilitar o desenvolvimento, não resolve o problema da interoperabilidade discutido anteriormente, sendo incapazes de usar toda a capacidade de transmissão que a rede pode oferecer. Com o GMTP, utilizam-se os recursos de rede de forma mais otimizada, explorando-os de forma compartilhada.

As soluções de aplicação são incomparáveis às soluções baseadas no núcleo da rede para distribuição de mídias ao vivo, por isso observaram-se confrontos mais acentuados entre o GMTP e o CCN-TV. No contexto de aplicação, constatou-se o pior desempenho do Denacast com relação às métricas avaliadas, apresentando-se com problemas de escalabilidade a partir de uma certa quantidade de nós clientes, sendo influenciado pela dinâmica da rede. Já no contexto do CCN-TV, conclui-se que o GMTP obteve um desempenho superior devido aos seguintes motivos:

1. O modelo de serviço *push/pull* empregado no GMTP é mais apropriado para distribuir conteúdos ao vivo, ou seja, para sistemas que necessitem, como principal requisito da rede, a entrega do conteúdo em um curto espaço de tempo, que não tenham restrições comerciais de compartilhamento de conteúdo e que não necessitem de entrega confiável. A principal consequência desse modelo de serviço é que os sistemas finais interessados em obter um determinado conteúdo não precisam

transmitir, periodicamente, requisições para continuar recebendo um determinado conteúdo, como ocorre na rede CCN.

2. Apesar dos pesquisadores que propuseram o CCN alegarem que não é preciso um sistema final conhecer a localização de um determinado conteúdo de interesse (endereço IP do servidor, por exemplo), neste trabalho, conclui-se o contrário. Ou seja, convém manter a identificação do sistema final de destino, visto que se pode otimizar métricas importantes, como o atraso de inicialização.
3. Devido ao modelo de serviço empregado na rede CCN, os nós clientes não conseguem saber, no caso de solicitar um determinado dado e não recebê-lo, se ainda é possível recuperá-lo ou não. Este dilema impacta diretamente na tomada de decisão se ainda faz sentido retransmitir um pacote de interesse correspondente a um pacote de dados a ser reproduzido ou se não faz mais sentido aguardar sua chegada. Além disso, os pacotes de interesse são dependentes da capacidade de *upload* do cliente, o que pode reduzir a quantidade de pacotes a serem recebidos pelo nó cliente, mesmo havendo largura de banda de *download* suficiente para comportar a transmissão dos pacotes de dados.

Além disso e apesar das constatações em trabalhos anteriores sobre o uso de algoritmos de controle de congestionamento assistidos pela rede, os quais melhoram o desempenho da rede nas transmissões de dados, neste trabalho, reforçou-se não somente este fato, mas também conclui-se que a aplicabilidade desse tipo de algoritmo é eficaz na busca de melhorar o desempenho no processo de distribuir mídias ao vivo através da Internet, utilizando-se informações mais acuradas para a formação de parcerias. Especificamente, conclui-se que:

1. utilizar informações explícitas sobre o congestionamento da rede para permitir a um servidor determinar parcerias, melhoram-se métricas importantes, como o índice de continuidade e a qualidade do conteúdo entregue aos sistemas finais;
2. utilizar o RCP e adaptá-lo para o cenário de distribuição de mídias ao vivo foi uma escolha adequada, pois se trata de uma proposta invariante à distribuição do tamanho do fluxo. Ou seja, independente se um fluxo de dados for de curta ou de longa duração, o RCP compartilha o canal de forma mais eficiente, pois emula-se o PS independente de quais e quantos são os fluxos, estabilizando-se a taxa de transmissão dos sistemas finais mais rapidamente, à medida que os fluxos são iniciados ou finalizados pelas aplicações;
3. o RCP, aliado ao conceito de sub-fluxo proposto neste trabalho, é simples e praticável, pois a taxa de transmissão ofertada é definida sem o conhecimento do número exato de fluxos, baseando-se somente no tamanho da fila, no tráfego agregado de entrada e no RTT do tráfego passando através do roteador;
4. a função de sub-fluxo empregado no GMTP-UCC permite que roteadores com maior capacidade de recepção continuem recebendo um fluxo multimídia em uma taxa possível, mesmo que outros roteadores no caminho até um determinado cliente possua uma capacidade inferior de recepção em um determinado instante. Essa função otimiza métricas como qualidade da mídia a ser reproduzida aos usuários finais.

Como trabalhos futuros, pretende-se conduzir estudos para explorar outras topologias de rede, incluindo redes com diferentes padrões de mobilidade, como redes veiculares, bem como entender o impacto do tráfego de dados gerados por outros protocolos sobre o GMTP. Investigar técnicas de adaptação de fluxo de dados multimídia ao vivo aplicadas ao GMTP, com base em estudos de técnicas para codificação do conteúdo multimídia de acordo com informações explícitas sobre o congestionamento da rede. Especificamente, pode-se integrar ao GMTP mecanismos avançados de adaptação de fluxo, como o apresentado em [Fernandes et al., 2011], que faz uso de informações explícitas

sobre o congestionamento da rede para permitir que os servidores adaptem o conteúdo de forma preditiva e com segurança para evitar super-utilização momentânea do canal. Implementar o protocolo GMTP e uma aplicação de referência em um sistema operacional a fim de executá-lo tanto em sistemas finais quanto em roteador. Nesse caso, sugere-se também avaliar o grau de complexidade de adaptar um sistema existente para utilizar o GMTP.

Referências

- V. Ciancaglini, G. Piro, R. Loti, L.A. Grieco, and L. Liquori. CCN-TV: A Data-centric Approach to Real-Time Video Services. In *WAINA Workshop, 2013*, 3 2013.
- Nandita Dukkkipati. *Rate Control Protocol (RCP): Congestion Control to Make Flows Complete Quickly*. PhD thesis, Stanford University, Stanford, CA, USA, 2008.
- Stenio Fernandes, Judith Kelner, and Djamel Sadok. An Adaptive-Predictive Architecture for Video Streaming Servers. *Journal of Network and Computer Applications*, 34(5): 1683–1694, 9 2011.
- Rosario G. Garroppo, Stefano Giordano, Stella Spagna, Saverio Niccolini, and Jan Seedorf. Topology Control Strategies on P2P Live Video Streaming Service with Peer Churning. *Computer Communication*, 35(6):759–770, 3 2012.
- Nianwang Liu, Zheng Wen, K.L. Yeung, and Zhibin Lei. Request-peer Selection for Load-balancing in P2P Live Streaming Systems. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, 4 2012.
- Yaning Liu, Hongbo Wang, Yu Lin, Shiduan Cheng, and G. Simon. Friendly P2P: Application-Level Congestion Control for Peer-to-Peer Applications. In *IEEE Global Telecommunications Conference, 2008*, 12 2008.
- Melika Meskovic, Himzo Bajric, and Mladen Kos. Content Delivery Architectures for Live Video Streaming: Hybrid CDN-P2P as the Best Option. In *The Fifth International Conference on Communication Theory, Reliability, and Quality of Service*, 2012.
- Y. Sakata, K. Takayama, R. Endo, and H. Shigeno. A Chunk Scheduling Based on Chunk Diffusion Ratio on P2P Live Streaming. In *NBiS*, 2012.
- S. M Y Seyyedi and B. Akbari. Hybrid CDN-P2P Architectures for Live Video streaming: Comparative Study of Connected and Unconnected Meshes. In *CNDS 2011*, 2 2011.
- M. Shibuya, Y. Hei, and T. Ogishi. ISP-Friendly Peer Selection Mechanism with ALTO-like Server. In *APNOMS*, 2011.
- Kyuyong Shin, D.S. Reeves, and Injong Rhee. Treat-before-trick: Free-riding Prevention for BitTorrent-like Peer-to-Peer Networks. In *IPDPS*, 2009.
- Thiago Silva, Jussara M. Almeida, and Dorgival Guedes. Live Streaming of User Generated Videos: Workload Characterization and Content Delivery Architectures. *Computer Network*, 55(18):4055–4068, 12 2011.
- Truong Cong Thang, Hung T. Le, Hoc X. Nguyen, Anh T. Pham, Jung Won Kang, and Yong Man Ro. Adaptive Video Streaming over HTTP with Dynamic Resource Estimation. *Journal of Communications and Networks*, 15(6):635–644, 12 2013.
- Alex B. Vieira and Sergio V. A. Campos. *Transmissão de Mídia contínua ao Vivo em P2P: Modelagem, Caracterização e Implementação de Mecanismo de Resiliência a Ataques*. PhD thesis, Universidade Federal de Minas Gerais - UFMG, 3 2010.
- Tzu-Ming Wang, Wei-Tsong Lee, Tin-Yu Wu, Hsin-Wen Wei, and Yu-San Lin. New P2P Sharing Incentive Mechanism Based on Social Network and Game Theory. In *WAINA Workshop*, 2012.
- J. Widmer and M. Handley. TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification, 8 2006. IETF RFC 4654.