

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
ENGENHARIA DE COMPUTAÇÃO

BRUNA RIZZARDO FIORENTIN

**Um Mecanismo de Relacionamento entre
Objetos para Economia de Recursos na
Arquitetura de Rede CCN: projeto,
implementação e avaliação**

Trabalho de Graduação apresentado como
requisito parcial para a obtenção do grau de
Engenheira de Computação

Prof. Dr. Marinho Pilla Barcellos
Orientador

Porto Alegre, Novembro de 2013

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Fiorentin, Bruna Rizzardo

Um Mecanismo de Relacionamento entre Objetos para Economia de Recursos na Arquitetura de Rede CCN: projeto, implementação e avaliação / Bruna Rizzardo Fiorentin. – Porto Alegre: UFRGS, 2013.

54 f.: il.

Trabalho de Conclusão – Universidade Federal do Rio Grande do Sul. Engenharia de Computação, Porto Alegre, BR-RS, 2013. Orientador: Marinho Pilla Barcellos.

1. ICN. 2. CCN. 3. Relacionamento de objetos. I. Barcellos, Marinho Pilla.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitora de Graduação: Prof. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do curso: Prof. Marcelo Goetz

Bibliotecário-Chefe do Instituto de Informática: Alexander Borges Ribeiro

*"We are what we repeatedly do.
Excellence, therefore, is not an act, but a habit."*

— ARISTOTLE

AGRADECIMENTOS

Este trabalho de graduação representa, para mim, não só a conclusão de mais uma etapa de minha vida, mas também a realização de um sonho. Desde o dia que assisti à formatura da minha irmã pela UFRGS, aos 13 anos, decidi que em meu futuro, um dia, eu também iria estar no palco do Salão de Atos. Estou concluindo com este trabalho, 10 anos mais tarde, a trajetória que me possibilitará transformar esse sonho em realidade; no entanto, se no dia 08 de março de 2014 eu estarei lá, foi porque eu pude contar com diferentes pessoas que me ajudaram e me impulsionaram rumo a esse objetivo, às quais eu gostaria de agradecer nesse espaço.

Primeiramente, gostaria de agradecer aos membros de minha família. Agradeço aos meus pais, Jaime e Nilse, que me mostraram desde pequena a importância dos estudos e que sempre me deram condições para alcançar meus objetivos. À minha irmã, Andressa, pela excelente demonstração de que, com dedicação e esforço, nada é impossível. Ao meu irmão, Diogo, pelo suporte intelectual. Ao meu sobrinho, Miguel, pelo sorriso sincero que alegrou meus dias durante o ano de realização desse trabalho. A todos, pelo amor e atenção que me foram dedicados até esse momento de minha vida. Afirmo, sem hesitação alguma, que minha família é de fundamental importância para todas minhas conquistas e para a minha felicidade.

Gostaria de agradecer à UFRGS, à École Centrale Paris e aos professores de ambas instituições por terem me proporcionado uma formação de qualidade. Agradeço ao meu orientador, professor Marinho, por ter me guiado na realização desse trabalho e por ter se mostrado sempre disponível quando precisei. Agradeço ao colega de grupo Rodolfo que me co-orientou nesse trabalho; além de todo o conhecimento agregado por ele, a presença do “tenso” no trabalho, por mais irônico que pareça, me tranquilizou. Agradeço ainda ao colega Lucas, um verdadeiro amigo que o grupo de redes me deu, tanto pelo auxílio no SecFuNet quanto pela força em minha procura por um futuro, estando sempre disposto a conversar e ajudar.

Por fim, agradeço aos meus amigos – *maridas*, amigos de Bento, amigos da Centrale Paris e da dupla-diplomação, colegas de UFRGS, pessoal da FOp, etc. – que de alguma maneira marcaram essa fase da minha vida. Nessa trajetória, pude contar com amigos que, além de terem me ajudado em momentos de dificuldade, compartilharam comigo momentos de lazer e de fuga em meio a tanto estudo, influenciando positivamente no meu bem-estar pessoal e profissional. Obrigada a todos!

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	8
RESUMO	9
ABSTRACT	10
1 INTRODUÇÃO	11
2 FUNDAMENTOS TEÓRICOS	13
2.1 Redes Orientadas a Conteúdo	13
2.2 Content-Centric Networking	14
2.3 Trabalhos relacionados	16
3 PROJETO DO RELACIONAMENTO DE OBJETOS	18
3.1 Conceito	18
3.2 API	19
3.3 Mecanismo proposto	20
3.4 Implementação	21
3.5 Aplicações	23
3.5.1 Distribuição de vídeos	23
3.5.2 Edição colaborativa de fotos	24
3.5.3 Publicidade em vídeos	25
3.5.4 Serviço online de mapas	26
3.5.5 Distribuição de conteúdo multimídia	26
4 METODOLOGIA	28
4.1 Ambiente padrão	28
4.2 Cenários	30
4.3 Métricas	30
5 RESULTADOS	32
5.1 Desempenho da rede	32
5.2 Qualidade de Experiência	35
5.3 Discussão dos resultados	37
6 CONCLUSÃO	39
REFERÊNCIAS	40

ANEXO A: TRABALHO DE GRADUAÇÃO I	43
---	-----------

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CCN	<i>Content-Centric Networking</i>
CDN	<i>Content Delivery Network</i>
CS	<i>Content Store</i>
DONA	<i>Data-Oriented Network Architecture</i>
FIB	<i>Forwarding Information Base</i>
ICN	<i>Information-Centric Networking</i>
IO	<i>Information Object</i>
NDO	<i>Named Data Object</i>
NetInf	<i>Network of Information</i>
PIT	<i>Pending Interest Table</i>
PoP	<i>Point of Presence</i>
PSIRP	<i>Publish-Subscribe Internet Routing Paradigm</i>
QoE	<i>Quality of Experience</i>
RNP	Rede Nacional de Ensino e Pesquisa
URI	<i>Uniforme Resource Identifier</i>

LISTA DE FIGURAS

3.1	Exemplo de relacionamento simples	18
3.2	Exemplo de relacionamento múltiplo	19
3.3	Exemplo de relacionamento linear	19
3.4	Distribuição de relações através de metadados	21
3.5	Comunicação para publicação de objetos em um repositório. Adaptado de (PARC, 2013a)	22
3.6	Modelagem de uma imagem utilizando relacionamento de objetos . .	24
3.7	Modelagem de uma região de mapa utilizando relacionamento de objetos	26
4.1	Mapa da topologia RNP	29
4.2	Árvore de distribuição da topologia experimental	29
5.1	Impacto de variar o número de clientes no tráfego total da rede	33
5.2	Impacto de variar o número de clientes na taxa de acerto da cache . .	33
5.3	Impacto de variar o número de conteúdos no tráfego total da rede . .	34
5.4	Impacto de variar o número de conteúdos na taxa de acerto da cache .	34
5.5	Impacto de variar o número de clientes no número de travamentos . .	35
5.6	Impacto de variar o número de clientes no tempo de inicialização . .	36
5.7	Impacto de variar o número de conteúdos no número de travamentos .	36
5.8	Impacto de variar o número de conteúdos no tempo de inicialização .	37

RESUMO

Aplicações de distribuição de conteúdo são atualmente responsáveis pela maior parte do tráfego de dados na Internet. Os recentes esforços para otimizar o desempenho dessas aplicações levaram à proposta de um novo paradigma para a Internet, denominado Redes Orientadas a Conteúdo. Em tal paradigma, conteúdos são modelados como objetos unicamente nomeados, de maneira a possibilitar uma melhor distribuição dos mesmos na rede. Diferentes arquiteturas foram propostas, destacando-se a arquitetura *Content-Centric Networking* (CCN).

Pode-se aumentar a eficiência da arquitetura CCN se conteúdos forem modelados como um conjunto de objetos relacionados. Tal abordagem possibilita o reuso de um mesmo objeto por diferentes conteúdos, podendo reduzir o volume de dados distribuídos na rede e aumentar a taxa de acerto da cache. Neste trabalho, propõe-se um mecanismo que permite o estabelecimento de relações entre objetos em CCN, modelando-o e apresentando exemplos de sua aplicação. Adicionalmente, implementa-se uma extensão para o protótipo CCN e realiza-se uma avaliação experimental, visando quantificar as melhorias de desempenho obtidas quando aplicações utilizam relacionamento entre objetos.

Os resultados obtidos mostram que o mecanismo proposto pode melhorar consideravelmente tanto o desempenho da rede quanto a qualidade de experiência percebida pelos usuários. Nos cenários de *streaming* de vídeo avaliados, o tempo de inicialização do vídeo é reduzido em 33,6% enquanto o tráfego da rede atinge uma redução de 34,4%.

Palavras-chave: ICN, CCN, relacionamento de objetos.

**An Object Relations Mechanism to Save Resources in the CCN Network
Architecture: design, implementation and evaluation**

ABSTRACT

Content distribution applications are nowadays responsible for most of Internet data traffic. Recent efforts aiming at optimizing their efficiency led to the proposal of a new Internet paradigm, known as Information-Centric Networking. In this paradigm, contents are modeled as uniquely named objects, allowing their dissemination to be more efficient. Different architectures were proposed, from which Content-Centric Networking (CCN) stands out as the most stable and popular one.

CCN efficiency may be increased if contents are modeled as a set of related objects. This allows the reuse of a single object by different contents, reducing the volume of data transferred in the network and increasing the cache hit rate. In this work, we propose a mechanism that allows the establishment of relations among objects in CCN, modeling it and presenting examples of its application. Moreover, we implement an extension to the CCN prototype and evaluate it experimentally, aiming at quantifying the benefits of employing object relations for content distribution.

Results show that the proposed mechanism can improve network efficiency as well as users' quality of experience. Considering the evaluated scenarios, the streaming start-up delay was reduced in up to 33.6% while the network traffic was reduced in 34.4%.

Keywords: ICN, CCN, object relations.

1 INTRODUÇÃO

A Internet foi criada com o intuito de possibilitar a comunicação fim-a-fim entre dois computadores – também denominados *hosts*. Dada a natureza estática dos *hosts* na época da concepção da Internet, a arquitetura tornou-se fortemente dependente da localização física dos dados. O protocolo IP, principal elemento da camada de rede da Internet atual, utiliza o endereço de rede de ambas as máquinas envolvidas em uma comunicação para rotear os dados entre elas (KUROSE; ROSS, 2012). No entanto, o uso da Internet vem se concentrando cada vez mais na distribuição de conteúdo. Segundo apontado por (AHLGREN et al., 2012), o tráfego IP global se aproximará de 80 exabytes/mês em 2015, sendo quadruplicado com base no tráfego de 2010; desse tráfego, aproximadamente 90% será devido à distribuição de vídeos.

A escalabilidade da distribuição de conteúdo na Internet atual, entretanto, está limitada pelos protocolos existentes, os quais exigem uma comunicação fim-a-fim entre o cliente que deseja um determinado conteúdo e o servidor que o possui. Dessa maneira, diferentes funcionalidades foram sendo incorporadas à Internet a fim de acompanhar a evolução em seu uso, como por exemplo as *Content Delivery Networks* (CDN), que replicam conteúdos multimídia em diferentes servidores para atender a mais requisições (VAKALI; PALLIS, 2003). Tal abordagem, entretanto, resultou em um crescimento desorganizado da Internet e em um conseqüente aumento da complexidade de manutenção, levando diversos grupos de pesquisa a considerar uma recriação da Internet, ou seja, a criação da Internet do Futuro.

Nesse contexto, surgiu o paradigma das Redes Orientadas a Conteúdo (ICN, do inglês *Information-Centric Networking*) (AHLGREN et al., 2012), cujo objetivo é fornecer uma infraestrutura de rede mais adequada ao atual uso da Internet, otimizando principalmente a distribuição de conteúdo e facilitando a mobilidade. Dentre as arquiteturas ICN existentes, destaca-se a *Content-Centric Networking* (CCN) (JACOBSON et al., 2012) (ZHANG et al., 2010), implementada no projeto CCNx (PARC, 2013b).

Em tal arquitetura, os principais componentes são os objetos de dados nomeados, os quais correspondem aos conteúdos a serem distribuídos. Através da requisição de objetos por nome e da presença de caches em todos os nodos da rede, CCN favorece a escalabilidade e a eficiência da Internet. Quando um objeto é requisitado, CCN o divide em pequenas partes (*chunks*) a fim de tornar sua transmissão mais eficiente. No entanto, o conceito de *chunks* é rígido, não existindo um mecanismo que considere que conteúdos podem ser compostos por um conjunto de objetos e relações entre eles.

Considerando-se uma divisão lógica de objetos poder-se-ia, por exemplo, modelar um conteúdo multimídia como um conjunto de três objetos: um vídeo, um áudio e uma legenda, os quais podem ser selecionados entre múltiplas opções de objetos variando a qualidade, o idioma ou outras propriedades. Nesse caso, essas múltiplas opções seriam

objetos e relações entre eles seriam mantidas na CCN. Tal divisão lógica pode influenciar na popularidade dos objetos, gerando impactos na distribuição dos mesmos em caches e, conseqüentemente, no uso de recursos da rede. Ademais, poder-se-ia utilizar a informação de como os objetos são relacionados para otimizar sua localização nas diferentes caches da rede.

Apesar do potencial que o relacionamento de objetos possui para otimizar a rede e a Qualidade de Experiência (QoE – *Quality of Experience*) dos usuários, CCN ainda não conta com um mecanismo para gerenciá-lo. Neste trabalho, propõe-se um modelo para estabelecer relações entre objetos em ICN. Tal modelo é opaco para a rede, com as vantagens de manter o núcleo da rede ICN simples e de não restringir aplicações desnecessariamente. Utilizando CCNx, implementa-se o mecanismo proposto visando a realização de experimentos e avalia-se os resultados obtidos, quantificando os benefícios para a rede e para os clientes.

O restante desse trabalho está organizado como segue. O Capítulo 2 apresenta o paradigma ICN e detalha a arquitetura CCN, através do projeto CCNx, além de analisar os trabalhos relacionados. O Capítulo 3 apresenta uma modelagem para o relacionamento de objetos e descreve o mecanismo proposto neste trabalho para seu gerenciamento, além de exemplificar como seria o seu uso em aplicações. No Capítulo 4, a metodologia de avaliação é apresentada, descrevendo-se o ambiente padrão, cenários e métricas utilizados. Os resultados obtidos na avaliação do mecanismo são descritos no Capítulo 5. Por fim, o Capítulo 6 encerra o documento, apresentando as principais conclusões e planos de trabalhos futuros.

2 FUNDAMENTOS TEÓRICOS

Atualmente, a comunidade de ICN propõe 4 arquiteturas principais: *Data-Oriented Network Architecture* (DONA) (KOPONEN et al., 2007), *Content-Centric Networking, Publish-Subscribe Internet Routing Paradigm* (PSIRP) (TROSSEN; PARISIS, 2012) e *Network of Information* (NetInf) (KAUFFMANN; PELTIER; TRUONG, 2013). A primeira seção desse capítulo, Seção 2.1, introduz de maneira genérica os componentes comuns a tais arquiteturas e discute as vantagens desse novo paradigma para a Internet. A Seção 2.2 concentra-se na arquitetura CCN, que é a escolhida para o estudo da mutabilidade de objetos nesse trabalho. Por fim, se encerra-se o capítulo com a apresentação dos trabalhos relacionados, na Seção 2.3.

2.1 Redes Orientadas a Conteúdo

Dentre as decisões de projeto de arquiteturas ICN, quatro aspectos destacam-se na literatura: nomeação, roteamento, *caching* e segurança.

Nomeação. Objetos Nomeados de Dados (NDO, do inglês *Named Data Objects*) são os principais componentes de uma Rede Orientada a Conteúdo. NDOs fornecem a abstração necessária para que um conteúdo seja distribuído na Internet sem necessidade de conhecimento da localização física na qual ele se encontra. Exemplos de NDOs são páginas web, imagens, vídeos, músicas, entre outros. Cada NDO deve ser inteiramente identificado pelo seu nome, ou seja, se dois servidores distintos armazenarem o mesmo NDO, o conteúdo recebido por um cliente que o requisitar deve ser o mesmo independente do servidor que o fornecer. A nomeação de objetos em ICN varia dependendo da arquitetura utilizada. Há dois esquemas principais de nomeação: plana utilizando *namespaces* e hierárquica. No contexto deste documento, o segundo esquema é mais importante, uma vez que é o utilizado pela arquitetura CCN. A nomeação hierárquica será, portanto, detalhada na Seção 2.2.

Roteamento. O roteamento em ICN pode ser genericamente resumido em duas fases. Primeiramente, a requisição de um conteúdo é encaminhada através de uma estratégia de roteamento baseada no esquema de nome da arquitetura. Após, ao alcançar um nodo que possua tal conteúdo, o mesmo é encaminhado de volta ao requisitante.

Com um roteamento baseado em nomes, ganha-se no quesito mobilidade em comparação com a Internet atual. O cenário de um cliente movendo-se por diferentes redes exemplifica esse ganho. Diferentemente do que ocorre atualmente, não é necessário que o cliente mantenha uma conexão fim-a-fim com o servidor original para obter um conteúdo, recalculando as melhores rotas; basta que ele continue requisitando os objetos desejados pelo nome, que é único no contexto em que está inserido. Um outro servidor - provavelmente mais próximo do cliente - pode responder à requisição dos dados.

Caching. Um dos princípios de ICN é que todos os nodos da rede potencialmente possuem uma cache. Consequentemente, um NDO requisitado pode vir a ser encontrado em diversos lugares da rede. Dessa maneira, a localização de um objeto não está restrita à localização do nodo que o publicou ou de nodos destinados a replicar este objeto. As caches de cada nodo são populadas conforme NDOs passam pelo mesmo em seu caminho ao destino. Duas vantagens da existência de um mecanismo de *caching* são: (i) o tráfego de dados distribui-se melhor pela rede; e (ii) proporciona-se uma melhor tolerância à interrupção através da existência de caminhos alternativos no caso de um nodo ou uma interface da rede falhar.

Segurança. A segurança na Internet normalmente é avaliada em termos de 4 componentes: integridade, confiabilidade, proveniência e disponibilidade. Espera-se que a rede garanta que os dados não sejam alterados indevidamente (integridade), nem lidos por outros (confiabilidade) ou corrompidos (disponibilidade). Além disso, deve certificar-se de que a origem dos dados é apropriada (proveniência) (GHODSI et al., 2011a).

Em ICN, a fim de garantir esses aspectos de segurança, é necessário certificar os NDOs diretamente, visto que os mesmos podem ser encontrados em diversas caches da rede. Proveniência, integridade e disponibilidade podem ser garantidas através da assinatura dos NDOs pelo publicador do objeto. Dessa maneira, os clientes podem certificar-se da validade do conteúdo simplesmente verificando a assinatura (GHODSI et al., 2011b).

Uma possível abordagem para assinar o conteúdo consiste na utilização de nomes auto-certificados, os quais são construídos através de estruturas criptográficas que permitem uma forte correspondência entre os conteúdos e seus nomes (RIBEIRO et al., 2012). Esses nomes podem ser verificados por *hash* ou por chave. No primeiro caso, o nome é gerado pela *hash* do conteúdo; o cliente, quando recebe um NDO, confere a integridade do conteúdo realizando a *hash* do mesmo e comparando com a *hash* informada pelo publicador. No segundo caso, os nomes são verificados pela *hash* da chave pública do publicador. Outra abordagem, que é a abordagem utilizada por CCN, é a utilização de uma assinatura do publicador, a qual é efetuada sob o mapeamento entre o nome e o conteúdo; um NDO, portanto, possui 3 elementos: o nome, o conteúdo e a assinatura. O cliente deve então obter a chave pública do publicador a fim de verificar tal assinatura. Através dessa verificação, o cliente pode certificar-se tanto da integridade do conteúdo quanto da proveniência do mesmo (RIBEIRO et al., 2012).

2.2 Content-Centric Networking

A arquitetura CCN faz parte das abordagens de arquiteturas ICN sendo ativamente estudadas (AHLGREN et al., 2012). Uma implementação de tal arquitetura é o projeto CCNx, desenvolvido no Centro de Pesquisas Palo Alto (PARC). Dada sua importância, esse projeto é utilizado como base na realização desse Trabalho de Graduação.

A seguir, os componentes arquiteturais do paradigma ICN apresentados previamente são retomados. Apresenta-se, para cada componente, a escolha de implementação utilizada no CCNx.

Nomeação. Na arquitetura CCNx, usa-se um esquema de nomeação hierárquico, assim como o endereço IP, sendo o caractere ‘/’ o delimitador dos componentes do nome. No entanto, contrariamente ao endereço IP, cada componente do nome pode ter um tamanho arbitrário. Os nomes assemelham-se a URIs, sendo legíveis para seres humanos; pelos roteadores CCNx, entretanto, os nomes não são interpretados, apenas comparados. Exemplificando, um arquivo de vídeo poderia possuir o nome `ccnx:/ufrgs.br/video.avi`.

Dentre as informações contidas em um componente do nome de um objeto, há potencialmente uma referência ao publicador; é importante ressaltar, contudo, que essa referência identifica o publicador e não seu endereço físico, seguindo os princípios de uma arquitetura ICN. Informações adicionais, que podem estar presentes em outros componentes do nome, incluem informações sobre a versão do objeto ou número de segmento. Como os roteadores em CCNx não interpretam nomes, o significado associado a esses é de responsabilidade das aplicações ou instituições que publicam os objetos, podendo seguir convenções globais.

Um nome em CCNx pode identificar diretamente um objeto ou, de maneira alternativa, identificar um nodo de uma árvore, o qual aponta para múltiplos objetos. A documentação do projeto CCNx argumenta que um nome identificando uma coleção de objetos, sendo esse nome um prefixo do nome de cada objeto dessa coleção, comporta-se de maneira análoga a um endereço de rede IPv4, o qual também pode ser um prefixo do endereço IP dos *hosts* pertencentes àquela rede.

Roteamento e Caching. Na arquitetura CCN, há dois tipos principais de mensagens: *Interest* e *Data*. Quando um nodo da rede está interessado em um conteúdo, ele envia uma mensagem do tipo *Interest*. Tal mensagem é composta pelo nome do objeto a ser requisitado, ou por um prefixo e um conjunto de restrições a serem aplicadas para indicar quais dos conteúdos apontados por esse prefixo são desejáveis. Apenas o nome (ou prefixo) do objeto desejado é obrigatório em uma mensagem *Interest*. Dentre as restrições, pode-se definir um número mínimo ou máximo de componentes do nome, limitar de onde a resposta virá, aconselhar a seleção do que enviar no caso em que múltiplos objetos correspondam ao requisitado, entre outros.

A mensagem de retorno, contendo o conteúdo requisitado pelo nodo, é uma mensagem do tipo *Data*, também denominada *Content* ou *Content Object*. Além do conteúdo em si, o qual pode ser opcionalmente cifrado, uma mensagem do tipo *Data* contém o nome do conteúdo, uma assinatura criptográfica e a identificação do publicador. Todas as mensagens do tipo *Data* devem obrigatoriamente conter uma assinatura válida.

A comunicação em CCNx é controlada pelo receptor. O receptor, interessado em um conteúdo, envia uma mensagem do tipo *Interest* às *faces* disponíveis, podendo utilizar broadcast ou multicast. Em CCN, uma *face* corresponde ao conceito de interface em redes de computadores. Qualquer nodo que possua o *Named Data Object* requisitado em sua cache, denominada *Content Store* (CS), e que satisfaça as condições impostas na mensagem de *Interest*, pode responder à requisição. A fim de evitar que dois nodos diferentes respondam a uma única requisição, mecanismos de supressão de mensagens do tipo *Interest* devem ser implementados pelos nodos. A versão atual do CCNx não especifica regras para gerenciar essa supressão, apenas recomenda que os mecanismos utilizados incluam tempos de resposta randômicos e que identifiquem quando um nodo já respondeu à requisição.

As mensagens do tipo *Interest* enviadas pelas aplicações são propagadas pela rede através da análise, utilizando uma lógica de *longest prefix match*, da tabela *Forwarding Information Base* (FIB), a qual está presente em todos os nodos que utilizam CCN. A FIB contém um mapeamento entre os *Interests* de entrada e as *faces* de saída adequadas.

Cada nodo que recebe um pacote *Interest*, antes de tomar a decisão sobre repassar o pacote, realiza uma busca na *Content Store* (CS). A CS deve implementar uma *flag*, conhecida como *Staleness Bit*, que é utilizada para indicar quando um objeto em cache não deve ser enviado como resposta a um *Interest*, a menos que seja especificado na requisição que tais conteúdos são aceitos. Os objetos marcados com o *Staleness Bit* são

os de maior prioridade em políticas de liberação de espaço da cache. Tal bit pode ser atualizado explicitamente ou através de um *timer*.

Cada nodo da rede possui ainda uma outra tabela, denominada *Pending Interest Table* (PIT), a qual é responsável pelo roteamento de um NDO até o cliente que o requisitou, seguindo o caminho reverso do *Interest*. Se o NDO desejado não for encontrado na CS, uma nova entrada é adicionada à PIT contendo a *face* pela qual o *Interest* foi encaminhado. Quando a requisição for satisfeita por um nodo, a mensagem do tipo *Data* retorna à aplicação requisitante através da análise de cada PIT no caminho ao receptor. Caso uma mensagem do tipo *Interest* chegue em um nodo que já apresenta uma entrada para aquele mesmo conteúdo na PIT, conclui-se que tal conteúdo já foi requisitado, mas que ainda não chegou ao destino. Nesse caso, a tabela PIT é atualizada adicionando-se a *face* pela qual o *Interest* chegou e a mensagem é descartada. Esse comportamento otimiza o uso da rede, visto que duas ou mais requisições pelo mesmo conteúdo, caso passem por um ponto de intersecção, não são distribuídas múltiplas vezes através da rede, como acontece na Internet atual.

Segurança. Como já citado anteriormente, todas as mensagens do tipo *Data* em CCNx devem obrigatoriamente conter uma assinatura válida, a qual será verificada pelas aplicações que receberem os NDOs. Para verificar a assinatura, pode-se precisar de *public keys*, que não estão incluídas na mensagem contendo os dados; no entanto, apesar de não haver um mecanismo destinado à distribuição de chaves, elas podem ser distribuídas como quaisquer outros dados em CCNx. Os autores do protocolo CCNx afirmam que o protocolo foi projetado para garantir entrega confiável de dados, sem limitar as decisões sobre distribuição de chaves e gerenciamento de segurança. Dessa maneira, a assinatura dos conteúdos é utilizada para confirmar que um objeto não foi corrompido em seu caminho até o destinatário. Visando maior segurança na rede, estimula-se o uso de políticas com a finalidade de determinar se um determinado publicador é confiável para um determinado tipo de conteúdo, as quais não estão especificadas diretamente no protocolo CCNx.

2.3 Trabalhos relacionados

Na literatura pesquisada, não foram encontradas outras iniciativas de investigação de como o relacionamento entre objetos pode melhorar o desempenho da CCN. Dois outros estudos, no entanto, apresentam similaridades com a solução proposta ou com seus objetivos, como discutido a seguir.

A arquitetura NetInf (KAUFFMANN; PELTIER; TRUONG, 2013) foi a primeira a considerar em seu projeto o conceito de relacionamento de objetos. Mais precisamente, NetInf introduz o conceito de *Information Objects* (IO), os quais consistem em coleções de metadados e ponteiros para os objetos propriamente ditos. Um cliente requisitando um conteúdo recebe inicialmente um IO, tendo acesso a ponteiros para as diferentes opções (se existentes) do conteúdo, como, por exemplo, versões de um vídeo que diferem em qualidade. Um IO pode mesmo ter um ponteiro para um outro IO. Além disso, os metadados contidos nos IOs permitem aos clientes realizar pesquisas semânticas de conteúdos publicados.

ICN-RE (PERINO; VARVELLO; PUTTASWAMY, 2012) busca otimizar o desempenho do sistema de *caching* através de um algoritmo de eliminação de redundância. O mecanismo calcula, através da aplicação de uma função hash no conteúdo, o *fingerprint* de cada *chunk*, o que permite a identificação de porções idênticas de bytes. Dessa ma-

neira, se dois objetos diferentes tiverem o mesmo *chunk fingerprint*, apenas um deles ficará armazenado em cache, evitando duplicações. A reconstrução do conteúdo original é possibilitada pela existência do arquivo *manifest*, o qual mantém uma lista dos *fingerprints* que constituem o objeto; tal arquivo é transferido ao cliente antes do envio do conteúdo.

O mecanismo proposto neste trabalho difere dos trabalhos citados em seus objetivos e avalia um modelo de relacionamento de objetos em arquiteturas CCN, quantificando seus benefícios através de experimentos em cenários realísticos. O trabalho desenvolvido em NetInf é específico a tal arquitetura e não pode ser aplicado ou estendido para CCN, visto que ambas arquiteturas não são compatíveis. INC-RE, por sua vez, possui uma abordagem diferente, focando em reduzir a redundância de conteúdos armazenados em roteadores ICN. Apesar de seus benefícios serem similares ao mecanismo proposto, há diferenças importantes. ICN-RE é transparente às aplicações, sendo a redundância identificada e eliminada pelos roteadores, e não conta com um mecanismo para relacionamento de objetos. A abordagem proposta, em contraste, possibilita o uso de relações cuja semântica pode ser definida em nível de aplicação.

3 PROJETO DO RELACIONAMENTO DE OBJETOS

Nesse capítulo, apresenta-se primeiramente, na Seção 3.1, um modelo para decompor conteúdos em objetos e de descrever como os mesmos estão associados, ilustrando a ideia de relações entre objetos proposta neste trabalho. Em seguida, na Seção 3.2 apresenta-se a API (do inglês, *Application Programming Interface*) necessária para permitir o funcionamento do modelo descrito. Na Seção 3.3, propõe-se um mecanismo de relacionamento de objetos para a arquitetura CCN, cuja implementação é explicada na Seção 3.4. Por fim, na Seção 3.5, apresenta-se exemplos de aplicações e descreve-se como o relacionamento de objetos seria por elas utilizado.

3.1 Conceito

Um objeto representa uma peça individual de informação na rede (isto é, um NDO) e uma relação, por sua vez, estabelece alguma associação entre dois objetos distintos. Um conteúdo é, então, o resultado da composição lógica de objetos orientada pela interpretação de um conjunto de relações. Há, entretanto, diversas possibilidades para a determinação de uma semântica para as relações, a qual é definida pela aplicação que publica e recupera um determinado conteúdo. Tal semântica é opaca à rede, evitando assim a adição de complexidade ao núcleo da mesma, a fim de não comprometer seu desempenho. A seguir, são discutidos três exemplos de como aplicações podem modelar o relacionamento entre objetos.

O primeiro exemplo consiste em uma relação simples entre dois objetos. A Figura 3.1 apresenta um vídeo V , o qual é um conteúdo composto por um objeto de vídeo o_v e por um objeto de legenda o_l relacionado. A relação r define que o_l é a legenda para o áudio em o_v .

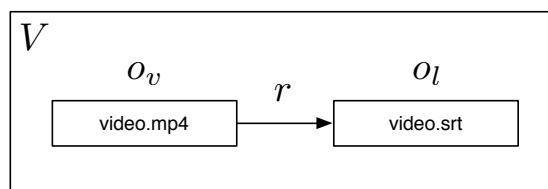


Figura 3.1: Exemplo de relacionamento simples

O segundo exemplo descreve uma situação onde um mesmo objeto está relacionado a múltiplos objetos diferentes. Mantém-se, nesse caso, um conjunto de relações $R = \{r_1, r_2, \dots, r_n\}$ definindo a associação que o objeto principal mantém com cada objeto o_1, o_2, \dots, o_n com o qual ele está relacionado. O relacionamento de objetos pode ainda,

nesse caso, atingir múltiplos níveis: um primeiro objeto pode estar relacionado a outros, os quais, por sua vez, estariam relacionados a um terceiro conjunto de objetos, gerando uma árvore de relacionamentos ou, se a aplicação permitir, um grafo – tendo o cuidado de evitar *loops* infinitos. A Figura 3.2 ilustra um exemplo de relações múltiplas: um vídeo V é composto por um objeto de vídeo (o_v) e por três outros objetos, dos quais dois são objetos de áudio (o_{a_en} e o_{a_fr}) e o terceiro corresponde a um objeto de legenda (o_{l_en}). As relações entre objetos estabelecem que a legenda e os objetos de áudio são versões alternativas ao áudio original do vídeo, o qual, por sua vez, também pode ser um objeto distribuído separadamente do vídeo. Nesse exemplo, um conjunto de objetos seria combinado para possibilitar que um usuário reproduza o vídeo.

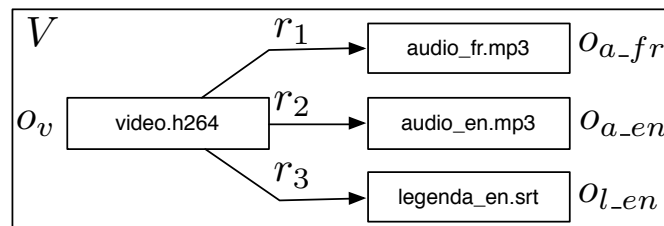


Figura 3.2: Exemplo de relacionamento múltiplo

O terceiro e último exemplo de relações entre objetos é usado para descrever o versionamento linear de um conteúdo. A Figura 3.3 ilustra o exemplo de um *log* L , o qual é representado por diversos objetos o_n , onde n é a versão do conteúdo. Objetos sucessivos contêm uma relação entre eles, definindo como os novos objetos são associados aos existentes. Por exemplo, duas alternativas poderiam ser consideradas: cada nova versão corresponderia a um conteúdo incremental em relação à versão anterior; ou cada nova versão corresponderia a um novo objeto que representaria o *log* em sua totalidade, independentemente dos objetos anteriores.

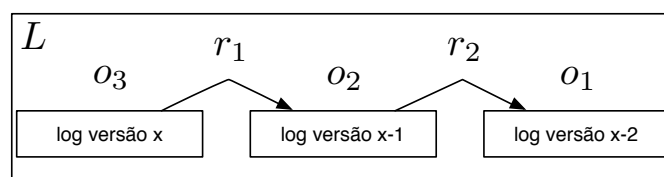


Figura 3.3: Exemplo de relacionamento linear

3.2 API

Apesar da semântica do relacionamento de objetos ser específica da aplicação, um modelo comum para a manipulação de relações facilitaria a sua adoção em larga escala. Para tanto, uma extensão para a arquitetura CCN deveria implementar o modelo descrito na seção anterior. Com tal extensão, um publicador deve poder criar, modificar e remover relações dos objetos publicados; um cliente, por sua vez, deve poder requisitar relações do objeto de interesse.

Visando permitir a realização das operações citadas, a API de um mecanismo para o relacionamento de objetos deve oferecer as seguintes funções:

- **Create.** Método que permite criar um conjunto de relações para um objeto publicado.
- **Modify.** Método que permite modificar o conjunto de relações previamente existente de um objeto publicado.
- **Remove.** Método que permite excluir o conjunto de relações definido para um objeto publicado.
- **Request.** Método que permite recuperar as relações que o objeto desejado possui com outros objetos da rede.

As funções descritas são a base para o projeto de um mecanismo de relacionamento de objetos para a arquitetura CCN. Na seção a seguir, apresenta-se tal mecanismo.

3.3 Mecanismo proposto

Dada a API a ser disponibilizada, é necessário definir o funcionamento de um mecanismo para o gerenciamento das relações. A seguir, discute-se aspectos importantes para a definição de tal mecanismo, apresentando-se as decisões tomadas.

Primeiramente, deve-se considerar como armazenar e publicar relações. Um método simples para implementar e tornar disponíveis os relacionamentos consiste em incluí-los nos dados do objeto. Tal método, no entanto, teria as seguintes limitações: (i) clientes teriam que recuperar o objeto antes de poder processar as relações, reduzindo sua eficiência; (ii) as funções de modificar e remover relações teriam um custo significativo, pois não haveria uma maneira de, após o objeto ter sido publicado, editar apenas as relações, seria necessário republicar o objeto por inteiro.

Uma alternativa às limitações citadas consiste em distribuir as relações de maneira independente dos dados do objeto, possibilitando que as mesmas sejam adicionadas, modificadas e removidas após a publicação do objeto. Dessa maneira, propõe-se, neste trabalho, a distribuição de relações através de metadados. Ao utilizar metadados, possibilita-se também que um cliente recupere as relações sem necessitar, para tanto, recuperar todo o conteúdo do objeto que as contém, minimizando a sobrecarga gerada pelo mecanismo.

Além de definir a maneira como as relações serão distribuídas, é importante definir um padrão para o formato usado para descrevê-las. Propõe-se, nesse trabalho, que o objeto que descreve as relações seja um arquivo de texto, no qual cada linha representa uma única relação do objeto. Relações são descritas através de dois elementos: uma *string*, correspondendo a um identificador que é usado para definir a semântica da relação, e o nome do objeto relacionado. Considerando, por exemplo, um objeto multimídia que possua canais de áudio relacionados em diferentes idiomas, o identificador poderia especificar, para cada relação, que o objeto relacionado consiste em um áudio e, além disso, indicar seu idioma.

A Figura 3.4 ilustra o uso de informações em metadados para definir e distribuir objetos de relações. Primeiro, um cliente faz uma requisição por um objeto de interesse, por exemplo o vídeo “Intro.mp4” (etapa 1). Entre a requisição e o recebimento do conteúdo, o cliente pode receber o conjunto de relações que tal conteúdo contém através da requisição do respectivo metadado (etapa 2). Após adquirir as relações, o usuário conhece todos os objetos relacionados ao conteúdo desejado e pode usar essa informação de acordo com a semântica das relações (etapa 3). Nesse exemplo, o cliente pode selecionar o áudio e a legenda desejados para complementar o vídeo.

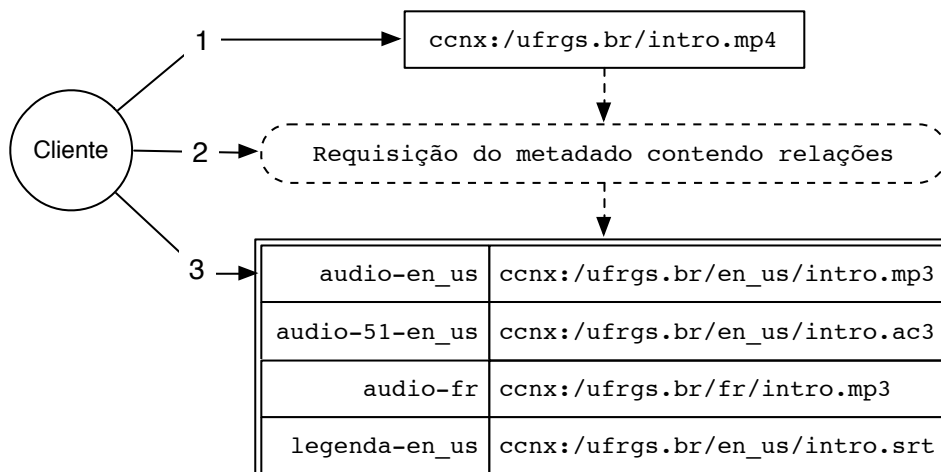


Figura 3.4: Distribuição de relações através de metadados

Por fim, considerando o gerenciamento das relações, conforme apresentado na seção anterior, o mecanismo permitirá que relações sejam criadas, editadas, removidas e eficientemente recuperadas. No entanto, é importante considerar quais entidades têm permissão para gerenciar as relações de um objeto. Duas alternativas são possíveis: (i) limitar as operações citadas à mesma entidade que publica um objeto ou (ii) possibilitar que múltiplas entidades tenham esse direito. Considerando a segunda alternativa, deve-se implementar um mecanismo de controle que evite problemas de inconsistência, os quais poderiam ser gerados pela atualização concorrente das relações por entidades diferentes. A fim de evitar tais problemas, opta-se pela primeira alternativa e limita-se, neste trabalho, o gerenciamento de relações de um objeto a seu publicador original.

3.4 Implementação

Implementa-se o mecanismo descrito anteriormente utilizando-se, como base, o mecanismo de metadados do protótipo CCNx, o qual permite a fácil distribuição de informações adicionais sobre os NDOs. A seguir, explica-se como a extensão é implementada.

O mecanismo de metadados de CCN trabalha com uma estrutura de nomeação bem definida: `/<nome do objeto>/%CI.META/<identificador>`. O marcador “%CI.META” determina que o objeto com tal nome consiste em um metadado, enquanto o componente “identificador” identifica o mesmo. Dessa maneira, através da publicação de metadados, pode-se distribuir diversas informações complementares juntamente com NDOs, como por exemplo, o *thumbnail* de um vídeo. Pode-se também, através de tal mecanismo, distribuir as relações; para tanto, é necessário apenas definir um padrão para identificar qual o metadado correspondente às relações e, assim, possibilitar sua requisição. Propõe-se, neste trabalho, que o identificador seja “relations”.

A função de criação de relações é implementada, dessa maneira, através da publicação de um arquivo texto – o qual contém as relações – como um metadado cujo identificador é *relations*. No CCNx, os objetos publicados na rede são armazenados em um repositório, o qual é responsável por preservá-los tal como instruído e por responder às mensagens do tipo *Interests* contendo requisições por objetos que ele possui. De maneira semelhante aos objetos, metadados também são armazenados em repositórios. Para publicar um metadado, deve-se, então, de acordo com o protocolo existente, enviar uma mensagem do

tipo *Interest* para indicar a ação desejada, a qual deve possuir o comando *Start Write* (%C1.R.sw) como um componente do nome. Tal comando indica que um repositório deve recuperar e armazenar um objeto. A resposta ao comando corresponde a uma mensagem do tipo *Data*, a qual é composta por um objeto do tipo INFO contendo detalhes sobre o repositório que aceitou a requisição. Por fim, o repositório recupera o conteúdo do publicador através do protocolo de mensagens padrão do CCNx. A Figura 3.5 ilustra o protocolo explicado.

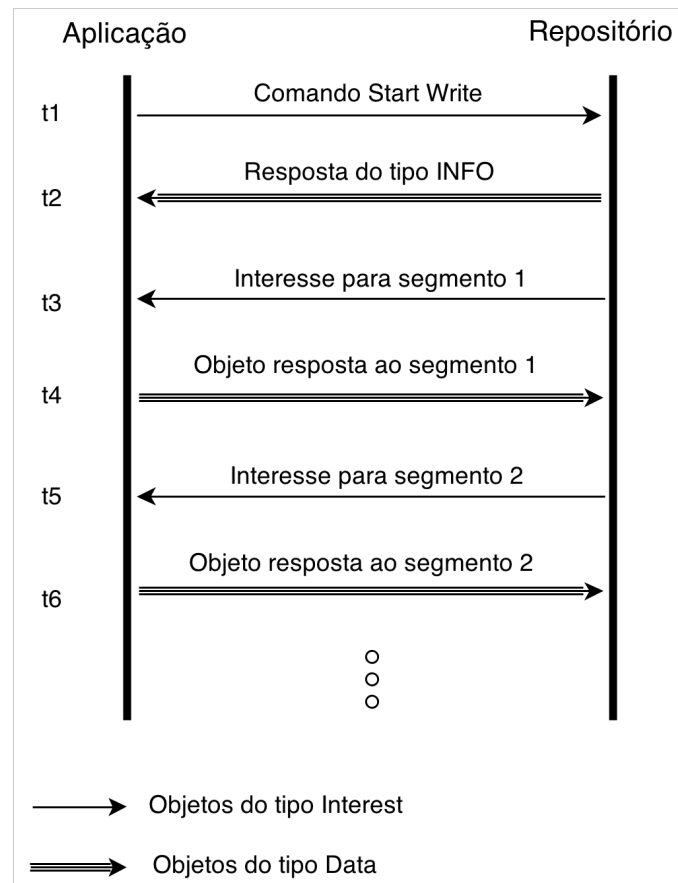


Figura 3.5: Comunicação para publicação de objetos em um repositório. Adaptado de (PARC, 2013a)

Quanto ao gerenciamento da publicação de relações, conforme já discutido anteriormente, restringe-se ao publicador do objeto original o direito de publicar relações sobre ele. Esse comportamento é compatível àquele imposto pelo mecanismo de metadados do CCN: apenas o publicador de um dado objeto pode publicar um metadado sobre ele.

As operações de edição e remoção, por sua vez, requerem que o objeto contendo as relações seja atualizado depois da publicação. Essas operações, por conseguinte, requerem o uso de um mecanismo de versionamento para a sua implementação. A arquitetura CCN possui um mecanismo padrão de versionamento, o qual adiciona ao nome do objeto um componente de versão baseado no horário de publicação. Apesar de simples comparado ao problema de versionamento de objetos em ICN (XYLOMENOS et al., 2013), esse mecanismo atende aos requisitos do sistema proposto nesse trabalho para o relacionamento de objetos. Assim, para editar relações, publica-se novamente o metadado *relations* através do mecanismo padrão disponível na arquitetura CCN. Quanto à exclusão de relações, implementa-se através da publicação de um arquivo de relações sem conteúdo, visto que,

conforme apontado por (BARI et al., 2012), os principais projetos de arquiteturas ICN não mencionam explicitamente um mecanismo para a remoção de objetos de dados; o projeto CCNx, igualmente, não fornece um meio de remover objetos publicados na rede.

Para a recuperação de relações, requisita-se o metadado que as contém. Para tal, obtém-se o nome do objeto contendo as relações através da estrutura anteriormente explicada: acrescenta-se ao final do nome do objeto a sequência */%CI.META/relations*. Após obter o nome do objeto, faz-se uma requisição pelo mesmo na rede e, uma vez recebido, pode-se interpretar as relações considerando o formato padrão de armazenamento das mesmas. Por exemplo, no mecanismo ilustrado na Figura 3.4, o nome utilizado para requisição das relações na etapa 2 seria *ccnx:/ufrgs.br/intro.mp4/%CI.META/relations*.

A implementação apresentada neste trabalho consiste em uma biblioteca a nível de aplicação que exporta uma API para publicação e requisição de relações. Dessa maneira, o mecanismo proposto para gerenciamento de relações tem a vantagem de não exigir mudanças na arquitetura CCN atual, nem em roteadores CCN. No entanto, considera-se como trabalho futuro investigar a possibilidade de estender o relacionamento de objetos ao nível de rede a fim de se estudar possíveis otimizações no roteamento e no uso das caches.

3.5 Aplicações

Nessa seção, apresenta-se cinco exemplos de aplicações que poderiam beneficiar-se do mecanismo de relacionamento de objetos descrito nesse capítulo. Inicia-se com uma aplicação que analisa a distribuição de vídeos em ICN, visando otimizar seu funcionamento. Em seguida, aborda-se uma aplicação que consiste na edição colaborativa de fotos. A terceira aplicação apresentada, por sua vez, trata da publicidade associada a conteúdos multimídia, enquanto a quarta analisa relacionamento de objetos na modelagem de mapas e pontos de interesse. Por fim, a última aplicação concentra-se na disseminação online de música.

3.5.1 Distribuição de vídeos

A distribuição online de vídeos tem um papel bastante significativo no uso atual da Internet. Com a evolução na velocidade de acesso dos usuários à rede, sites de notícias, por exemplo, passaram a publicar reportagens gravadas, semelhantes às apresentadas na televisão, ao invés de utilizar somente texto e imagens. Sites educacionais, por sua vez, também aproveitaram-se da possibilidade de gravar e distribuir vídeos online para tornar aulas de instituições bastante renomadas, como o MIT, acessíveis para qualquer usuário que desejar assistir a elas. Outros exemplos são sites como o Youtube ou sites de *streaming*, que concentram-se na distribuição de vídeos de diversas origens e contam com um número expressivo de usuários.

Utilizando-se relacionamento de objetos em ICN, pode-se remodelar a maneira como vídeos são publicados na Internet a fim de aumentar a eficiência de sua distribuição. Conforme ilustrado na Figura 3.2, na Seção 3.1, um conteúdo representando um vídeo poderia ser composto por diversos objetos, os quais corresponderiam ao objeto de vídeo e a objetos representando diferentes possibilidades de áudio e legenda.

Quando um usuário requisitar um conteúdo de vídeo, a primeira ação da aplicação será requisitar pelo objeto de vídeo e pelo metadado contendo as relações. Após analisar as opções existentes para o áudio e para a legenda, as quais estarão descritas no objeto *relations*, a aplicação requisitará os objetos correspondentes aos idiomas desejados pelo

cliente. Tendo recebido a totalidade do conteúdo (nesse caso, 3 objetos: vídeo, áudio e, opcionalmente, a legenda) a aplicação poderá então reproduzir o vídeo. Caso o usuário decida mudar o idioma do vídeo durante a reprodução do mesmo, é suficiente que a aplicação recupere o novo objeto de áudio (ou de legenda), podendo aproveitar o objeto de vídeo que já possui.

Tal modelagem permite uma redução do tráfego da rede em comparação a um cenário onde objetos de vídeo são distribuídos com áudio embutido. Devido ao reuso de um mesmo objeto para diferentes conteúdos – por exemplo, o mesmo objeto de vídeo sendo usado para uma versão do áudio em inglês e para uma versão do áudio em português – aumenta-se a popularidade do objeto de vídeo e, também, a taxa de acerto da cache.

3.5.2 Edição colaborativa de fotos

Atualmente, um conceito bastante popular entre os usuários da Internet é o compartilhamento de fotos. Dois exemplos são o Flickr e o Instagram. Nesse último, a aplicação consiste em uma rede social que permite a seus usuários aplicar facilmente filtros em fotos e, em seguida, publicar a imagem resultante. Cada usuário pode optar entre compartilhar suas imagens com qualquer pessoa que queira visualizá-las ou apenas com uma lista restrita de usuários que ele aceita que o “sigam”. Um usuário pode também seguir outros, a fim de visualizar as imagens postadas por eles.

Pode-se, em ICN, estender a ideia existente no Instagram para uma aplicação de edição colaborativa de fotos, utilizando em sua modelagem o conceito de relacionamento de objetos. A aplicação considerada, assim como o Instagram, possibilitaria aos usuários optar por tornar suas fotos públicas ou compartilhadas apenas com uma lista restrita de usuários. As fotos publicadas, entretanto, não seriam exclusivas ao usuário que as publicou: outros usuários poderiam aplicar novos filtros à imagem original, gerando novas versões da foto. A fim de possibilitar a edição colaborativa descrita, o usuário carregaria, inicialmente, a foto original na aplicação. Uma lista de possíveis filtros e opções de edição seria então apresentada ao usuário, que escolheria o de sua preferência e publicaria a imagem resultante.

Tal aplicação seria implementada com base no relacionamento de objetos. Idealmente, deve-se permitir aos usuários editar a foto original; logo, a foto publicada não deve ser um arquivo único já editado. Dessa maneira, uma foto é composta por dois objetos, a foto original que o usuário carregou na aplicação e um objeto contendo uma lista dos filtros que foram aplicados na versão em questão, sendo este último relacionado ao primeiro, assim como mostrado na Figura 3.6. Um outro usuário que tenha gostado da foto e que se interesse por editá-la poderá, então, recuperar a imagem original e começar a editá-la sem influências de edições anteriores.

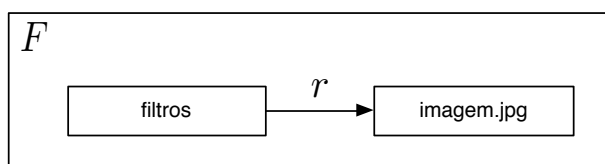


Figura 3.6: Modelagem de uma imagem utilizando relacionamento de objetos

Assim como no caso da distribuição de vídeos apresentada anteriormente nesse trabalho, espera-se que essa aplicação proporcione um melhor desempenho da rede comparado à implementação de uma aplicação semelhante sem o relacionamento de objetos. Como

não serão publicados novos objetos de imagem a cada nova versão, e sim apenas novos objetos contendo a lista de edições aplicadas, espera-se reduzir o tráfego global da rede. Tal redução explica-se devido ao reuso de objetos (imagens originais) em diferentes conteúdos, influenciando em sua popularidade e potencialmente aumentando a taxa de acerto da cache. A economia trazida pelo mecanismo, conseqüentemente, será proporcional ao tamanho das imagens que estão sendo compartilhadas. No entanto, quando um usuário deseja visualizar uma foto editada publicada, introduz-se um custo de processamento no cliente devido à necessidade de aplicar os filtros descritos; seria, por conseguinte, necessário avaliar o impacto de tal processamento.

3.5.3 Publicidade em vídeos

Com o crescente aumento da visualização de conteúdo multimídia na Internet, substituindo, de certa maneira, a televisão (CAUMONT, 2013), algumas aplicações investiram na inserção de propagandas no início e no final de vídeos. Um exemplo é o Youtube, que torna obrigatório aos usuários assistir no mínimo 5 segundos de publicidade antes da reprodução do conteúdo realmente desejado. O usuário não tem, no entanto, escolha quanto à propaganda que irá assistir, visto que a mesma é imposta pela aplicação que distribui o vídeo.

Almejando uma melhor satisfação dos usuários, pode-se, utilizando relacionamentos de objetos, implementar em ICN uma alternativa onde os clientes recebam uma lista de possíveis propagandas e escolham entre elas, de acordo com seu interesse, qual preferem assistir. Essa solução proporciona aos usuários um maior controle sobre o que assistem, reduzindo a probabilidade de serem obrigados a assistir a propagandas incompatíveis com seu perfil (por exemplo, uma mulher provavelmente não está interessada em assistir à propaganda de um gel de barbear). Utilizar-se-ia, idealmente, apenas informações de localização do usuário para determinar as propagandas a serem oferecidas.

Para tanto, existiriam objetos abstratos na rede, que corresponderiam, cada um, a uma lista de opções, de acordo com algum perfil; por exemplo, um objeto conteria uma lista de propagandas para usuários localizados no sul do Brasil enquanto outro objeto conteria uma lista de propagandas para usuários na França. Tais objetos manteriam relações para os objetos correspondentes às propagandas citadas nas opções e seriam publicados por alguma entidade controladora da aplicação. Assim, a fim de atualizar a lista de opções de acordo com o surgimento de novas propagandas, basta que tal entidade atualize o metadado contendo as relações.

Resumidamente, a aplicação funcionaria da maneira descrita a seguir. Antes de reproduzir um vídeo, a aplicação buscaria pelos objetos abstratos descritos anteriormente, escolhendo o que apresentasse opções de acordo com o perfil do usuário. Em seguida, ela apresentaria a lista de opções obtida, possibilitando a escolha de uma propaganda pelo usuário. Por fim, tendo o nome do objeto correspondente registrado no metadado de relações, a aplicação requisitaria a propaganda escolhida.

A principal vantagem do uso de relacionamento de objetos para essa aplicação consiste em facilitar a implementação do mecanismo descrito, permitindo ao usuário escolher uma propaganda de interesse e recuperá-la rapidamente: para tal, é suficiente a manutenção de objetos abstratos na rede que forneçam as listas de opções, mantendo relações para os vídeos de cada propaganda oferecida. Quanto ao desempenho da rede, no entanto, acredita-se que não haveria um ganho significativo na taxa de acerto da cache e, conseqüentemente, no tráfego total, visto que, nesse caso, não haveria uma divisão de conteúdos onde o reuso de objetos seria favorecido.

3.5.4 Serviço online de mapas

Uma das funcionalidades de destaque de aplicações online de mapas consiste em mostrar, para cada região do mapa, um conjunto de pontos de interesse (POI, *Point of Interest*). Tais POIs representam locais, monumentos históricos e estabelecimentos comerciais, entre outros, que potencialmente são relevantes para os usuários. Cada POI contém informações sobre o ponto que representa, como a pontuação dada pelos usuários e suas opiniões, telefone de contato, endereço, informações úteis, etc. Dois exemplos de aplicações que trabalham com pontos de interesses em seus mapas são o Google Maps e o Trip Advisor.

A modelagem de uma aplicação semelhante em ICN é facilitada pelo uso de relacionamento de objetos. Cada POI consistiria em um objeto contendo todas as informações sobre aquele ponto de interesse. Ainda, cada região do mapa, em uma determinada escala, corresponderia a um objeto diferente. Os objetos representando as regiões manteriam, então, relações para os pontos de interesse que nelas estão presentes. A Figura 3.7 ilustra tal modelagem: dada uma região de mapa RM , o objeto que corresponde à região (R) mantém relações (r_1, r_2, r_3, r_4 e r_5) para os diferentes pontos de interesse (POI) existentes.

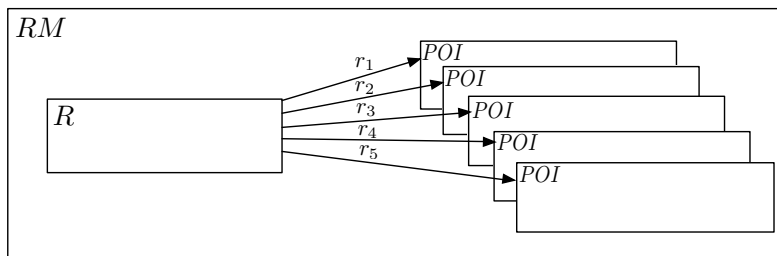


Figura 3.7: Modelagem de uma região de mapa utilizando relacionamento de objetos

Quanto à manutenção dos pontos de interesse, deve haver uma entidade centralizada da aplicação que publique todos as regiões do mapa, a fim de tornar possível a inserção de novos POI e edição ou remoção de POIs existentes. Existindo tal entidade, é suficiente que ela atualize os metadados correspondentes às relações quando houver necessidade de executar alguma das operações citadas.

Considerando um trabalho futuro onde roteadores ICN seriam modificados para compreender algumas relações, poder-se-ia implementar uma condição estabelecendo que quando um objeto correspondendo a uma região se encontra em cache, deve-se populá-la também com os pontos de interesse relacionados. Dessa maneira, quando o usuário requisitar o acesso a POIs, eles já estariam cache. No entanto, antes de implementar tal alternativa, deve-se estudar os custos adicionais que seriam gerados, a fim de definir sua viabilidade.

3.5.5 Distribuição de conteúdo multimídia

Atualmente, há um número expressivo de usuários interessados na distribuição online de músicas. Respondendo a essa demanda de mercado, surgiram aplicações inovadoras, tais como LastFM, Spotify e Rdio. Em tais aplicações o usuário pode ouvir músicas avulsas, álbuns completos de artistas, listas de reprodução organizadas por gênero musical, entre outros. O usuário pode mesmo criar listas de reprodução próprias que ficam acessíveis a seus amigos, ou públicas, em um comportamento semelhante àquele de redes sociais.

Discografias, álbuns e listas de reprodução podem ser modelados de forma natural em

ICN utilizando relacionamento de objetos. Basta que, para tanto, cada música seja um objeto único e um álbum seja um objeto contendo relações para os objetos correspondentes às faixas do mesmo. Uma lista de reprodução, igualmente, seria um objeto contendo relações para os objetos de músicas pertencentes à lista ou mesmo para objetos de álbuns, se aplicável. Uma discografia consistiria em um objeto contendo relações para os álbuns que a compõem. Trata-se de um agrupamento de músicas em discos, e discos em discografias, bastante popular em redes de compartilhamento de arquivos, como BitTorrent.

Dessa maneira, a disseminação de músicas via ICN seria mais facilmente implementada; cada novo álbum, ou nova lista de reprodução, consistiria em apenas um novo objeto comparativamente leve na rede que conteria relações para os objetos de música. As faixas musicais, por sua vez, seriam únicas; não existiriam, por exemplo, dois objetos da mesma música em dois servidores distintos, como poderia acontecer nas redes atuais. Influencia-se então na popularidade dos objetos, potencialmente melhorando o desempenho da rede, através do aumento da taxa de acerto da cache.

Ademais, assim como no caso dos POIs em aplicações de mapas explicadas anteriormente, considerando-se um cenário onde roteadores ICN possam compreender a semântica de relações, pode-se otimizar a distribuição de objetos na rede. No caso de um álbum estar em cache, poder-se-ia estudar os impactos de manter, igualmente, todas as músicas que o compõem em cache, analisando o tempo de espera do usuário para receber uma determinada faixa.

4 METODOLOGIA

Visando avaliar e quantificar os benefícios do mecanismo de relacionamento de objetos proposto neste trabalho, procura-se responder a três questões:

- Q1:** Em quanto pode-se reduzir o tráfego de rede se relações forem utilizadas?
- Q2:** Qual é o impacto do uso de relações para os componentes principais de CCN?
- Q3:** A existência de relações afetará positivamente a qualidade de experiência (QoE) do usuário?

Para avaliar tais pontos, escolheu-se a aplicação de distribuição de vídeos descrita na Seção 3.5.1, através da implementação de um serviço de *streaming* em C++. Utilizou-se em tal desenvolvimento a API em C do CCNx e a implementação do mecanismo proposto, descrita na Seção 3.4. No restante deste capítulo, apresenta-se o ambiente padrão para a execução dos experimentos e, em seguida, descreve-se os principais cenários e métricas utilizados na avaliação.

4.1 Ambiente padrão

Para realizar a avaliação, utiliza-se um ambiente de emulação composto por 2 servidores. Cada servidor possui um processador Intel Xeon com 6 núcleos de processamento físicos (12 lógicos) operando a 1,9GHz, 32GB de memória e duas interfaces de rede. Distribui-se, em ambos os servidores, 27 máquinas virtuais para a execução de roteadores CCNx e clientes. As máquinas virtuais contam com Debian 7.1 e CCNx 0.8.1 (PARC, 2013c) instalados.

A rede virtual utilizada é baseada na topologia atualmente empregada na rede Ipê, a rede acadêmica brasileira (RNP, 2013a), a qual é mantida pela Rede Nacional de Ensino e Pesquisa (RNP). Conforme apresentado na topologia da Figura 4.1, tal rede é composta por 27 pontos de presença (PoPs, *Points of Presence*), sendo um por estado, os quais são interconectados por *links* de até 10Gbps. Nela, conectam-se praticamente todas as unidades de pesquisa e instituições públicas de ensino superior do Brasil, além de outras organizações de ensino e pesquisa públicas e privadas; há, no total, cerca de 800 instituições utilizando sua infraestrutura.

A fim de possibilitar a execução de experimentos no *hardware* disponível, limitou-se o número de clientes a 130, os quais compartilham larguras de banda entre 1 e 100Mbps (proporcionais às larguras de banda do *backbone* original), e possuem latências realísticas de acordo com medidas realizadas pela própria RNP (RNP, 2013b). Posiciona-se o publicador dos vídeos em MG, a partir do qual gera-se uma árvore de distribuição com caminho

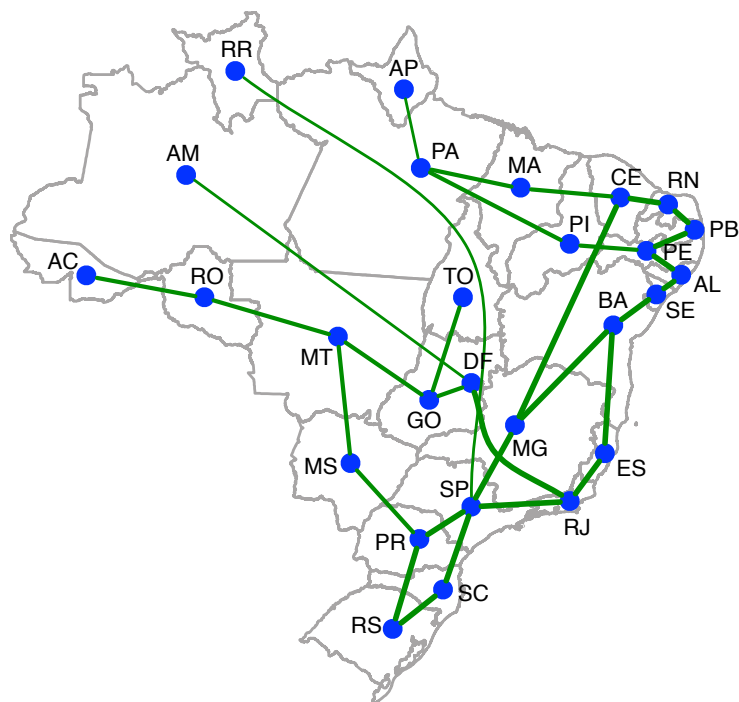


Figura 4.1: Mapa da topologia RNP

mínimo aos outros roteadores, conforme mostrado na Figura 4.2. MG foi escolhido pois este roteador representa o ponto central da topologia de acordo com o índice de centralidade de vértices; é importante observar, porém, que qualquer outro roteador poderia ter sido escolhido. Os clientes que requisitarão os vídeos publicados são distribuídos entre os demais 26 nodos, iniciando por 1 cliente por nodo e chegando em até 5 clientes por nodo, ou seja, 130 no total. Cada nodo da topologia executa também um roteador CCNx, o qual, com base nos resultados apresentados em (WANG; ZHANG; BENSOU, 2013), é configurado com um tamanho de cache que corresponde a 6% do conteúdo distribuído na rede.

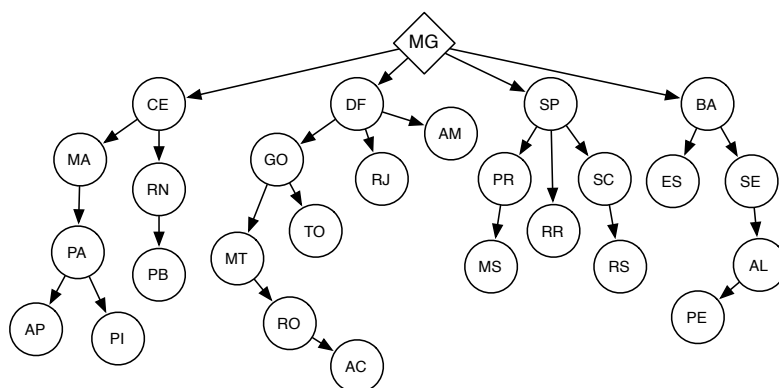


Figura 4.2: Árvore de distribuição da topologia experimental

Conforme apresentado anteriormente, os experimentos baseiam-se na disseminação de múltiplos conteúdos de vídeo. Nos cenários avaliados, distribui-se até 6 objetos de vídeo diferentes na rede, os quais, combinados através de relações com áudios e legendas em diferentes idiomas, resultam em até 66 conteúdos distintos. Para distribuir as requi-

sições pelos conteúdos entre os clientes, faz-se um sorteio baseado na Lei de Zipf; tal lei consiste em uma distribuição de probabilidades na qual a frequência de um item é inversamente proporcional a sua posição em uma lista de frequências. Para definir o intervalo entre as requisições de diferentes clientes, utiliza-se uma distribuição de Poisson. Os parâmetros utilizados para a reprodução do *streaming* são semelhantes àqueles observados no Youtube (FINAMORE et al., 2011). Clientes armazenam em *buffer* pelo menos 6 segundos antes de começar a reprodução do vídeo. Sempre que um cliente estiver reproduzindo um vídeo e o *buffer* ficar vazio, o vídeo trava, e a reprodução só recomeça quando houver dados para continuá-la. A taxa de reprodução de conteúdos é baseada na taxa de um vídeo 480p: a taxa total é de 1,5Mbps, dos quais 128Kbps correspondem ao áudio.

4.2 Cenários

Na avaliação do mecanismo proposto, define-se dois cenários principais a serem analisados, os quais são descritos nesta seção. Em ambos cenários, cada cliente requisita apenas um conteúdo. A escolha de cada cliente é dada através de um sorteio baseado em uma distribuição Zipf dos conteúdos disponíveis. Há, na rede, um máximo de 11 combinações de áudio e legendas diferentes para cada vídeo. Analisando as possibilidades de idiomas populares em uma rede brasileira, disponibiliza-se 5 opções de áudios: português, inglês, espanhol, francês e alemão. Para essas cinco opções de áudios, disponibiliza-se também uma versão com legenda em português. Para a versão em inglês, há também a possibilidade de assistir ao filme com legenda em inglês.

No primeiro cenário avaliado, fixa-se o número de clientes em 130 e o número de objetos de vídeos diferentes em 6. Varia-se, no entanto, entre 7 e 11, o número de diferentes idiomas para cada objeto de vídeo. Dessa maneira, tem-se uma variação de 42 até 66 conteúdos na rede. No segundo cenário, fixa-se o número de vídeos em 3, cada um com as 11 possíveis combinações, e varia-se o número de clientes por nodo entre 1 e 5, correspondendo a uma variação de 26 até 130 no número total de clientes.

Para cada um dos cenários apresentados, executa-se dois conjuntos de experimentos: no primeiro, conteúdos são modelados de maneira convencional (um objeto único para cada combinação de vídeo, áudio e legenda) e, no segundo, conteúdos são modelados utilizando-se relacionamento de objetos. Quando relações são usadas, o cliente requisitará inicialmente o metadado que as contém e, após análise do mesmo, requisitará o áudio e legenda desejados.

4.3 Métricas

Com base nas questões propostas no início deste capítulo, quatro métricas foram definidas para avaliar os experimentos, as quais são explicadas a seguir.

Total de dados transferidos. Essa métrica representa o volume total de dados transferido na rede, a fim de analisar a eficiência de empregar-se o relacionamento de objetos. Seu valor é a soma do tráfego dos *links* entre roteadores da topologia da rede. Devido à disseminação de conteúdo focada em um uso mais eficiente da cache, espera-se que a quantidade de dados transferida na rede seja reduzida pela utilização de relacionamento de objetos.

Taxa de acerto da cache. Essa métrica avalia o número de requisições que os roteadores podem responder com cópias de objetos mantidas em suas caches. Tal taxa de

acerto indica quão eficiente e efetiva está sendo a presença de caches na rede. Altos valores para essa métrica implicam em menos mensagens do tipo *Interest* sendo encaminhadas para outros nodos, reduzindo assim o tráfego da rede. Usando relacionamento de objetos, espera-se que as caches tornem-se mais estáveis e atinjam valores mais altos para a taxa de acerto.

Tempo de inicialização. Representa o intervalo entre a primeira requisição do cliente e o início da reprodução do vídeo. Valores baixos indicam que a rede é capaz de disponibilizar o *streaming* aos clientes de maneira eficiente. Espera-se que o tempo de carregamento seja reduzido com o uso de relações devido à disseminação de conteúdo mais eficiente na rede.

Número de travamentos. Um travamento é causado por um *buffer* vazio no cliente. O número de travamentos é mais uma métrica que indica a eficiência da rede para distribuir conteúdos e, assim como o tempo de inicialização, está relacionada com a QoE do usuário. Um valor alto indica que a rede não é capaz de garantir a taxa de reprodução necessária para o *streaming*. Espera-se que a solução proposta melhore a eficiência da rede, reduzindo, conseqüentemente, o número de travamentos.

5 RESULTADOS

Nesse capítulo, apresenta-se os resultados obtidos na avaliação experimental através da análise das métricas descritas na Seção 4.3. Aborda-se primeiramente, na Seção 5.1, resultados relacionados ao desempenho da rede. Em seguida, na Seção 5.2, analisa-se os resultados obtidos no que diz respeito à qualidade de experiência do usuário. Por fim, na Seção 5.3, apresenta-se considerações finais sobre os resultados obtidos, apresentando respostas para as perguntas de pesquisa propostas no Capítulo 4.

Para avaliar o mecanismo, realizou-se dois conjuntos de experimentos. No primeiro, variou-se o número de clientes, mantendo-se fixo o número de conteúdos distribuídos na rede; foram executadas 30 rodadas para o caso convencional e 30 rodadas para o caso utilizando o mecanismo de relacionamento de objetos. No segundo conjunto de experimentos, variou-se o número de conteúdos na rede, mantendo-se o número de clientes fixo; executou-se 25 repetições para cada caso, convencional ou com relações. Cada gráfico gerado apresenta, por conseguinte, duas curvas: uma representando o resultado de experimentos com o mecanismo de relações proposto e outra, o comportamento convencional de redes CCN, sem relacionamento de objetos. As barras de erros apresentadas em tais curvas, por sua vez, representam o intervalo de confiança para um grau de 95%.

5.1 Desempenho da rede

Inicia-se a análise do impacto do mecanismo proposto no desempenho da rede através da análise do tráfego total. No primeiro cenário avaliado, conta-se com 3 vídeos distribuídos na rede, cada um com 11 opções de combinações entre áudios e legendas; logo, há um total de 33 conteúdos. O gráfico da Figura 5.1 apresenta, para tal cenário, o tráfego total da rede – contabilizado em Megabytes no eixo vertical – de acordo com o número de clientes, o qual varia entre 26 e 130. Os resultados mostram um comportamento intuitivo: conforme o número de clientes cresce, o tráfego total da rede aumenta em ambos os casos. Tal aumento explica-se pelo fato de que cada cliente requisita um conteúdo; logo, quanto mais clientes, maior o número de requisições por diferentes objetos – as quais originam-se em diferentes nodos da topologia – e maior o tráfego. Ademais, os resultados confirmam que através do uso de relacionamento de objetos, reduz-se o tráfego total da rede em relação ao comportamento convencional. Observa-se ainda que a vantagem constatada com o uso do mecanismo proposto aparece em todos os casos e aumenta de acordo com o crescimento do número de clientes; para 26 clientes, reduziu-se o tráfego da rede em 12,8%, enquanto para 130, em 34,4%.

Tal redução no tráfego total da rede ao utilizar-se relações pode ser explicada devido ao reaproveitamento de um mesmo objeto por diferentes conteúdos. Com a modelagem proposta, aumenta-se a probabilidade de encontrar, em cache, um determinado objeto de

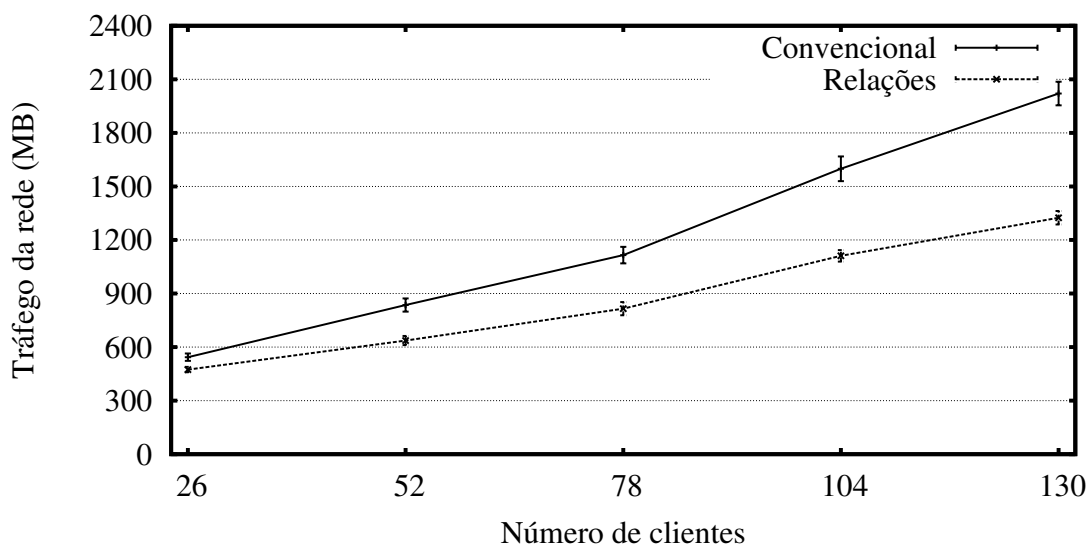


Figura 5.1: Impacto de variar o número de clientes no tráfego total da rede

interesse. Ao recuperar um objeto em uma cache próxima, reduz-se então o número de *hops* necessários para obtê-lo e, conseqüentemente, o tráfego total da rede. A Figura 5.2 apresenta os resultados obtidos em relação à taxa de acerto da cache, comprovando que houve uma melhoria com o uso do mecanismo de relacionamento de objetos. Essa melhoria cresce de maneira logarítmica em relação ao aumento na quantidade de clientes na rede: para 26 clientes, a taxa de acerto da cache é elevada em 25,8%; para 130, em 48,9%. O comportamento logarítmico é explicado pelo aumento na rotatividade dos conteúdos da cache, o qual limita o crescimento da taxa de acerto. Tal aumento na rotatividade deve-se tanto ao aumento do número de clientes e, conseqüentemente, do número de requisições por diferentes objetos, quanto ao intervalo de tempo existente entre requisições.

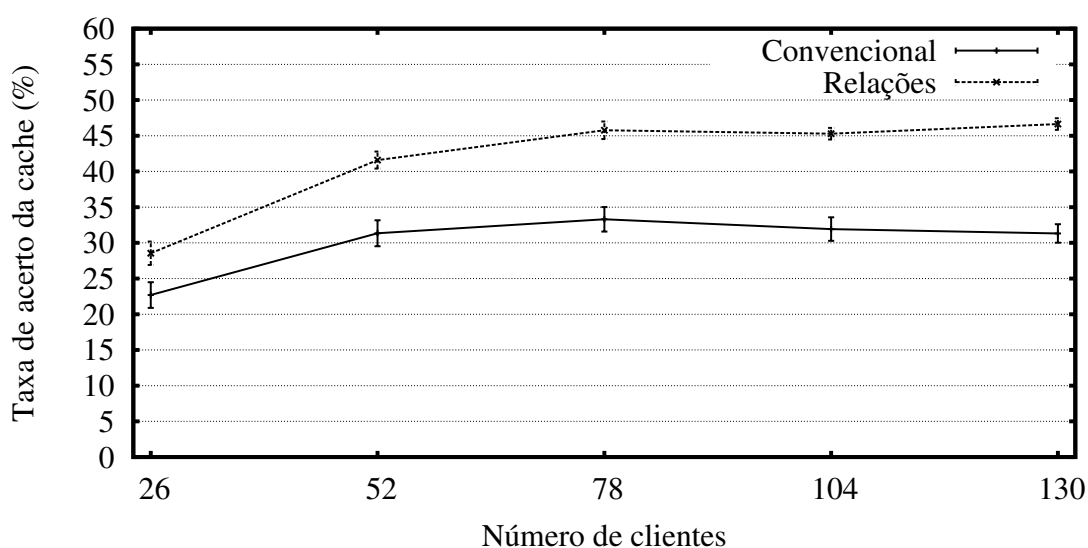


Figura 5.2: Impacto de variar o número de clientes na taxa de acerto da cache

Analisa-se, adicionalmente, os resultados obtidos em relação ao tráfego da rede no cenário no qual varia-se o número de conteúdos e mantém-se fixo o número de clientes,

os quais são apresentados na Figura 5.3. Novamente, o tráfego total está apresentado no eixo vertical, em Megabytes. Devido à Lei de Zipf utilizada para distribuir as requisições, a qual atribui uma popularidade bastante superior a poucos conteúdos, obtém-se um valor aproximadamente constante para o tráfego total. Isso ocorre pois as requisições concentram-se em poucos conteúdos, os de maior popularidade, e a probabilidade de encontrar o objeto desejado em cache se torna parecida tanto para 42 quanto para 66 conteúdos. Pode-se constatar, entretanto, que o mecanismo proposto ocasionou uma redução no tráfego em comparação com o mecanismo convencional existente, independente da quantidade de conteúdos distribuídos na rede. A taxa de acerto da cache também apresentou melhorias com o uso do mecanismo de relacionamento de objetos, conforme apresentado na Figura 5.4; tal resultado está de acordo com o esperado, visto que a taxa de acerto da cache influencia de maneira direta o tráfego da rede.

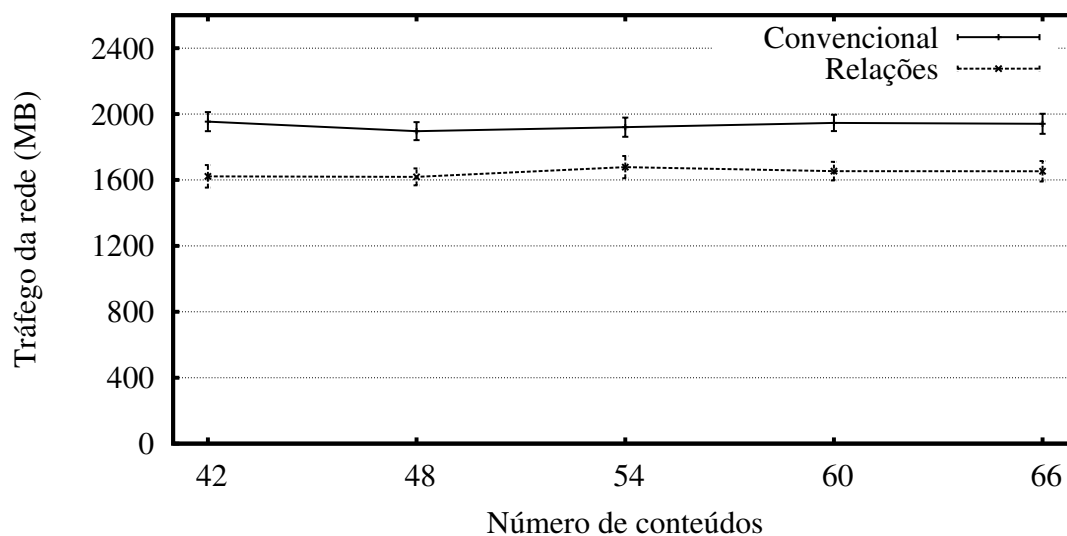


Figura 5.3: Impacto de variar o número de conteúdos no tráfego total da rede

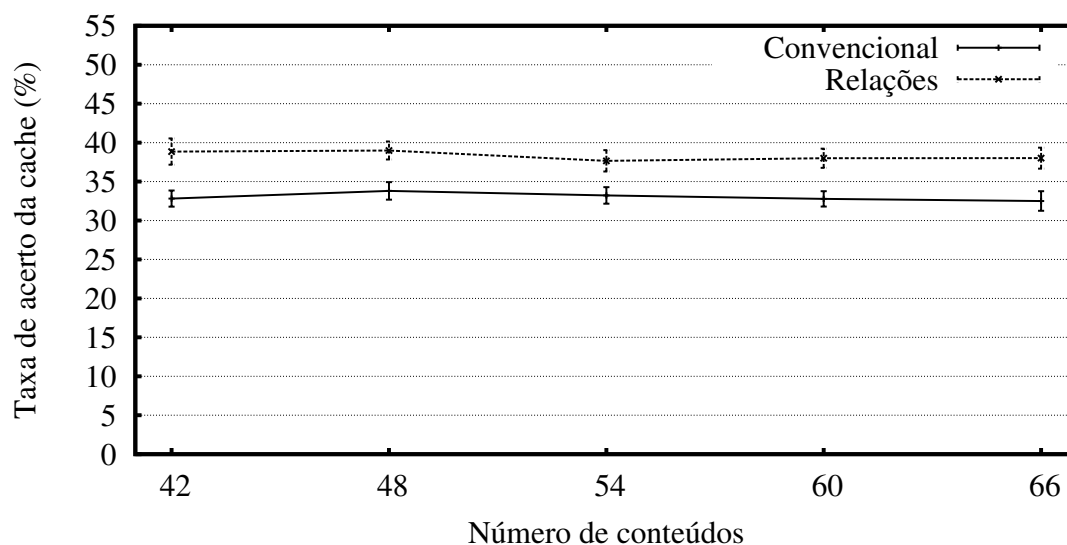


Figura 5.4: Impacto de variar o número de conteúdos na taxa de acerto da cache

5.2 Qualidade de Experiência

Dentre as métricas citadas no Capítulo 4, utiliza-se o número de travamentos e o tempo de inicialização para avaliar o impacto do mecanismo proposto na qualidade de experiência do usuário. O número de travamentos (*stalls*) indica quantas vezes houve uma interrupção na reprodução do vídeo devido à falta de dados. Quanto mais travamentos ocorrerem, ou quanto mais duradouros forem os períodos de recuperação, maior será o impacto para a experiência do usuário. O tempo de inicialização indica o tempo que um usuário tem que esperar, a partir do momento que requisitou um vídeo, para poder iniciar a reprodução do mesmo.

A Figura 5.5 apresenta os resultados obtidos em relação à quantidade de *stalls* ocorridos nos clientes durante a execução do *streaming* de vídeo para o cenário no qual varia-se o número de clientes. Em tal gráfico, o eixo horizontal apresenta uma variação no número de clientes enquanto o eixo vertical, o número de ocorrências de travamentos. Nota-se um aumento neste número decorrente do aumento da quantidade de usuários requisitando os vídeos, comportamento que se justifica pela maior necessidade de recursos da rede comparado a um cenário com poucos clientes. Observa-se também, através da análise dos intervalos de confiança obtidos, que o mecanismo convencional apresenta uma maior instabilidade nos resultados conforme o número de clientes aumenta, enquanto o mecanismo proposto mantém-se mais estável.

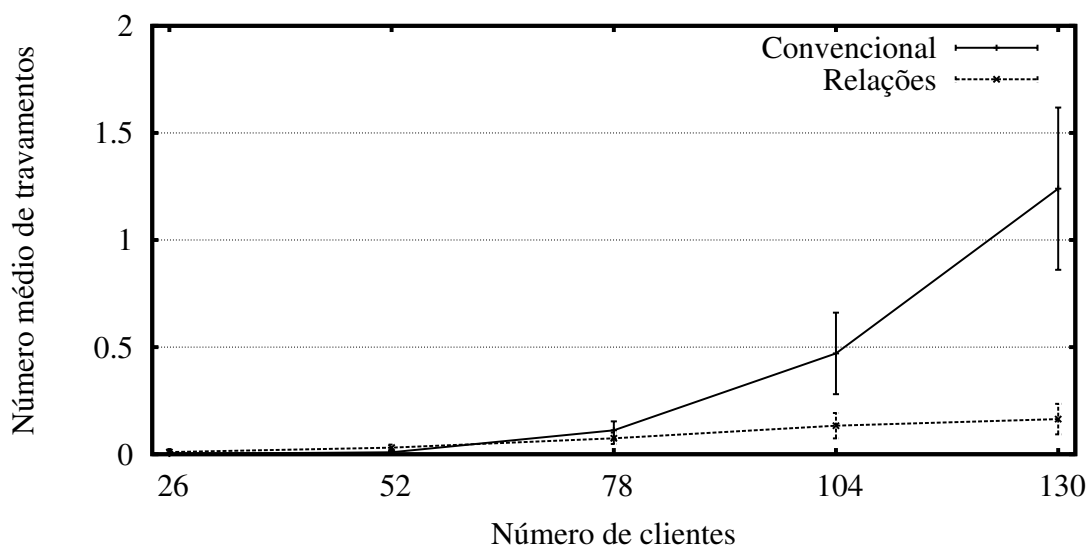


Figura 5.5: Impacto de variar o número de clientes no número de travamentos

Quanto ao tempo de inicialização do *streaming*, para o mesmo cenário no qual varia-se o número de clientes, obteve-se os resultados apresentados na Figura 5.6. Observa-se que a partir de 104 clientes obtém-se uma melhoria em tal tempo através do uso de relacionamento de objetos, fator que influencia positivamente na qualidade de experiência do usuário, que terá que aguardar durante menos tempo para iniciar a reprodução do vídeo desejado. Ademais, a diferença entre as inclinações das curvas do caso convencional e do caso com relações entre 104 e 130 usuários leva a acreditar que, para um maior número de clientes, o mecanismo proposto tende a ser ainda mais eficiente que o mecanismo convencional.

Considerando o cenário onde varia-se o número de conteúdos e mantém-se um número fixo de 130 clientes, obteve-se também uma melhoria no número de travamentos,

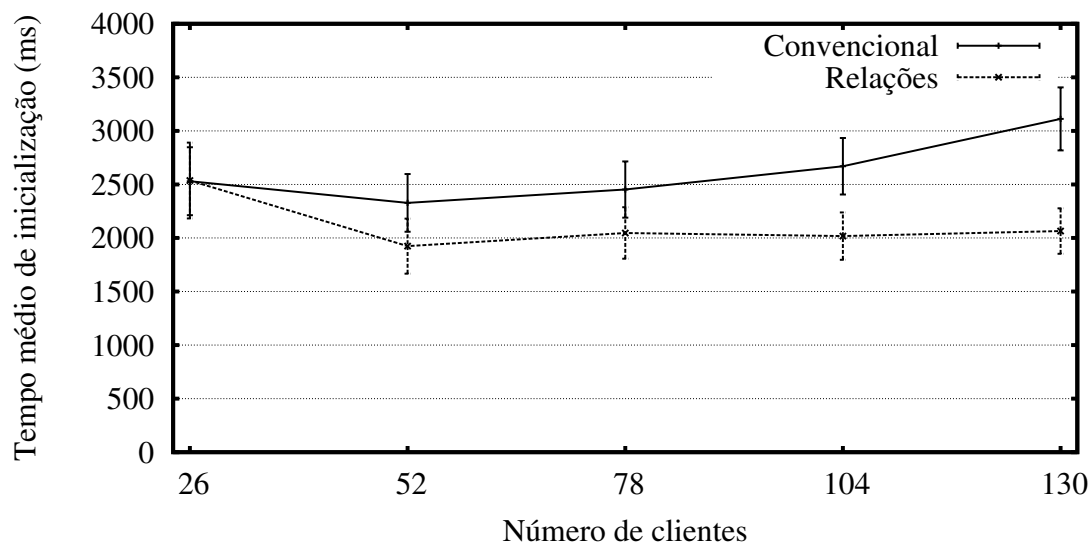


Figura 5.6: Impacto de variar o número de clientes no tempo de inicialização

como mostrado na Figura 5.7. Conforme explicado anteriormente, devido à distribuição Zipf utilizada, o número de travamentos manteve-se próximo a uma constante independentemente do número de conteúdos; houve, entretanto, uma diferença entre o mecanismo convencional e o mecanismo de relacionamento de objetos. Tal resultado, além de confirmar que há uma consequente redução no número de travamentos devido ao uso de relacionamento de objetos, conforme constatado no experimento variando-se o número de clientes, também demonstra que essa redução é válida para diferentes quantidades de conteúdos distribuídos na rede.

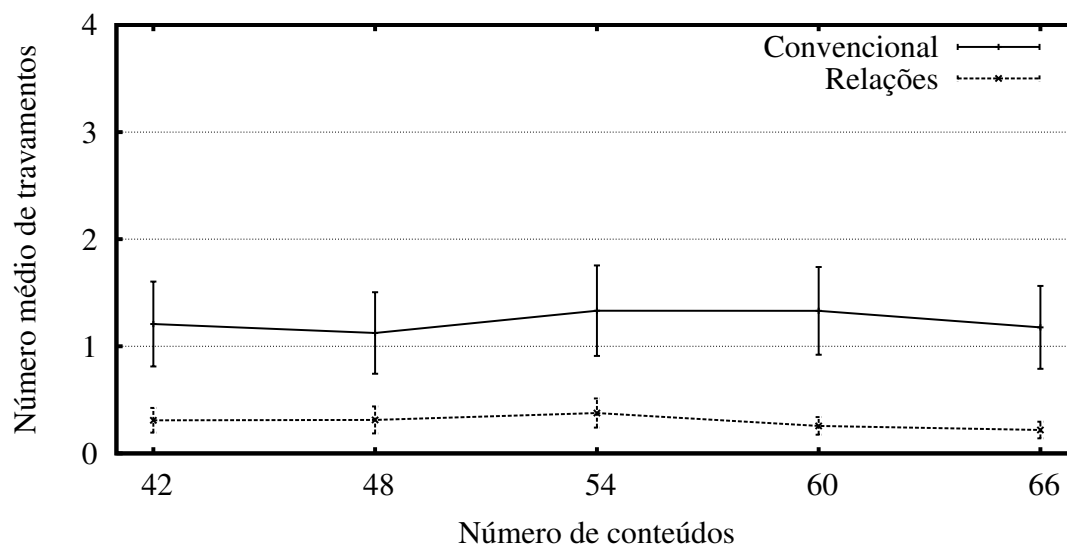


Figura 5.7: Impacto de variar o número de conteúdos no número de travamentos

Por fim, avalia-se o tempo médio de inicialização do vídeo para diferentes quantidades de conteúdos. O gráfico com os resultados obtidos é apresentado na Figura 5.8. Percebe-se que o mecanismo de relacionamento de objetos reduz o tempo de inicialização do *streaming* para quase todos os casos. Para 48 conteúdos, entretanto, apesar da

curva do mecanismo de relacionamento de objetos estar abaixo da curva convencional no gráfico, os intervalos de confiança interceptam-se; não é estatisticamente possível, por conseguinte, afirmar que o mecanismo proposto reduz significativamente o tempo de inicialização nesse caso. Mais execuções seriam necessárias para obter-se um resultado mais preciso; no entanto, dado o tempo disponível para execução deste trabalho, não foi possível aumentar o número de experimentos realizados.

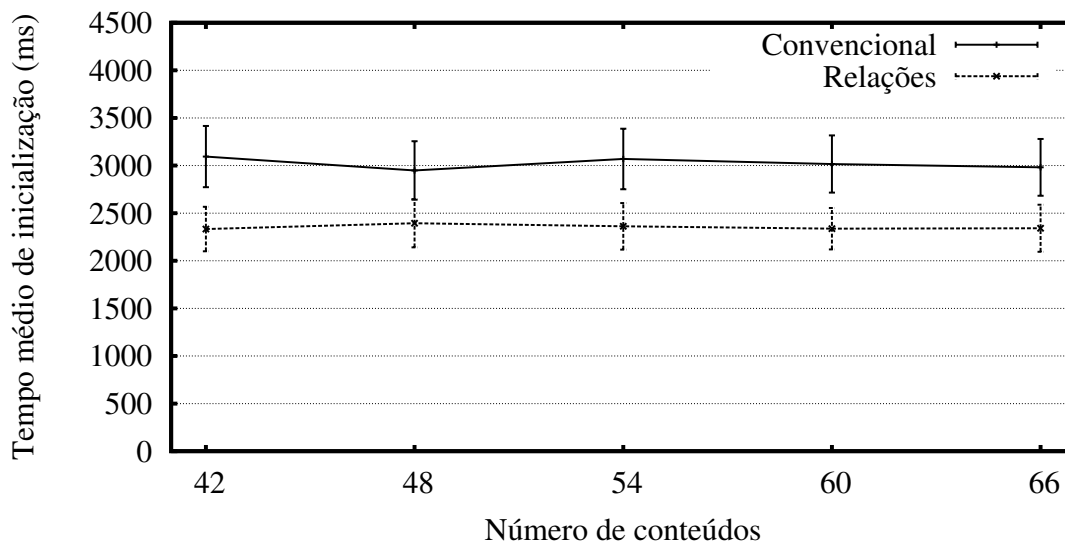


Figura 5.8: Impacto de variar o número de conteúdos no tempo de inicialização

5.3 Discussão dos resultados

Os resultados apresentados anteriormente permitem responder às três questões introduzidas no Capítulo 4. Considerando a primeira questão (Q1), observou-se que modelar objetos através de relações pode reduzir o tráfego da rede em até 34,4%. Quanto ao impacto do mecanismo proposto nos componentes principais de CCN (Q2), observou-se que há uma melhoria na eficiência do sistema de *caching*, com um aumento de até 48,9% na taxa de acerto da cache. Tal aumento possui um impacto, também, na memória necessária para manter as tabelas PIT. Por fim, considerando o impacto do mecanismo proposto na qualidade de experiência percebida pelos usuários (Q3), observou-se uma redução de até 86,7% no número de travamentos e uma redução de até 33,6% no tempo de inicialização do *streaming*.

É importante considerar que, dadas as características de redes orientadas a conteúdo, os resultados obtidos são influenciados pela popularidade e pelo tamanho dos objetos distribuídos. Para que aplicações possam tirar o maior proveito possível do relacionamento de objetos, recomenda-se, por conseguinte, que tais fatores sejam levadas em consideração durante a modelagem de conteúdos através de múltiplos objetos. As aplicações têm, então, através da escolha de como modelar conteúdos, um impacto expressivo nas melhorias trazidas ao desempenho da rede.

Outro ponto a ser observado é a dependência dos resultados obtidos da cache presente na rede. Sem cache, acredita-se que o tráfego necessário para atender às demandas dos clientes seja constante, com ou sem relações, exceto por alguns pequenos benefícios da agregação de requisições feita pela PIT. Consequentemente, o tempo de inicialização

e número de travamentos esperado através da utilização do relacionamento de objetos tornam-se praticamente os mesmos obtidos pelo mecanismo convencional. No entanto, é importante considerar que o mecanismo de cache é um dos principais benefícios de CCN; logo, desabilitar tal mecanismo iria impactar negativamente o desempenho da arquitetura independentemente do uso do mecanismo proposto. Assim, ao empregar o mecanismo de relações, deve-se considerar as configurações de *caching*.

Por fim, acredita-se que as informações de relacionamento distribuídas pelas aplicações têm potencial para melhorar o desempenho dos principais componentes da arquitetura CCN. Um exemplo consiste no uso de relações para ajudar as *Content Stores* a selecionar quais conteúdos manter em cache. Para implementar tal possibilidade, deve-se integrar o mecanismo proposto ao CCNx. No entanto, ainda é necessário estudar e planejar cuidadosamente tal integração antes de implementá-la, a fim de analisar sua viabilidade e de não comprometer a compatibilidade com aplicações que não utilizam o mecanismo de relacionamento de objetos.

6 CONCLUSÃO

Com o intuito de tornar a disseminação de conteúdo em redes de computadores ainda mais eficiente, surgiu um interesse na comunidade científica para a criação da Internet do Futuro. Dentre as diferentes propostas, encontra-se a arquitetura CCN. Em tal arquitetura, desvincula-se da necessidade de conhecer a localização física de um conteúdo para obtê-lo: conteúdos são distribuídos como objetos nomeados e roteados através de mecanismos envolvendo seu nomes. Com tal abordagem, melhora-se o desempenho da rede e, conseqüentemente, das aplicações que dela dependem.

Neste trabalho, estudou-se possibilidades para aumentar ainda mais a eficiência da arquitetura CCN, através da análise de como conteúdos são distribuídos na rede. Propôs-se um mecanismo para a modelagem de conteúdos em diferentes objetos, os quais são ligados através de relações. Após projetar tal mecanismo, implementou-se, com base no projeto CCNx, as funções necessárias para permitir sua avaliação e, também, uma aplicação de *streaming* que o utiliza. Por fim, visando quantificar os benefícios de utilizar o relacionamento de objetos em tal arquitetura, cenários realísticos foram determinados e experimentos foram realizados.

Quanto ao desempenho da rede, os resultados obtidos através da avaliação experimental por emulação mostram que o uso do mecanismo proposto pode reduzir o tráfego na rede em até 34,4%. Tal fato está relacionado com o aperfeiçoamento na eficiência do mecanismo de cache, que apresentou um aumento de 48,9% em sua taxa de acerto: quando um objeto é encontrado em uma cache próxima, o caminho percorrido por ele até o cliente é menor, reduzindo então o tráfego total. Quanto à qualidade de experiência dos usuários nos cenários de distribuição de vídeo-sob-demanda avaliados, observa-se uma redução de até 33,6% no tempo que um cliente deve esperar antes de poder iniciar a reprodução do vídeo desejado e uma redução de até 86,7% no número de travamentos. Dessa maneira, os resultados confirmam o potencial que o relacionamento de objetos possui tanto para aumentar a eficiência de redes orientadas a conteúdo quanto para melhorar a qualidade de experiência dos usuários.

Como trabalhos futuros, planeja-se estudar e analisar o impacto do abandono de vídeos na rede, incluindo tal possibilidade nas avaliações realizadas com a aplicação de vídeo-sob-demanda. Planeja-se, também, executar os experimentos através do uso de simulação, para analisar os resultados obtidos sem as interferências inerentes à emulação, tal como a disputa por recursos físicos. Ainda, planeja-se estudar a possibilidade de estender às caches a compreensão da existência de relacionamentos de objetos, potencialmente permitindo uma alocação mais inteligente dos recursos. Por fim, espera-se que os resultados obtidos neste trabalho auxiliem na proposta e implementação de futuras melhorias nas arquiteturas ICN existentes, especialmente CCN.

REFERÊNCIAS

AHLGREN, B.; DANNEWITZ, C.; IMBRENDA, C.; KUTSCHER, D.; OHLMAN, B. A Survey of Information-Centric Networking. **IEEE Communications Magazine**, New York, NY, USA, v.50, n.7, p.26–36, 2012.

BARI, F.; CHOWDHURY, S. R.; AHMED, R.; BOUTABA, R.; MATHIEU, B. A Survey of Naming and Routing in Information Centric Networks. **IEEE Communications Magazine**, [S.l.], p.1–9, 2012.

CAUMONT, A. **12 trends shaping digital news**. Disponível em <<http://www.pewresearch.org/fact-tank/2013/10/16/12-trends-shaping-digital-news/>>. Acesso em: 30 out. 2013.

FINAMORE, A.; MELLIA, M.; MUNAFÒ, M. M.; TORRES, R.; RAO, S. G. YouTube everywhere: impact of device and infrastructure synergies on user experience. In: IMC 2011: ACM SIGCOMM CONFERENCE ON INTERNET MEASUREMENT CONFERENCE, BERLIN. **Proceedings...** [S.l.: s.n.], 2011. p.345–360.

GHODSI, A.; KOPONEN, T.; RAJAHALME, J.; SAROLAHTI, P.; SHENKER, S. Naming in content-oriented architectures. In: ICN '11: ACM SIGCOMM WORKSHOP ON INFORMATION-CENTRIC NETWORKING, Orlando. **Proceedings...** ACM, 2011. p.1–6.

GHODSI, A.; KOPONEN, T.; RAGHAVAN, B.; SHENKER, S.; SINGLA, A.; WILCOX, J. Information-Centric Networking: seeing the forest for the trees. In: HOTNETS '11: ACM WORKSHOP ON HOT TOPICS IN NETWORKS, Cambridge. **Proceedings...** [S.l.: s.n.], 2011. p.1:1–1:6.

JACOBSON, V.; SMETTERS, D. K.; THORNTON, J. D.; PLASS, M.; BRIGGS, N.; BRAYNARD, R. Networking Named Content. **Communications of the ACM**, [S.l.], v.55, n.1, p.117–124, 2012.

KAUFFMANN, B.; PELTIER, J.-f.; TRUONG, P. **Final NetInf Architecture**. Disponível em <<http://www.sail-project.eu/wp-content/uploads/2013/01/SAIL-DB3-v1.1-final-public.pdf>>.

KOPONEN, T.; CHAWLA, M.; CHUN, B.-G.; ERMOLINSKIY, A.; KIM, K. H.; SHENKER, S.; STOICA, I. A data-oriented (and beyond) network architecture. In: SIGCOMM '07: ACM SIGCOMM CONFERENCE, Kyoto. **Proceedings...** ACM, 2007. p.181–192.

KUROSE, J. F.; ROSS, K. W. **Computer Networking: a top-down approach**. 6th.ed. USA: Addison-Wesley Publishing Company, 2012.

PARC. **CCNx Repository Protocols**. Disponível em <<http://www.ccnx.org/releases/latest/doc/technical/RepoProtocol.html>>. Acesso em: 11 nov. 2013.

PARC. **CCNx Project, Palo Alto Research Center**. Disponível em <<http://www.ccnx.org>>. Acesso em: 04 nov. 2013.

PARC. **Index of /releases/ccnx-0.8.1**. Disponível em <<http://www.ccnx.org/releases/ccnx-0.8.1/>>. Acesso em: 04 nov. 2013.

PERINO, D.; VARVELLO, M.; PUTTASWAMY, K. P. N. ICN-RE: redundancy elimination for information-centric networking. In: ICN '12: ACM SIGCOMM INFORMATION-CENTRIC NETWORKING WORKSHOP, Helsinki. **Proceedings...** ACM Press, 2012. p.91–96.

RIBEIRO, I. C. G.; GUIMARÃES, F. Q.; KAZIENKO, J. F.; ROCHA, A. A. d. A.; VELLOSO, P. B.; MORAES, I. M.; ALBUQUERQUE, C. V. N. Segurança em Redes Centradas em Conteúdo: vulnerabilidades, ataques e contramedidas. In: Minicursos do XII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, Curitiba, PR, Brasil. **Proceedings...** SBC SBSEG, 2012.

RNP. **rede Ipê**. Disponível em <<http://www.rnp.br/ipe>>. Acesso em: 19 ago. 2013.

RNP. **Medições da rede Ipê**. Disponível em <<http://www.rnp.br/ceo/2013.html>>. Acesso em: 03 set. 2013.

TROSSEN, D.; PARISIS, G. Designing and Realizing an Information-Centric Internet. **IEEE Communications Magazine**, [S.l.], v.50, n.7, p.60–67, July 2012.

VAKALI, A.; PALLIS, G. Content Delivery Networks: status and trends. **IEEE Internet Computing**, [S.l.], v.7, n.6, p.68–74, 2003.

WANG, J. M.; ZHANG, J.; BENSOU, B. Intra-AS cooperative caching for content-centric networks. In: ICN '13: ACM SIGCOMM INFORMATION-CENTRIC NETWORKING WORKSHOP, Hong Kong. **Proceedings...** ACM Press, 2013. p.61–66.

XYLOMENOS, G.; VERVERIDIS, C. N.; SIRIS, V. A.; FOTIOU, N.; TSILOPOULOS, C.; VASILAKOS, X.; KATSAROS, K. V.; POLYZOS, G. C. A Survey of Information-Centric Networking Research. **IEEE Communications Surveys Tutorials**, [S.l.], v.PP, n.99, p.1–26, July 2013.

ZHANG, L.; ESTRIN, D.; BURKE, J.; JACOBSON, V.; THORNTON, J. D.; SMETTERS, D. K.; ZHANG, B.; TSUDIK, G.; CLAFFY, K. C.; KRIOUKOV, D.; MASSEY, D.; PAPADOPOULOS, C.; ABDELZAHER, T.; WANG, L.; CROWLEY, P.; YEH, E. **Named Data Networking Project**. [S.l.: s.n.], 2010.

ANEXO A: TRABALHO DE GRADUAÇÃO I

Mutabilidade de Objetos em Redes Orientadas a Conteúdo

Bruna R. Fiorentin¹, Marinho P. Barcellos¹

¹Instituto de Informática - Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500 - Porto Alegre, RS - Brasil

{bflorentin,marinho}@inf.ufrgs.br

Abstract. *Applications based on content distribution are nowadays responsible for most of the Internet's data traffic. Recent efforts for optimizing the efficiency of these applications led to the proposal of a new Internet paradigm, known as Information-Centric Networking. In this paradigm, data is represented as objects of immutable content, which are identified by an arbitrary naming scheme. Although immutable content allows the efficient distribution of static data, gains offered by Information-Centric Networking architectures can be lost in the context of applications that require frequent update of data. This paper presents a study about Information-Centric Networks and how they can be modified in order to support mutable content. Also, this study explores semantic and implementation aspects related to object mutability. This study represents the theoretical basis for the Graduation Work 2.*

Resumo. *Aplicações de distribuição de conteúdo são atualmente responsáveis pela maior parte do tráfego de dados na Internet. Os recentes esforços para otimizar o desempenho dessas aplicações levaram à proposta de um novo paradigma para a Internet, denominado Redes Orientadas a Conteúdo. Nesse paradigma, conteúdos são representados por objetos de dados imutáveis, os quais são unicamente identificados por um esquema de nomeação arbitrário. Embora objetos imutáveis permitam a distribuição eficiente de conteúdos estáticos, os ganhos oferecidos por arquiteturas de Redes Orientadas a Conteúdo podem ser perdidos no contexto de aplicações que requerem atualização frequente do conteúdo. Este artigo apresenta um estudo sobre Redes Orientadas a Conteúdo e como estas podem ser modificadas a fim de suportar a mutabilidade de objetos, analisando aspectos de semântica e implementação da mesma. O estudo apresentado servirá de embasamento teórico para a realização do Trabalho de Graduação 2.*

1. Introdução

Atualmente, um grande número de aplicações da Internet troca conteúdo mutável. Nesse tipo de conteúdo, a informação pode ser atualizada com o tempo, mas continua fundamentalmente a mesma. Exemplos incluem redes sociais, *feed* de notícias e mercado de ações. Ademais, a evolução da Internet atual instiga a pensar em futuras aplicações, tal como edição colaborativa de conteúdo multimídia.

Aplicações que utilizam objetos mutáveis requerem mecanismos para controlar interações entre entidades que acessam e atualizam informações. Entre esses mecanismos, versionamento de objetos destaca-se, pois fornece meios de controlar replicação e consistência de atualização. Uma versão de um objeto mutável é uma informação que permite

a identificação de que tal objeto foi atualizado ou de que teve conteúdo agregado. Dessa maneira, possibilita-se que clientes requisitem por versões específicas de um objeto, ou, por exemplo, pela mais nova versão existente. Clientes podem também, possivelmente, indicar objetos desatualizados.

O desempenho desses mecanismos está relacionado com os protocolos e serviços de redes subjacentes. Por conseguinte, tais mecanismos, quando utilizados sob Redes Orientadas a Conteúdo (*Information-Centric Networking* - ICN) [Ahlgren et al. 2012], podem apresentar um desempenho diferente do apresentado na Internet atual. Dados o avanço na pesquisa em ICN e a possível implantação dessa tecnologia na Internet, deve-se, portanto, realizar um estudo do comportamento dos mesmos sob o paradigma ICN.

ICN distribui os dados na forma de objetos imutáveis, identificados de maneira única por um esquema de nome definido pela arquitetura utilizada. Essa abstração é adequada para conteúdos que se tornam estáticos após sua publicação, como áudio ou vídeo sob demanda. No entanto, há desvantagens quando usada para distribuir conteúdo versionado, como discutido a seguir.

Nas propostas de arquitetura ICN atuais, um possível método para implementar versionamento é explorar as capacidades do esquema de nomes, ou seja, incluir a versão no nome do objeto; dessa maneira alterações na arquitetura existente podem ser minimizadas. Intuitivamente, se novas versões de um objeto são frequentemente geradas, essas podem acarretar a sobrecarga das estruturas usadas para roteamento, encaminhamento e gerenciamento de cache. De fato, essas estruturas são conhecidas por apresentar problemas de escalabilidade em arquiteturas ICN, possivelmente conduzindo a uma degradação do desempenho da rede [Bari et al. 2012].

Nesse contexto, o objetivo deste trabalho é investigar o suporte a versionamento de objetos em ICN. Dentre as propostas de arquitetura ICN disponíveis, será utilizada como base do estudo a arquitetura denominada *Content-Centric Network* (CCN) [Jacobson et al. 2012] [Zhang et al. 2010], por se tratar daquela com o protótipo mais estável e avaliado pela comunidade científica [Ribeiro et al. 2012]. Neste primeiro artigo, o trabalho consiste no estudo e entendimento do problema e na procura de uma alternativa para otimizar o desempenho de aplicações baseadas em objetos mutáveis. No Trabalho de Graduação 2, serão investigadas, implementadas e avaliadas alterações nos mecanismos da arquitetura CCN, visando tornar possível a distribuição eficiente de objetos mutáveis.

O restante deste artigo está organizado como segue. A Seção 2 introduz o paradigma ICN e apresenta 4 aspectos gerais de arquitetura: nomeação, roteamento, *caching* e segurança. Na Seção 3, a arquitetura CCN utilizada nesse trabalho é detalhada com base nos tópicos apresentados na Seção 2. A Seção 4 discute diferentes abordagens para a semântica e implementação da mutabilidade de objetos. Na Seção 5, a proposta do Trabalho de Graduação 2 é apresentada, seguida da metodologia adotada e do cronograma de atividades a serem realizadas. Por fim, a Seção 6 conclui o documento.

2. Redes Orientadas a Conteúdo

A Internet foi criada com o intuito de possibilitar comunicação fim-a-fim entre dois computadores – também denominados *hosts*. Dada a natureza estática dos *hosts* na época da concepção da Internet, a arquitetura tornou-se fortemente dependente da localização física das máquinas. O protocolo IP, principal elemento da camada de rede da Internet

atual, utiliza o endereço de rede de ambas as máquinas envolvidas em uma comunicação para rotear os dados entre elas [Kurose and Ross 2012]. No entanto, o uso da Internet vem se concentrando cada vez mais na distribuição de conteúdo. Segundo apontado por [Ahlgren et al. 2012], o tráfego IP global se aproximará de 80 exabytes/mês em 2015, sendo quadruplicado com base no tráfego de 2010; desse tráfego, aproximadamente 90% será devido à distribuição de vídeos. Consequentemente, faz-se necessário um sistema de Internet bem mais escalável e eficiente. Como alternativa de um novo paradigma para a Internet, surgiu o conceito de ICN, que defende uma rede baseada em conteúdo ao invés de localização [Bari et al. 2012].

Atualmente, a comunidade de ICN propõe 4 arquiteturas principais: *Data-Oriented Network Architecture* (DONA) [Koponen et al. 2007], *Content-Centric Networking, Publish-Subscribe Internet Routing Paradigm* (PSIRP) [Trossen and Parisis 2012] e *Network of Information* (NetInf) [Kauffmann et al. 2013]. A presente seção introduz os componentes comuns às arquiteturas ICN de maneira genérica e discute as vantagens desse novo paradigma para a Internet. A seção subsequente concentra-se na arquitetura CCN, que é a escolhida para o estudo da mutabilidade de objetos nesse trabalho.

2.1. Componentes da arquitetura ICN

Nomeação. Objetos Nomeados de Dados (NDO, do inglês *Named Data Objects*) são os principais componentes de uma Rede Orientada a Conteúdo. NDOs fornecem a abstração necessária para que um conteúdo seja distribuído na Internet sem necessidade de conhecimento da localização física na qual ele se encontra. Exemplos de NDOs são páginas web, imagens, vídeos, músicas, entre outros. Cada NDO deve ser inteiramente identificado pelo seu nome, ou seja, se dois servidores distintos armazenarem o mesmo NDO, o conteúdo recebido por um cliente que o requisitar deve ser o mesmo independente do servidor que o fornecer. A nomeação de objetos em ICN varia dependendo da arquitetura utilizada. Há dois esquemas principais de nomeação: plana utilizando *namespaces* e hierárquica. No contexto deste documento, o segundo esquema é mais importante, uma vez que é o utilizado pela arquitetura CCN. A nomeação hierárquica será, portanto, detalhada na Seção 3.

Roteamento. O roteamento em ICN pode ser genericamente resumido em duas fases. Primeiramente, a requisição de um conteúdo é encaminhada através de uma estratégia de roteamento baseada no esquema de nome da arquitetura. Após, ao alcançar um nodo que possua tal conteúdo, o mesmo é encaminhado de volta ao requisitante.

Com um roteamento baseado em nomes, ganha-se no quesito mobilidade em comparação com a Internet atual. O cenário de um cliente movendo-se por diferentes redes exemplifica esse ganho. Diferentemente do que ocorre atualmente, não é necessário que o cliente mantenha uma conexão fim-a-fim com o servidor original para obter um conteúdo, recalculando as melhores rotas; basta que ele continue requisitando os objetos desejados pelo nome, que é único no contexto em que está inserido. Um outro servidor - provavelmente mais próximo ao cliente - pode responder à requisição dos dados.

Caching. Um dos princípios de ICN é que todos os nodos da rede potencialmente possuem uma cache. Consequentemente, um NDO requisitado pode vir a ser encontrado em diversos lugares da rede. Dessa maneira, a localização de um objeto não está restrita à localização do nodo que o publicou ou a nodos destinados a replicar este objeto. As caches de cada nodo são populadas conforme NDOs passam pelo mesmo em seu caminho

ao destino. Duas vantagens da existência de um mecanismo de *caching* são: (i) o tráfego de dados distribui-se melhor pela rede; e (ii) proporciona-se uma melhor tolerância à interrupção através da existência de caminhos alternativos no caso de um nodo ou uma interface da rede falhar.

Segurança. A segurança na Internet normalmente é avaliada em termos de 4 componentes: integridade, confiabilidade, proveniência e disponibilidade. Espera-se que a rede garanta que os dados não sejam alterados indevidamente (integridade), nem lidos por outros (confiabilidade) ou corrompidos (disponibilidade). Além disso, deve certificar-se de que a origem dos dados é apropriada (proveniência) [Ghodsi et al. 2011b].

Em ICN, a fim de garantir esses aspectos de segurança, é necessário certificar os NDOs diretamente, visto que os mesmos podem ser encontrados em diversas caches da rede. Proveniência, integridade e disponibilidade podem ser garantidas através da assinatura dos NDOs pelo publicador do objeto. Dessa maneira, os clientes podem certificar-se da validade do conteúdo simplesmente verificando a assinatura [Ghodsi et al. 2011a].

Uma possível abordagem para assinar o conteúdo é a utilização de nomes auto-certificados, os quais são construídos através de estruturas criptográficas que permitem uma forte correspondência entre os conteúdos e seus nomes [Ribeiro et al. 2012]. Esses nomes podem ser verificados por *hash* ou por chave. No primeiro caso, o nome é gerado pela *hash* do conteúdo; o cliente, quando recebe um NDO, confere a integridade do conteúdo realizando a *hash* do mesmo e comparando com a *hash* informada pelo publicador. No segundo caso, os nomes são verificados pela *hash* da chave pública do publicador. Outra abordagem, que é a abordagem utilizada por CCN, é a utilização de uma assinatura do publicador, a qual é efetuada sob o mapeamento entre o nome e o conteúdo; um NDO, portanto, possui 3 elementos: o nome, o conteúdo e a assinatura. O cliente deve então obter a chave pública do publicador a fim de verificar tal assinatura. Através dessa verificação, o cliente pode certificar-se tanto da integridade do conteúdo quanto da proveniência do mesmo [Ribeiro et al. 2012].

3. Content-Centric Networking

A arquitetura CCN faz parte das abordagens de arquiteturas ICN sendo ativamente estudadas [Ahlgren et al. 2012]. Uma implementação de tal arquitetura é o projeto CCNx [PARC 2013b], desenvolvido no Centro de Pesquisas Palo Alto (PARC). Dada sua importância, esse projeto será utilizado como base na realização desse Trabalho de Graduação.

Nesta seção, os componentes arquiteturais do paradigma ICN apresentados previamente são retomados. Apresenta-se, para cada componente, a escolha de implementação utilizada no CCNx.

Nomeação. Na arquitetura CCNx, usa-se um esquema de nomeação hierárquico, assim como o endereço IP, sendo o caractere ‘/’ o delimitador dos componentes do nome. No entanto, contrariamente ao endereço IP, cada componente do nome pode ter um tamanho arbitrário. Os nomes assemelham-se a URIs, sendo legíveis para seres humanos; pelos roteadores CCNx, entretanto, os nomes não são interpretados, apenas comparados.

Dentre as informações contidas em um componente do nome de um objeto, há potencialmente uma referência ao publicador; é importante ressaltar, contudo, que essa referência identifica o publicador e não seu endereço físico, seguindo os princípios de uma arquitetura ICN. Informações adicionais, que podem estar presentes em outros com-

ponentes do nome, incluem informações sobre a versão do objeto ou número de segmento. Como os roteadores em CCNx não interpretam nomes, o significado associados a esses é de responsabilidade das aplicações ou instituições que publicam os objetos, podendo seguir convenções globais.

Um nome em CCNx pode identificar diretamente um objeto ou, de maneira alternativa, identificar um nodo de uma árvore, o qual aponta para múltiplos objetos. A documentação do projeto CCNx argumenta que um nome identificando uma coleção de objetos, sendo esse nome um prefixo do nome de cada objeto dessa coleção, comporta-se de maneira análoga a um endereço de rede IPv4, o qual também pode ser um prefixo do endereço IP dos *hosts* pertencentes àquela rede.

Roteamento e Caching. Na arquitetura CCN, há dois tipos principais de mensagens: *Interest* e *Data*. Quando um nodo da rede está interessado em um conteúdo, ele envia uma mensagem do tipo *Interest*. Tal mensagem é composta pelo nome do objeto a ser requisitado, ou por um prefixo e um conjunto restrições a serem aplicadas para indicar quais dos conteúdos apontados por esse prefixo são desejáveis. Apenas o nome (ou prefixo) do objeto desejado é obrigatório em uma mensagem *Interest*. Dentre as restrições, pode-se definir um número mínimo ou máximo de componentes do nome, limitar de onde a resposta virá, aconselhar a seleção do que enviar no caso em que múltiplos objetos correspondam ao requisitado, entre outros.

A mensagem de retorno, contendo o conteúdo requisitado pelo nodo, é uma mensagem do tipo *Data*, também denominada *Content* ou *Content Object*. Além do conteúdo em si, o qual pode ser opcionalmente cifrado, uma mensagem do tipo *Data* contém o nome do conteúdo, uma assinatura criptográfica e a identificação do publicador. Todas mensagens do tipo *Data* devem obrigatoriamente conter uma assinatura válida.

A comunicação em CCNx é controlada pelo receptor. O receptor, interessado em um conteúdo, envia uma mensagem do tipo *Interest* às *faces* disponíveis, podendo utilizar broadcast ou multicast. Em CCN, uma *face* corresponde ao conceito de interface em redes de computadores. Qualquer nodo que possua o *Named Data Object* requisitado em sua cache, denominada *Content Store* (CS), e que satisfaça as condições impostas na mensagem de *Interest*, pode responder à requisição. A fim de evitar que dois nodos diferentes respondam a uma única requisição, mecanismos de supressão de mensagens do tipo *Interest* devem ser implementados pelos nodos. A versão atual do CCNx não especifica regras para gerenciar essa supressão, apenas recomenda que os mecanismos utilizados incluam tempos de resposta randômicos e que identifiquem quando um nodo já respondeu à requisição.

As mensagens do tipo *Interest* enviadas pelas aplicações são propagadas pela rede através da análise, utilizando uma lógica de *longest prefix match*, da tabela *Forwarding Information Base* (FIB), a qual está presente em todos os nodos que utilizam CCN. A FIB contém um mapeamento entre os *Interests* de entrada e as *faces* de saída adequadas.

Cada nodo que recebe um pacote *Interest*, antes de tomar a decisão sobre repassar o pacote, realiza uma busca na *Content Store* (CS). A CS deve implementar uma *flag*, conhecida como *Staleness Bit*, que é utilizada para indicar quando um objeto em cache não deve ser enviado como resposta a um *Interest*, a menos que seja especificado na requisição que tais conteúdos são aceitos. Os objetos marcados com o *Staleness Bit* são os de maior prioridade em políticas de liberação de espaço da cache. Tal bit pode ser

atualizado explicitamente ou através de um *timer*.

Cada nodo da rede possui ainda uma outra tabela, denominada *Pending Interest Table* (PIT), a qual é responsável pelo roteamento de um NDO até o cliente que o requisitou, seguindo o caminho reverso do *Interest*. Se o NDO desejado não for encontrado na CS, uma nova entrada é adicionada à PIT contendo a *face* pela qual o *Interest* foi encaminhado. Quando a requisição for satisfeita por um nodo, a mensagem do tipo *Data* retorna à aplicação requisitante através da análise de cada PIT no caminho ao receptor. Caso uma mensagem do tipo *Interest* chegue em um nodo que já apresenta uma entrada para aquele mesmo conteúdo na PIT, conclui-se que tal conteúdo já foi requisitado, mas que ainda não chegou ao destino. Nesse caso, a tabela PIT é atualizada adicionando-se a *face* pela qual o *Interest* chegou e a mensagem é descartada. Esse comportamento otimiza o uso da rede, visto que duas ou mais requisições pelo mesmo conteúdo, caso passem por um ponto de intersecção, não são distribuídas múltiplas vezes através da rede, como acontece na Internet atual.

Segurança. Como já citado anteriormente, todas as mensagens do tipo *Data* em CCNx devem obrigatoriamente conter uma assinatura válida, a qual será verificada pelas aplicações que receberem os NDOs. Para verificar a assinatura, pode-se precisar de *public keys*, que não estão incluídas na mensagem contendo os dados; no entanto, apesar de não haver um mecanismo destinado à distribuição de chaves, elas podem ser distribuídas como quaisquer outros dados em CCNx. Os autores do protocolo CCNx afirmam que o protocolo foi projetado para garantir entrega confiável de dados, sem limitar as decisões sobre distribuição de chaves e gerenciamento de segurança. Dessa maneira, a assinatura dos conteúdos é utilizada para confirmar que um objeto não foi corrompido em seu caminho até o destinatário. Visando maior segurança na rede, estimula-se o uso de políticas com a finalidade de determinar se um determinado publicador é confiável para um determinado tipo de conteúdo, as quais não estão especificadas diretamente no protocolo CCNx.

4. Mutabilidade de Objetos

No paradigma ICN, os objetos e seu conteúdo são os elementos principais da Internet. Nesse contexto, estuda-se a mutabilidade de objetos, a qual refere-se à possibilidade de gerar novas versões de objetos tendo como base o conteúdo de outros já existentes. A relação entre as versões e os objetos originais é mantida através de uma relação de mutabilidade, que pode ser definida como um apontador, presente em cada versão, que referencia os objetos originais.

Nesta seção, apresenta-se primeiramente os mecanismos existentes no CCNx que podem ser relacionados à mutabilidade de objetos. Em seguida, apresenta-se diferentes abordagens para a semântica da mutabilidade de objetos. Por fim, discute-se a implementação da mutabilidade de objetos na arquitetura CCNx.

4.1. Suporte à Mutabilidade no CCNx

A versão atual do protótipo do CCNx [PARC 2013a] não possui mecanismos que tratem especificamente a mutabilidade de objetos. Entretanto, o CCNx possui três mecanismos que são de interesse para a implementação de relações de mutabilidade: versionamento, metadados e requisições por prefixos. O versionamento do CCNx permite que aplicações atualizem uma publicação sem provocar inconsistências nas *Content Stores*. A versão é

um componente adicional, presente no final do nome do objeto, que permite às CSs distinguir diferentes versões de uma publicação. Tal mecanismo tem como principal objetivo evitar o envio de *chunks* pertencentes a versões distintas de um objeto em resposta a um *Interest*. Outras funcionalidades, como a limpeza de caches em função da análise de versões do objeto, não são suportadas pela implementação. O mecanismo de metadados do CCNx, por sua vez, permite que informações adicionais sobre o conteúdo de um objeto sejam distribuídas através de um cabeçalho. Tal informação é acessada através do nome do objeto concatenado de uma palavra-chave reservada que sinaliza o interesse pelos metadados. Por fim, o mecanismo de interesse por prefixos permite que um conjunto de objetos seja recuperado através de apenas uma requisição, a qual aponta para um prefixo comum no nome destes objetos. Tal mecanismo permite, por exemplo, que uma árvore de versões de um objeto seja obtida com um número reduzido de mensagens do tipo *Interest*. Porém, para que tal seja possível, é necessária a imposição de regras para o modo como objetos relacionados são nomeados.

As funcionalidades de ambos mecanismos apresentados podem ser utilizadas como base para a implementação das relações de mutabilidade no CCNx e, portanto, serão avaliadas mais detalhadamente durante o desenvolvimento do trabalho. Tais funcionalidades por si só, entretanto, limitam as possibilidades de semântica de mutabilidade de objetos, as quais são apresentadas na próxima seção.

4.2. Semântica da Mutabilidade

Uma versão corresponde, no contexto deste trabalho, a uma tupla $\langle o, R \rangle$ que contém o objeto o e um conjunto de relações de mutabilidade R estabelecidas com outros objetos. Na literatura científica, não existem trabalhos que explorem a questão da mutabilidade de objetos em ICN. Neste trabalho, identificamos duas possibilidades principais quanto à semântica de mutabilidade, conforme discutido a seguir.

A primeira semântica possível consiste em entender versões como representando um objeto a medida que seu conteúdo sofre alterações. Por exemplo, um objeto que representa um *log*, e cada nova versão traz um conjunto de eventos adicionados ao final, ou um vídeo que é processado através de uma sequência de filtros, sendo que cada versão é usada como entrada no passo seguinte. Outro exemplo possível seria um objeto que é periodicamente modificado, sendo as versões anteriores consideradas progressivamente obsoletas. Há três questões relacionadas importantes, como visto a seguir. Primeiro, se o objeto que corresponde à versão anterior é substituído pela nova versão, ou se um novo objeto é criado e existe uma forma de representar a relação entre os objetos (por exemplo, via nome). Segundo, quem pode criar uma nova versão de um objeto: apenas seu publicador ou outro usuário qualquer. Terceiro e último, considerando que um objeto vai sendo modificado e que essas alterações precisam ser vistas pelo resto da rede, que tipo de consistência é desejável (e viável) quando versões de um mesmo objeto são geradas e publicadas.

A segunda semântica generaliza o conceito de relacionamento ou dependência entre objetos. Ou seja, as relações não são restritas a uma sequência de versões de um objeto. A semântica consiste em especificar um ou mais objetos que servem de base para uma nova versão, sem contudo modificar o conteúdo dos mesmos. Por exemplo, considere três objetos, que representam um vídeo (imagem), seu canal de áudio, e legenda em uma determinada língua. Os mesmos podem ser "combinados" em um objeto único, represen-

tando um filme com uma dada escolha de áudio e legendas. Outro exemplo possível é a criação de uma discografia com base na combinação de um conjunto de álbuns, ou de uma filмотeca com base em um conjunto de obras cinematográficas. A nova versão do objeto, em contraste à abordagem anterior, não tem como objetivo substituir os objetos que lhe deram origem, mas sim acrescentar mais possibilidades de conteúdo à rede. Neste caso, a relação entre objetos deve expressar esta combinação, possivelmente formando uma "hierarquia de objetos". Questões relacionadas importantes são como expressar a relação de combinação entre objetos e como lidar com múltiplos níveis de combinação (um objeto já combinado é usado como base para criação de um novo).

Entre as duas possibilidades de semântica descritas, focar-se-á na segunda pelo fato de ser mais genérica, permitindo que aplicações utilizem a mutabilidade em contextos além do simples versionamento linear de objetos. A semântica das relações entre objetos pertence à aplicação, sendo o papel da ICN permitir que essas relações sejam devidamente gerenciadas. Por exemplo, a publicação de NDOs em ICN deverá contar com um parâmetro que permita indicar quais são as relações presentes nos objetos que estão sendo publicados. Nesse caso, uma questão importante consiste em permitir que as relações de mutabilidade sejam criadas e atualizadas em qualquer momento desejado ou restringir a criação das mesmas ao momento de publicação de um objeto. Dado que a aplicação tem conhecimento dos objetos que originaram uma nova versão já no momento de criação dessa última, optamos por exigir que as relações sejam informadas na publicação, evitando-se assim ciclos. A operação de busca por objetos, por sua vez, pode incluir filtros referentes às relações de mutabilidade. Considerando-se que cada objeto tem a possibilidade de armazenar relações que apontam para versões originais, a aplicação pode solicitar à rede que obtenha um objeto e todos os originais diretamente relacionados. Além disso, é possível generalizar essa operação com a consulta de um dado número de "níveis anteriores" na hierarquia de um objeto.

4.3. Aspectos de implementação

Conforme já citado, no modelo considerado a semântica das relações de mutabilidade é compreendida pela aplicação. A ICN, por sua vez, conhece a existência de tais relações, mas não as interpreta; a rede é responsável apenas por transmitir os objetos e as relações entre eles.

Assim, no caso específico de CCNx, as relações de mutabilidade podem ser implementadas através do armazenamento das mesmas nos metadados dos objetos, os quais já são disponibilizados no referido projeto. Um cliente, ao receber um objeto, recebe também o conjunto de relações de mutabilidade que esse objeto mantém com outros. Porém, caso um cliente esteja interessado nas relações de mutabilidade, mas não no conteúdo do NDO em si, é necessário que a API da arquitetura permita a recuperação apenas dos metadados do mesmo. Tendo as relações de mutabilidade e conhecendo a semântica de tais relações, a aplicação possui todas as informações necessárias para decidir as próximas requisições que deve fazer.

Ademais, a arquitetura pode permitir que a rede, assim como as aplicações, tenha acesso somente aos metadados dos objetos, a fim de possibilitar, por exemplo, a inclusão de filtros às operações de busca conforme citado na Seção 4.2. Além disso, políticas de cache também podem fazer uso de tais relações a fim de solicitar pró-ativamente NDOs que possivelmente serão solicitados em seguida por clientes. Essa funcionalidade, porém,

requer o emprego de análises estatísticas por parte da rede sobre o uso das relações de mutabilidade entre objetos.

As operações sobre objetos permanecem, portanto, as mesmas: publicação, remoção e busca. Adiciona-se à arquitetura, entretanto, operações de busca e leitura sobre os metadados dos objetos de maneira a tornar possível a implementação das funcionalidades citadas acima; a operação de busca tornaria possível a recuperação apenas dos metadados por parte das aplicações e a operação de leitura tornaria possível as buscas condicionadas por filtros.

5. Objetivos e Organização do Trabalho

No segundo semestre deste Trabalho de Graduação, será feita uma análise do comportamento de aplicações que trabalham com objetos mutáveis em redes ICN. Tornando possível que as aplicações gerenciem as relações de mutabilidade, espera-se que as mesmas tenham seu desempenho otimizado. Ademais, reduzindo-se a redundância do conteúdo armazenado em cache através da mutabilidade de objetos descrita na Seção 4, reduz-se o consumo de recursos computacionais nos roteadores ICN. Intuitivamente, estima-se que tal abordagem geraria uma melhoria no desempenho geral da rede.

O objetivo deste trabalho é, então, avaliar o desempenho obtido com tal abordagem em redes baseadas na arquitetura CCN atual, visando contribuir com as pesquisas sendo feitas em torno do paradigma ICN. Diferentes ferramentas podem ser utilizadas para avaliação de redes baseadas no CCN, como por exemplo os simuladores ccn-Sim [Chiocchetti et al. 2013] ou ndnSim [Afanasyev et al. 2012]. Além desses simuladores, existe também o Mini-CCNx, o qual é uma ferramenta para emulação de redes CCNx que apresenta comportamento e fidelidade de desempenho compatíveis ao de um sistema real, sendo, ao mesmo tempo, flexível, escalável e de baixo custo [Cabral et al. 2013]. Dentre essas ferramentas, tem-se preferência pelo Mini-CCNx, visto que o mesmo utiliza o próprio protótipo do CCNx como base experimental, gerando resultados mais próximos daqueles de um ambiente real.

5.1. Metodologia

A seguinte metodologia está sendo empregada, visando atingir os objetivos propostos.

1. **Estudo:** o principal objetivo desta etapa é melhorar a compreensão do problema. Para isso, a mesma é composta por duas tarefas: (i) estudo aprofundado em arquiteturas de Redes Orientadas a Conteúdo, com enfoque em CCN; e (ii) levantamento bibliográfico relacionado a propostas para lidar com objetos mutáveis em ICN.
2. **Análise:** com base na etapa anterior, faz-se uma investigação dos mecanismos existentes para versionamento de objetos, procurando identificar situações nas quais a mutabilidade de objetos pode apresentar-se nociva à eficiência da arquitetura proposta pelo projeto CCNx.
3. **Busca de alternativas:** após a identificação das situações descritas na etapa de Análise, busca-se alternativas para aperfeiçoar o tratamento de objetos mutáveis no projeto CCNx. Adicionalmente, busca-se manter a eficiência atual em relação aos objetos imutáveis.
4. **Implementação:** esta etapa consiste na implementação da proposta descrita na etapa anterior, utilizando as ferramentas brevemente descritas na Seção 5.

5. **Avaliação dos resultados:** por fim, esta última etapa destina-se à avaliação da proposta de mutabilidade de objetos e sua implementação no CCNx. Para isso, os resultados obtidos serão utilizados para analisar a viabilidade da alteração proposta.

No momento atual, as etapas 1 e 2 foram realizadas e a etapa 3 está em andamento. A compreensão do problema através do estudo e análise do projeto CCNx deve, contudo, continuar sendo aprofundada durante a realização do Trabalho de Graduação 2.

5.2. Cronograma

As tarefas a serem realizadas na segunda etapa do Trabalho de Graduação encontram-se enumeradas a seguir. A Tabela 1 apresenta o cronograma de trabalho.

1. Detalhamento do modelo escolhido para o tratamento de objetos mutáveis.
2. Projeto do mecanismo de mutabilidade de objetos.
3. Implementação do modelo definido.
4. Avaliação experimental, através de experimentos com simulação e emulação do mecanismo em CCNx.
5. Redação do Trabalho de Graduação 2.
6. Apresentação do Trabalho de Graduação 2.

Tabela 1. Cronograma para as atividades do Trabalho de Graduação 2.

Tarefa	2013						
	Jun	Jul	Ago	Set	Out	Nov	Dez
1	X	X					
2	X	X					
3		X	X	X			
4				X	X	X	
5				X	X	X	
6							X

6. Considerações Finais

Através da realização desse trabalho, foi possível compreender melhor o contexto de Redes Orientadas a Conteúdo e as razões que levaram às decisões tomadas no desenvolvimento da arquitetura CCN, relatada nesse trabalho através do projeto CCNx. Além disso, através do estudo e de discussões sobre mutabilidade de objetos, pode-se também identificar possíveis contribuições ao paradigma ICN. Por fim, esse trabalho inicial permitiu estruturar de maneira mais precisa as contribuições esperadas para o Trabalho de Graduação 2.

Referências

- Afanasyev, A., Moiseenko, I., and Zhang, L. (2012). ndnSIM: NDN simulator for NS-3. Technical Report NDN-0005, NDN.

- Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., and Ohlman, B. (2012). A Survey of Information-Centric Networking. *Communications Magazine, IEEE*, 50(7):26–36.
- Bari, F., Chowdhury, S. R., Ahmed, R., Boutaba, R., and Mathieu, B. (2012). A Survey of Naming and Routing in Information Centric Networks. *IEEE Communications Magazine*, pages 1–9.
- Cabral, C. M. S., Rothenberg, C. E., and Magalhães, M. F. (2013). Mini-CCNx: prototipagem rápida para Redes Orientadas a Conteúdo baseadas em CCN. In *Anais do 31º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 1143–1150, Brasília, DF, Brasil.
- Chiocchetti, R., Schmidt, T. C., and Vahlenkamp, M. (2013). ccnSim: an Highly Scalable CCN Simulator. In *2013 IEEE International Conference on Communications, Budapest*, pages 1–6.
- Ghods, A., Koponen, T., Raghavan, B., Shenker, S., Singla, A., and Wilcox, J. (2011a). Information-Centric Networking: Seeing the Forest for the Trees. In *HotNets '11: ACM Workshop on Hot Topics in Networks*, pages 1:1–1:6, Cambridge.
- Ghods, A., Koponen, T., Rajahalme, J., Sarolahti, P., and Shenker, S. (2011b). Naming in content-oriented architectures. In *ICN '11: ACM SIGCOMM workshop on Information-Centric Networking*, pages 1–6, Orlando. ACM.
- Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M., Briggs, N., and Braynard, R. (2012). Networking Named Content. *Communications of the ACM*, 55(1):117–124.
- Kauffman, B., Peltier, J.-f., and Truong, P. (2013). Final NetInf Architecture.
- Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S., and Stoica, I. (2007). A data-oriented (and beyond) network architecture. In *SIGCOMM '07: ACM SIGCOMM Conference*, pages 181–192, Kyoto. ACM.
- Kurose, J. F. and Ross, K. W. (2012). *Computer Networking: A Top-Down Approach*. Addison-Wesley Publishing Company, USA, 6th edition.
- PARC (2013a). CCNx Documentation Index. Disponível em <<http://www.ccnx.org/releases/latest/doc/>>.
- PARC (2013b). CCNx Project, Palo Alto Research Center. Disponível em <<http://www.ccnx.org>>.
- Ribeiro, I. C. G., Guimarães, F. Q., Kazienko, J. F., Rocha, A. A. d. A., Velloso, P. B., Moraes, I. M., and Albuquerque, C. V. N. (2012). Segurança em Redes Centradas em Conteúdo: Vulnerabilidades, Ataques e Contramedidas. In *Minicursos do XII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, Curitiba, PR, Brasil. SBC SBSEG.
- Trossen, D. and Parisi, G. (2012). Designing and Realizing an Information-Centric Internet. *IEEE Communications Magazine*, 50(7):60–67.
- Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J. D., Smetters, D. K., Zhang, B., Tsodik, G., Claffy, K. C., Krioukov, D., Massey, D., Papadopoulos, C., Abdelzaher, T., Wang, L., Crowley, P., and Yeh, E. (2010). Named Data Networking Project. Technical report.