

# Redes Orientadas a Conteúdo: Abordagem no Nível de Enlace

Lisiane Maria Bannwart Ambiel<sup>1</sup>, Christian Esteve Rothenberg<sup>2</sup>,  
Maurício Ferreira Magalhães<sup>1</sup>

<sup>1</sup> Faculdade de Engenharia Elétrica e Computação (FEEC)  
Universidade Estadual de Campinas (UNICAMP)  
Campinas, São Paulo, Brasil

<sup>2</sup>Fundação CPqD – Centro de Pesquisa e Desenvolvimento em Telecomunicações  
Campinas, São Paulo, Brasil

{lisiane,mauricio}@dca.fee.unicamp.br, esteve@cpqd.com.br

**Abstract.** *This paper proposes an architecture for content oriented networking at the link layer (Ethernet) without the use of network addressing schemes. Content Routers (CR) are the basis for this architecture and are in charge of packet caching and routing directly on content names. Different from IP environments where the destination address of the content source is known, the proposed link-level architecture requests content by controlled message flooding. Questions arise concerning the introduced overhead and the overall scalability. The paper proposes design options to contain the impacts of the content-oriented flooding approach and validates the prototype implementation in some scenarios compared to an IP approach. Results suggest that content-oriented IP-less architectures may be interesting for small networks such as home networks that would largely benefit from a plug-and-play architecture that avoids network configuration and management as required with IP.*

**Resumo.** *O artigo propõe uma arquitetura de rede orientada a conteúdo no nível de enlace (Ethernet) sem uso de qualquer esquema de endereçamento. Os Content Routers (CR) são a base desta arquitetura, responsáveis pelo armazenamento de dados e roteamento de pacotes diretamente no conteúdo. Diferente do ambiente IP onde existe o conhecimento do endereço do provedor de conteúdo, a arquitetura proposta no nível de enlace requisita conteúdos através da inundação de mensagens de forma controlada. Esta condição traz consigo questões relacionadas à sobrecarga da rede e escalabilidade. O artigo propõe opções de projeto para minimizar o impacto da inundação de mensagens e valida a implementação do protótipo em alguns cenários comparando com uma abordagem IP. Os resultados sugerem que arquiteturas orientadas a conteúdo sem IP podem ser interessantes para redes de menor escala como home networks que se beneficiariam de uma arquitetura plug-and-play sem necessidade de configuração e gerenciamento como é o caso do IP.*

## 1. Introdução

As Redes Orientadas a Conteúdo (ROC) se apresentam como uma nova forma de pensar a Internet: mudam o paradigma de comunicação apresentando uma nova abordagem

com base no conteúdo independente de sua localização [de Brito et al. 2012]. A Internet foi projetada para interconectar computadores para aplicações como troca de correspondências e transferência de arquivos. Hoje é usada principalmente para disseminação e recuperação de conteúdos ou informação. Os usuários, em geral, não estão interessados em saber de onde vem a informação e somente precisam de uma garantia que a informação é autêntica. Esta abordagem torna a arquitetura da rede mais adequada para a distribuição de conteúdo e se utiliza de novos conceitos como conteúdo nomeado, roteamento baseado em nomes, segurança aplicada diretamente a conteúdos e armazenamento de dados nos nós intermediários da rede (*caching*).

Entre as arquiteturas propostas estão: DONA [Koponen et al. 2007], CCN [Jacobson et al. 2009], PSIRP [Tarkoma et al. 2009] e NetInf [Ahlgren et al. 2008]. Em comum estas arquiteturas utilizam nomes de conteúdo únicos e persistentes, modelo de segurança que permite verificação da integridade e origem do dado independente da fonte, simplicidade para suportar situações de mobilidade e múltiplos provedores (*multihoming*) e tolerância a falhas. Porém, ainda existem desafios para que estas redes sejam utilizadas em grande escala devido ao fato do número de conteúdos ser muito maior do que o número de endereços na rede atual, o que causa impacto no sistema de roteamento e resolução de nomes [Ahlgren et al. 2012]. Estudos indicam uma mudança no espaço de endereçamento de um bilhão de endereços IP para um trilhão de nomes de conteúdos [Perino and Varvello 2011]. Algumas questões são colocadas para pesquisa com relação à manutenção de estado por pacote, custos de processamento e de armazenamento no roteador, redução do tamanho de tabelas armazenadas de forma eficiente nos roteadores e como fazer buscas e operações em alta velocidade. Com relação às estratégias de encaminhamento é preciso investigar como descobrir um caminho para envio de uma requisição de um novo conteúdo e como escolher o melhor caminho quando se tem múltiplas opções [Yi et al. 2012].

Este trabalho propõe uma rede orientada a conteúdos no nível da camada de enlace (Ethernet) utilizando uma arquitetura de Content Routers (CRs) sem uso de endereçamento de rede com o objetivo de tornar o conteúdo mais importante. Desta forma não há necessidade de configuração de parâmetros de rede como no caso do IP. São utilizados mecanismos como: uso de identificadores planos para nomeação; armazenamento através de *caching* nos CRs; encaminhamento (*forwarding*) com base em tabela de roteamento ao invés de inundação (*flooding*) e entrega de conteúdo sem uso do endereço do cliente (*delivery*). A arquitetura dos CRs foi inicialmente definida por Wong et al. [Wong et al. 2011] como uma rede *overlay* sobre IP com suporte ao *caching* e utilização de nomes planos entre outras características. Diferente do ambiente IP onde existe o conhecimento do endereço do provedor, nesta proposta de arquitetura no nível de enlace (Ethernet) parte-se do princípio de que o conteúdo não é conhecido e a requisição de conteúdos é feita através de mecanismo de inundação de mensagens pelas interfaces nos CRs. Ao propor tal condição surgem questões relacionadas à quantidade de mensagens geradas na rede. Assim, para reduzir o número de mensagens, além do roteamento oportunístico baseado nas mensagens de RESPONSE, introduz-se: (a) primitivas para divulgação de conteúdos como forma de minimizar o impacto causado pelo *flooding*, (b) estrutura de dados compactadas (Bloom Filters) para manter a lista de requisições pendentes por interface; (c) conceito de Super CR como elemento especializado em *caching* para evitar inundação de mensagens, criando plano de controle e hierarquia.

Neste trabalho é estudado o comportamento de uma rede orientada a conteúdos através de experimentos com versões do protótipo para as alternativas IP e Ethernet. Entre outras dimensões de desempenho, são avaliadas a relação de *caching* disponível nos CRs sobre a quantidade de mensagens atendidas por um servidor de conteúdo e a distância percorrida em *hops* para atender a solicitação. Também é possível avaliar a relação do mecanismo de busca de conteúdo sobre a quantidade de mensagens geradas na rede. O tempo de transferência do conteúdo requisitado também pode ser observado nos diversos cenários porém deve ser considerado de forma relativa pois os CRs (protótipo) executam em ambiente virtual num mesmo servidor. A proposta no nível de enlace introduzida neste artigo apresenta-se como uma alternativa interessante para redes locais tais como *home networks* onde os custos relativos às dependências do IP (ex: gerência de rede, configuração de protocolos de roteamento e interfaces, segurança) seriam eliminados com uma orientação nativa da rede para um serviço de busca e entrega de conteúdo.

O artigo está organizado de acordo com a seguinte estrutura: a Seção 2 apresenta algumas definições relacionadas; na Seção 3 apresenta-se a arquitetura seguida da Seção 4 que descreve detalhes da implementação do protótipo; a Seção 5 apresenta a metodologia utilizada e o ambiente de teste; a Seção 6 apresenta uma discussão dos resultados obtidos. Ao final, na Seção 7 são apresentadas as conclusões e trabalhos futuros.

## 2. Background

Os seguintes conceitos chaves são propostos no contexto de Redes Orientadas a Conteúdo (ROC) [Ahlgren et al. 2012, de Brito et al. 2012]:

**Objetos de dados nomeados:** representam os conteúdos como, por exemplo, páginas WEB, documentos, filmes, fotos, músicas, enfim, todos os tipos de objetos que podem ser armazenados e acessados via computador. Um conteúdo mantém o seu nome independente da sua localização ou método de armazenamento.

**Nomeação de conteúdos (*naming*):** esquemas de nomeação que permitem identificar o conteúdo e requisitar sua distribuição à infraestrutura de rede. Esquemas básicos: nomeação plana, nomeação hierárquica e nomeação por atributo.

**Roteamento baseado em nomes (*name-based routing*):** a rede entrega os conteúdos requisitados por nome sem qualquer informação referente à localização, tanto de usuário quanto de armazenamento de conteúdos.

**Armazenamento de conteúdos (*caching*):** qualquer nó da rede pode atuar como um *cache* independente da aplicação ou protocolo. É aplicável a qualquer conteúdo, inclusive aos gerados pelo usuário cliente. Uma requisição por determinado conteúdo pode ser respondida por qualquer nó que tenha a cópia em seu *cache*.

Os mecanismos para nomeação podem ser classificados em três classes básicas: nomes planos, hierárquicos e baseados em atributos [Choi et al. 2011]. Cada uma das classes de nomes atende parcialmente os requisitos exigidos dos mecanismos de nomeação: persistência, escalabilidade e inteligibilidade ao usuário final. A nomeação hierárquica é legível e escalável, mas possui restrições ao uso persistente dos nomes. Esta restrição não existe no caso de nomes planos que apresentam vantagem com relação à segurança, mas seu uso depende de um processo de resolução entre um nome inteligível e o nome plano. O uso de atributos relacionados ao conteúdo pode facilitar a busca mas depende

de definições adequadas para evitar ambiguidades. Os mecanismos de roteamento de conteúdos podem ser classificados em dois grupos: roteamento não-estruturado, normalmente baseado em *flooding*, e roteamento hierárquico (ou estruturado) baseado em árvores hierárquicas ou tabelas hash distribuídas (*Distributed Hash Tables* - DHT).

Além destes, devem ser considerados como aspectos comuns aos projetos de ROC as primitivas e o modelo de segurança [Ghodsi et al. 2011]. As primitivas básicas utilizam o nome do conteúdo e o paradigma *publish/subscribe* onde o provedor do conteúdo não conhece a localização do solicitante e vice-versa. Da mesma forma provedor e solicitante de conteúdo não precisam estar *online* simultaneamente. No modelo de segurança o conteúdo é assegurado pelo provedor original de forma que os elementos da rede e os consumidores possam verificar a sua validade.

Fazendo uma avaliação sistemática dos componentes de hardware e software existentes nos roteadores para suporte às redes de conteúdo, Perino e Varvello [Perino and Varvello 2011] afirmam que o novo paradigma vai provocar mudanças nos roteadores: maior velocidade para suportar encaminhamento com base no nome do conteúdo e armazenamento de pacotes de dados; mudança no espaço de endereçamento (de um bilhão de endereços IP para um trilhão de nomes de conteúdos). Apesar do aumento da informação a ser armazenada, o *caching* local pode aliviar potencialmente a frequência das operações de encaminhamento. Entre as soluções propostas, estratégias probabilísticas baseadas em estruturas como os Bloom Filters aparecem como abordagens promissoras. O trabalho [Perino and Varvello 2011] conclui que a tecnologia atual não está pronta para suportar as redes de conteúdo na escala da Internet, mas é viável no nível de uma rede de distribuição de conteúdos (CDN – *Content Delivery Network*) ou no escopo de um provedor de serviços (ISP – *Internet Service Provider*).

### 3. Arquitetura de CRs no Nível de Enlace

A arquitetura é composta por roteadores, os *Content Routers* (CR), que são os elementos de rede que realizam o encaminhamento de mensagens e o armazenamento (*caching*) oportunístico de conteúdos. A proposta original da arquitetura [Wong et al. 2011] é baseada numa rede *overlay* sobre IP com suporte ao *caching* e utilizando nomes planos entre outras características. Também são considerados elementos da rede: Cliente, que faz as requisições e Servidor, que tem o conteúdo original, faz o anúncio do mesmo e responde às requisições. A arquitetura se aplica aos ambientes IP e Ethernet com algumas diferenças que são explicitadas a seguir. Primeiramente, são introduzidas as seguintes definições:

**Data Chunk:** Unidade básica de comunicação; parte de um conteúdo (ex. arquivo) que será identificado e requisitado. Cada *Data Chunk*, ou simplesmente *chunk*, é identificado pelo seu *chunkId*.

**Cryptographic Identifier (CryptoId):** Identificação do conteúdo que resulta de um hash criptográfico aplicado a um bloco de dados (*Data Chunk*).

**Metadata:** Estrutura de dados que agrega a lista de *chunkIds* relacionados a um conteúdo e outras informações adicionais como versão e validade. Informação gerada pelo provedor do conteúdo (Servidor) e que deve ser recuperado pelo Cliente para proceder às requisições dos *Data Chunks* por um mecanismo de resolução de nomes.

**Caching Threshold:** Determina a probabilidade de se armazenar um conteúdo. Valor possível de ser configurado.

**Neighbor Zones (NZones):** Total de CRs a visitar antes de encaminhar uma requisição para o servidor (ambiente IP) ou via *flooding* (ambiente Ethernet). Significa que a requisição será desviada na tentativa de buscar um conteúdo em um CR vizinho até o limite de NZones. Como esse número pode indicar uma área em volta do CR, usa-se o termo *zones*. Valor possível de ser configurado.

### 3.1. Características principais da rede

Na sequência são apresentadas as principais características adotadas para a arquitetura.

**Nomeação:** utiliza identificador plano para nomear os conteúdos (*chunkIds*). Este tipo de identificação atende ao requisito de persistência pois está desacoplado da informação de localização e dá suporte ao modelo de segurança para rede de conteúdos.

**Registro dos conteúdos:** os servidores anunciam opcionalmente os conteúdos através da primitiva `ANNOUNCE_CONTENT` que contém o *chunkId*. Os roteadores armazenam a informação na tabela de rotas. Esta informação evita o *flooding* pois a requisição por um *chunkId* conhecido será encaminhada somente na direção do vizinho que repassou a mensagem de anúncio do conteúdo anteriormente.

**Roteamento:** somente as requisições são roteadas. O roteamento é feito exclusivamente com base nos *chunkId*. A tabela é preenchida com a informação anunciada pelos servidores ou de forma oportunística com a informação da mensagem `RESPONSE` que trafega na rede. Desta forma o CR cria o conhecimento das rotas para um conteúdo, mesmo que este não esteja disponível no cache local. A tabela mantém somente o melhor caminho para um conteúdo, ou seja, a menor distância em *hops*.

- No ambiente IP: se o conteúdo não é encontrado na tabela de rotas a requisição é encaminhada ao servidor (endereço de destino) com base no roteamento IP ativo na rede (ex: OSPF), ou seja, neste ambiente temos dois níveis de roteamento (ou roteamento híbrido): inicialmente no conteúdo e, caso este alcance o limite definido pela variável NZones, pelo roteamento IP na direção do servidor do conteúdo.
- No ambiente Ethernet: se o conteúdo não é encontrado na tabela de rotas é feita a inundação da requisição pelas interfaces disponíveis pois não existe o conhecimento de endereços como é o caso do ambiente IP. O roteador mantém informação sobre quais conteúdos (*chunkIds*) estão sendo aguardados por interface – equivalente a tabela PIT (Pending Interest Table) da arquitetura CCN [Jacobson et al. 2009] e não encaminha requisições por conteúdos que já estão na lista de pendências. Esta tabela é implementada com estruturas de Bloom filter [Tarkoma et al. 2012] para reduzir os requisitos de memória.

#### Mecanismo de entrega de dados:

- No ambiente IP o dado é roteado para o elemento que fez a requisição.
- No ambiente Ethernet o dado faz o caminho inverso ao da requisição. Ao receber uma mensagem `RESPONSE` o CR verifica se o *chunkId* é esperado e repassa a mensagem às interfaces que tem o *chunkId* na sua lista de pendências. Em seguida o *chunkId* é removido das listas.

**Caching:** O objetivo desse armazenamento é que solicitações futuras pelo mesmo conteúdo sejam atendidas a partir dos CRs ao invés de percorrerem toda a rede até o servidor original, atendendo ao requisito de disponibilidade.

**Segurança:** Os chunkIds são nomes auto verificáveis. O sistema utiliza Merkle Tree e tem o objetivo de permitir a recuperação segura dos conteúdos de múltiplas fontes ou de fonte não conhecida. Cada conteúdo tem a informação de segurança associada permitindo que o receptor verifique a validade e a integridade do mesmo independente do local de recuperação. Atende ao requisito de autenticação [Wong et al. 2011].

**Primitivas (API - *Application Programming Interfaces*):** REQUEST, originada pelo Cliente que requisita um *chunkId*; RESPONSE, originada por um Servidor ou por um CR que tem o dado correspondente ao *chunkId* no seu cache em resposta a um REQUEST; ANNOUNCE\_CONTENT, originada pelo Servidor para anunciar os *chunkIds* disponíveis, reconhecida pelos CRs; e ANNOUNCE\_SCR, originada pelo CR que está configurado como Super CR, reconhecida pelos CRs.

### 3.2. Content Router (CR)

O CR faz o tratamento das mensagens que trafegam na rede conforme descrito a seguir.

As mensagens REQUEST são respondidas diretamente pelo CR quando o *chunkId* e o dado correspondente são encontrados na sua tabela de *caching* (*cache-hit*). Se o CR não tem o conteúdo (*cache-miss*), verifica a tabela de rotas e faz o roteamento da mensagem conforme explicado anteriormente. A base para esse processo de decisão está na presença ou não de informação na tabela de roteamento e na variável *NZones*.

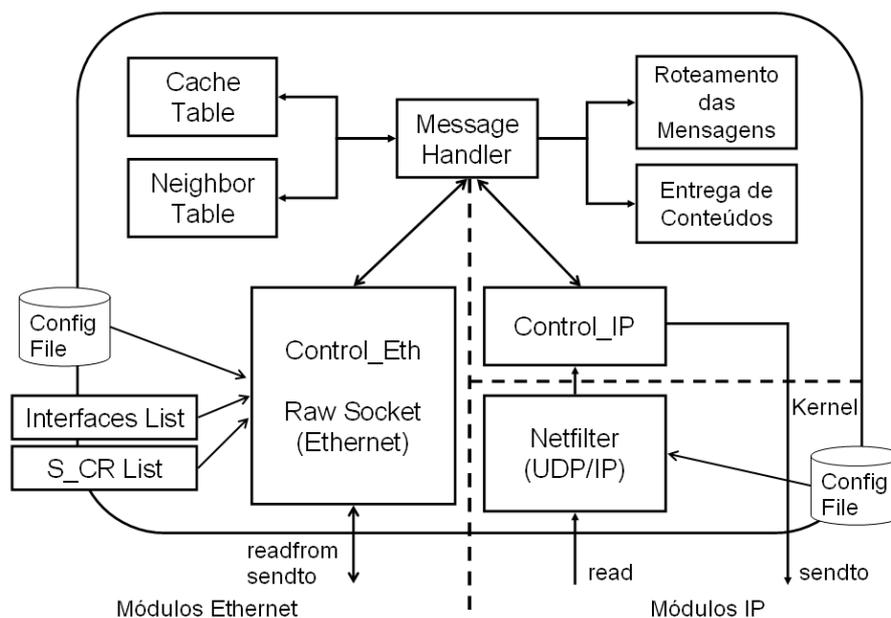
As mensagens RESPONSE são encaminhadas aos clientes que originaram a requisição. No ambiente IP este processo é feito pelo protocolo de roteamento ativo. No ambiente Ethernet é utilizada a informação de requisições pendentes. No tratamento da mensagem RESPONSE pode ocorrer o *caching* do conteúdo nos CRs, em função do parâmetro *Caching Threshold*. Esse parâmetro é aplicado na função de probabilidade a fim de não sobrecarregar o cache pois o mesmo tem capacidade limitada.

Não existe nenhum tipo de sinalização entre os CRs, ou seja, eles não funcionam de forma cooperada com relação ao *caching*. Um mesmo conteúdo pode estar armazenado probabilisticamente em todos os CRs no caminho feito por uma mensagem RESPONSE.

Os CRs podem ser especializados (Super CR) na função de *caching* como forma de atender a requisitos de escalabilidade evitando inundação de mensagens. Este tipo de CR cria hierarquia e conceito de plano de controle pela troca de mensagem específica (ANNOUNCE\_SCR). Assim, um CR que não tem o conteúdo no cache local e não conhece a rota, tenta o Super CR antes de recorrer ao *flooding*.

## 4. Implementação

O CR é composto por um módulo de controle que trata da comunicação e um módulo que trata da manipulação das mensagens. O módulo de controle é especializado para atender aos ambiente IP e Ethernet; intercepta mensagens em trânsito, inspeciona o cabeçalho e faz recepção e envio de mensagens. O módulo de manipulação das mensagens verifica e armazena conteúdos na tabela de *caching*, verifica e armazena informação na tabela de rotas e responde às requisições relativas aos conteúdos por ele armazenado.



**Figura 1. Módulos do CR**

O CR mantém duas estruturas de dados importantes: *Cache Table* e *Neighbor Table* (ou tabela de rotas). As duas tabelas são indexadas pelo identificador do conteúdo, *chunkId*, têm capacidade limitada e estratégia de substituição simples (FIFO). A Figura 1 dá uma visão geral dos módulos.

No ambiente Ethernet o CR mantém a lista das interfaces disponíveis (*Interfaces List*) e a lista de Super CRs (*S\_CR List*). Cada interface mantém as requisições pendentes (como a PIT no CCN [Jacobson et al. 2009]) utilizando estrutura de Bloom Filter tipo contador, assim o CR evita que a requisição por um mesmo *chunkId* não seja reencaminhada e minimiza a quantidade de mensagens gerada na rede. Ao receber uma resposta o CR encaminha o dado a todas as interfaces que tem o *chunkId* pendente e remove a requisição pendente.

O protótipo foi desenvolvido em linguagem C e implementa as funções dos elementos CR, Cliente e Servidor. A implementação atual não inclui: mecanismo de resolução de nomes e modelo de segurança. O Cliente tem disponível o metadata dos conteúdos para iniciar as requisições. No ambiente IP não se tem o anúncio de conteúdos.

Basicamente, o funcionamento dos elementos do protótipo é conforme o seguinte processo: o *cliente* requisita conteúdos (*chunkId*) através das mensagens REQUEST e aguarda os dados pela mensagem RESPONSE; re-envia a requisição no caso de *timeout*. O *servidor* espera por solicitação de conteúdos e responde com dados através da mensagem RESPONSE quando tem o conteúdo. Os CRs interceptam as mensagens na rede e processam conforme as informações presentes nas tabelas de roteamento e *caching* e configuração do NZones.

## 5. Metodologia Experimental

Este trabalho utiliza método de pesquisa quantitativa e procedimento do tipo experimento [Creswell 2009]. Participam do experimento todos os elementos da rede: clientes, servi-

dores e roteadores de conteúdo (CRs). Todos os participantes são designados por conveniência, ou seja, não existe nenhum mecanismo aleatório na seleção dos participantes.

Os elementos da rede são um conjunto de máquinas virtuais criadas utilizando a ferramenta XEN<sup>1</sup> que permite que elas atuem como roteadores. Todos os elementos são executados em ambiente Debian GNU/Linux. Para os testes da rede em modo *overlay*/IP, os roteadores utilizam a pilha de protocolos *open-source* Quagga<sup>2</sup> mediante o protocolo OSPF (*Open Shortest Path First*).

Como variáveis independentes (as que causam ou influenciam nos resultados) temos as seguintes:

- *caching*: probabilidade de armazenamento dos roteadores da rede.
- *neighbor zones*: busca de conteúdo em caminho conhecido/aprendido até determinado limite de *hops* (profundidade).
- *flooding*: possibilidade de uso de mecanismo de inundação da requisição até encontrar um conteúdo não conhecido (somente CR no ambiente Ethernet).

Além destas variáveis, colaboram na consequência dos resultados a dimensão das tabelas de *caching* e de roteamento.

Variáveis dependentes ou de resultado (as consequências):

- utilização dos servidores e roteadores (quantidade de mensagens tratadas)
- distância da resolução em número de *hops*
- tempo de transferência do conteúdo

Uma topologia de 12 CRs, 4 servidores e 8 clientes é utilizada para os testes da rede de CRs nos ambientes IP ou Ethernet (Figura 2). Para fins deste artigo somente os resultados de testes a partir de um único cliente são apresentados para facilitar entendimento das variáveis envolvidas. Testes com variação da topologia foram executados e apresentaram comportamento comparáveis.

O protótipo é usado como instrumento no experimento. Todos os elementos geram arquivos com registro de eventos (arquivos de LOG) e arquivos com contadores no formato CSV (*comma separated value*). Esses arquivos são gerados quando o elemento de rede recebe mensagem de controle ao final de uma solicitação de um conteúdo e ao final do teste (seqüência de 30 solicitações). As mensagens de controle têm função operacional (ex: LOG, STOP) sem relação com o protocolo da rede. Como a topologia e as interfaces de cada CR são conhecidas, o envio destas mensagens é feito de forma direcionada para interfaces pré-definidas em cada CR garantindo que todos os elementos recebam a mensagem sem necessidade de recorrer ao *flooding*.

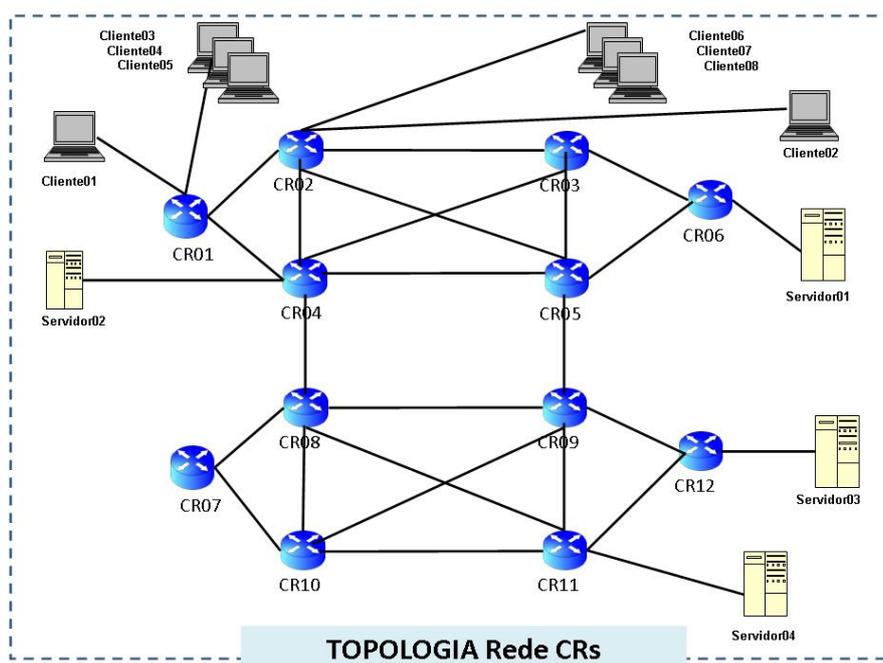
## 5.1. Procedimento

Um teste é composto de uma seqüência de 30 solicitações de conteúdos seguindo uma distribuição *power law*: 10 arquivos são solicitados sendo que cada um deles tem um grau de popularidade, ou seja, alguns vão ser solicitados com maior freqüência. A distribuição de popularidade por conteúdo é resultado de programa que gera número aleatório e o *script* de execução do teste. A Tabela 1 mostra a lista dos arquivos com respectiva quantidade de *chunks* e quantas vezes cada arquivo é solicitado no teste.

---

<sup>1</sup><http://www.xensource.com>

<sup>2</sup><http://www.quagga.org>



**Figura 2. Participantes do experimento - Topologia**

O teste é iniciado por um elemento cliente que faz o papel do solicitante. Ao final do teste os arquivos de LOG e CSV são coletados nos diversos elementos. Os testes são repetidos com variação de valores das variáveis independentes descritas anteriormente. Os valores são atribuídos através de arquivo de configuração.

## 5.2. Medidas

As medidas são coletadas através de mensagens de controle enviadas pelo elemento que faz a requisição. Os roteadores e servidores ao receberem uma mensagem de LOG registram os contadores (ex: total de mensagens recebidas, enviadas, cache hit etc). Tais medidas são armazenadas em arquivos CSV e então são traduzidas para tabelas no aplicativo Excel para cálculo de média entre testes do mesmo cenário, análise entre diferentes cenários, geração de gráficos e comparações.

**Tabela 1. Arquivos de conteúdo do procedimento de testes.**

#	Nome do arquivo	Qtde. chunks	# Requisições/teste
1	aliancas.jpg	1059	0
2	biblioteca.jpg	563	1
3	DistributedCachingAlgoritms.pdf	175	2
4	EnhanceContentBroadcast.pdf	118	1
5	GatewayControlledContentCaching.pdf	286	2
6	PacktpubMoodleSecurity.pdf	3406	16
7	SantaClausIsComingToTown.mp3	5099	2
8	TeachYourselfFreeBSD.pdf	7176	1
9	TikosGrooveFeatGosha.mp3	6590	1
10	towerEiffel.jpg	411	4
	<b>TOTAL</b>	<b>24888</b>	<b>30</b>

## 6. Resultados

Após a execução dos testes e consolidação dos resultados foram gerados gráficos para análise e avaliação. Nestes cenários, o requisitante é o *Cliente01* conforme Figura 2.

### 6.1. Ambiente IP/Overlay

Os gráficos na Figura 3(a) e na Figura 3(b) mostram o comportamento da rede orientada a conteúdo para ambiente IP/overlay<sup>3</sup> nos diferentes cenários: probabilidade de *caching* (0%, 50%, 70%) e valor de *neighbor zones* *NZ* (0, 3, 4, 6). É possível observar uma complementariedade entre a resolução nos servidores e nos CRs.

Nos cenários (1) sem disponibilidade de cache nos CRs, a resolução das requisições acontece sempre nos servidores. O valor de *NZ* não colabora no processo pois a requisição é encaminhada sempre para o endereço do servidor.

Para cache habilitado observar que a resolução acontece em grande parte nos CRs, diminuindo a carga nos servidores, porém sem muitas vantagens no aumento da probabilidade de 50% para 70%. A busca na vizinhança (*NZ*) também colabora para aumentar a resolução na rede.

As Figuras 4(a) e 4(b) mostram, no caso do IP, a tendência da resolução das requisições no *CR1*, mais próximo do *Cliente01* que fez as requisições, quando *NZ* = 6. Alguns CRs não participam do tratamento de mensagens (CRs 3,7 e 10) pois o OSPF não fez a opção pela rota que passa por eles.

### 6.2. Ambiente Ethernet

Os gráficos na Figura 5(a) e na Figura 5(b) mostram o comportamento da rede orientada a conteúdo para a proposta no nível de enlace (ambiente Ethernet) para os diferentes cenários sob avaliação (Probabilidade de *caching*, # *Neighbor Zones* – *NZ*).

Nos cenários sem cache (1) onde a resolução das requisições acontece nos servidores o valor de *NZ* tem impactos diversos: *NZ* = 0 : não faz busca, pára, conforme esperado. *NZ* = 3 : gera menor número de mensagens; resolução no servidor mais

<sup>3</sup>Vale a pena notar que não são contabilizadas as mensagens do protocolo OSPF (ex: Hello, LSA, LSU).

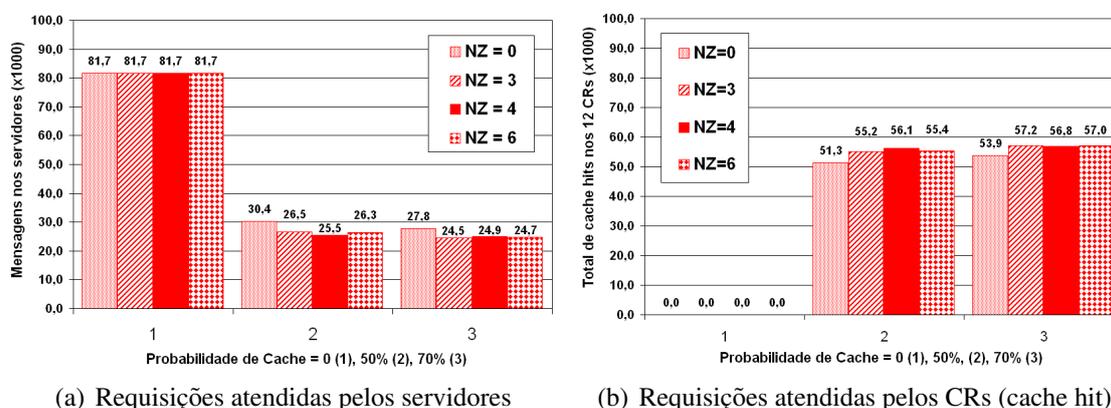
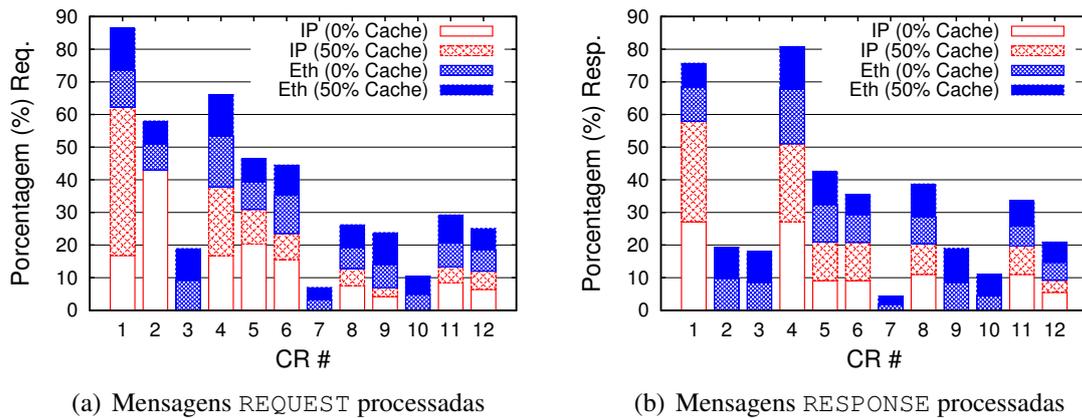


Figura 3. Tratamento de requisições (IP)



**Figura 4. Distribuição de mensagens nos CRs (IP vs. Ethernet).**

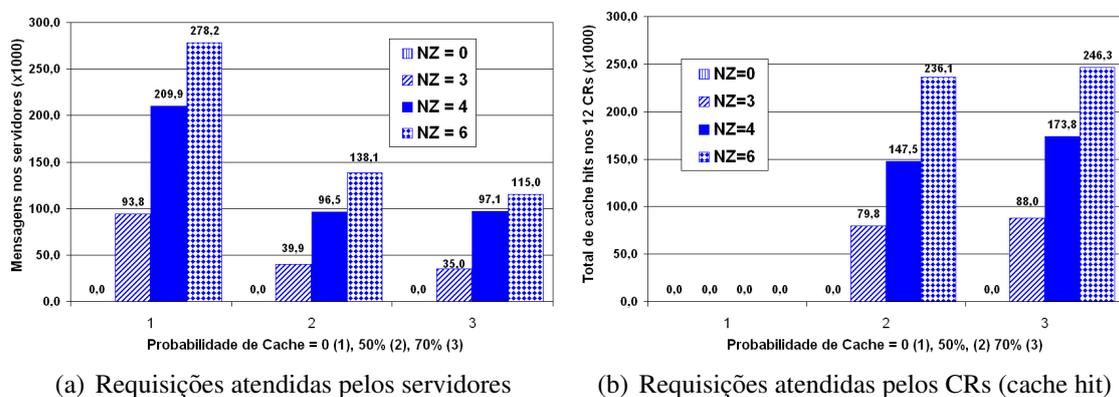
próximo (*Servidor02*).  $NZ = 4$  ou  $6$  : aumenta o número de mensagens geradas na rede; resolução distribuída nos demais servidores.

Para cache habilitado verificar que a resolução acontece nos CRs, diminuindo a carga nos servidores, porém sem muitas vantagens no aumento da probabilidade de 50% para 70% como no caso do IP. A busca na vizinhança ( $NZ$ ) colabora para aumentar o número de mensagens na rede.

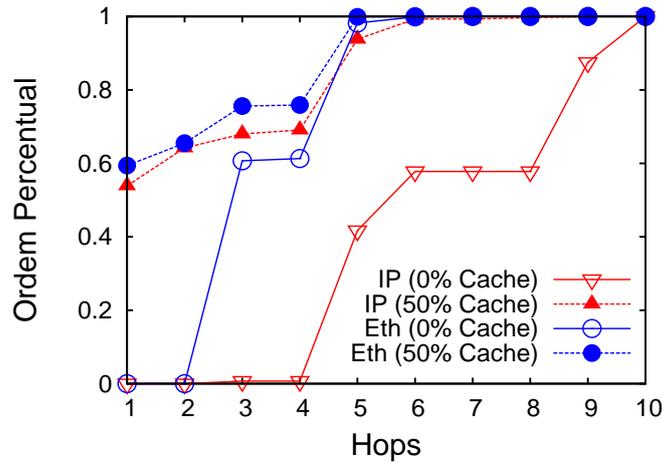
As Figuras 4(a) e 4(b) mostram a distribuição das requisições nos CRs quando  $NZ = 6$ . Nesta condição todos os CRs respondem às requisições apesar de nem todas serem “utilizadas”: o *flooding* causa várias respostas para uma mesma requisição; estas respostas serão ignoradas pelos CRs que não têm a requisição pendente.

### 6.3. Ambiente IP vs. Ethernet

A topologia do tipo Internet utilizada nos testes coloca a solução no nível de enlace em desvantagem comparada à solução IP/overlay pois aumenta o número de pontos de inundação e aumenta a quantidade de mensagens. Nas figuras 3 e 5 é possível observar que apesar das quantidades de mensagens geradas na rede serem diferentes (Ethernet precisando de mais mensagens que o IP), o comportamento dos dois ambientes tende a ser o mesmo. As Figuras 4(a) e 4(b) mostram que a rede Ethernet distribui melhor o



**Figura 5. Tratamento de requisições (Ethernet)**



**Figura 6. IP vs. Ethernet: Distância em saltos para resolução das requisições**

processamento entre os diferentes CR sem sobrecarregar nenhum CR em particular.

Já o gráfico na Figura 6 faz uma comparação dos dois ambientes em termos de número de *hops* desde a fonte do conteúdo (cache ou servidor) até o cliente consumidor. Vale a pena destacar que o ambiente Ethernet consegue caminhos mais curtos (60% das vezes com 4 ou menos *hops*) que o IP (60% das vezes com 6 ou menos *hops*) e atinge resultados quase comparáveis aos da rede IP com cache ativado (concentração entre 3 e 4 *hops*). Observar que em alguns casos a distância percorrida é maior que NZones definido para o teste ( $NZ = 6$ ). Isso ocorre porque houve tentativa de busca na vizinhança sem sucesso (roteamento no conteúdo) e ocorre um acréscimo de *hops* além da distância normal até o servidor (roteamento no IP).

Finalmente, a Tabela 6.3 compara a requisição de um mesmo arquivo. No *testbed* com os elementos virtualizados em uma máquina física o tempo não é um resultado que possa ser considerado de forma absoluta pelos efeitos da virtualização e o consumo concorrente de CPU, mas apresenta certa coerência. Os números mostram que a rede Ethernet tem um custo maior para a resolução da primeira requisição devido a busca por *flooding*, porém, após o conhecimento da rota, nas requisições subsequentes as duas redes se comportam de forma equivalente. O tempo se refere ao tempo de recepção do arquivo (3406 chunks) pelo *Cliente01* medido desde o tempo de envio do primeiro REQUEST.

**Tabela 2. IP vs. Ethernet: Tempo de transferência em segundos de um arquivo de 3406 chunks para  $NZ = 4$  e  $Cache = 50\%$ .**

Requisição #	1 <sup>a</sup>	2 <sup>a</sup>	3 <sup>a</sup>	4 <sup>a</sup>	5 <sup>a</sup>
IP	24,55	5,62	3,50	2,86	2,56
Ethernet	42,67	4,92	3,68	2,96	2,77

## 7. Conclusões e Trabalhos Futuros

O objetivo da arquitetura proposta consiste em uma abordagem de roteamento diferente das baseadas na propagação de mensagens de alcançabilidade como no caso do CCN. O artigo discute duas arquiteturas diferentes: a primeira é efetivamente uma arquitetura

híbrida com roteamento no conteúdo e no IP e a segunda é completamente limitada ao nível de enlace com roteamento no conteúdo. Com os resultados obtidos até o momento algumas conclusões já podem ser apresentadas. A principal vantagem desta proposta é dispensar o roteamento baseado em localização e operar diretamente na camada de enlace através do roteamento baseado no conteúdo. A contrapartida da abordagem é um maior número de mensagens trocadas na rede. Porém, considerando o desempenho apresentado pela proposta uma vez aprendidos os caminhos até o conteúdo, vale a pena considerar o *trade-off* de sobrecarga em tráfego de controle pela simplicidade de uma arquitetura *plug-and-play* no nível de enlace. Como na arquitetura CCN, a nossa proposta não precisa do IP e faz roteamento com base no conteúdo, porém o CCN não tem uma proposta específica para o nível de enlace.

Abordagens no espírito da proposta deste trabalho podem ser uma opção para redes domiciliares (HomeNets) [Velloso et al. 2004] ou outros tipos de redes menores onde não existe necessidade de endereçamento IP ou configurações mais complexas. Com o aumento de dispositivos do tipo *tablets, laptops, smartphones* e aparelhos de segurança para monitoramento é esperada adição de roteadores nestas redes. O uso de *gateways* não IP como o Super CR é uma opção para aumentar a complexidade da rede sem aumento da complexidade de operação da rede.

Entre trabalhos relacionados, destacamos a arquitetura CONET [Detti et al. 2011] por apresentar uma abordagem no nível de enlace além de IP/overlay e IP integrado. Similar à proposta dos CRs no aprendizado das rotas, a arquitetura CONET limita o tamanho da tabela mas introduz um elemento centralizado com informações de rotas que serve todos os demais elementos em um sub-sistema. A idéia de especialização dos nós é semelhante à nossa proposta de Super CR que pode atuar como *gateway* para integrar a rede Ethernet com a rede IP.

Como trabalhos futuros temos identificadas as seguintes áreas de atuação a serem implementadas e testadas com o protótipo:

**Arquitetura:** o Super CR pode ter suas funções ampliadas e funcionar como: (i) ponto de *rendezvous* para suporte ao modelo *publish/subscribe* como PSIRP [Tarkoma et al. 2009], (ii) ponto de acesso inter-domínio ou *gateway* para compor um ambiente misto IP/Ethernet. Outros testes e análises serão feitos para verificar comportamento com diferentes opções de *flooding* e o uso das mensagens de anúncio de conteúdo na rede Ethernet, avaliando o impacto da inserção de um elemento especializado como Super CR.

**Otimização de mecanismos:** as tabelas de roteamento e *caching* podem utilizar outras estratégias além da utilizada atualmente (FIFO); a tabela de roteamento mantém somente o melhor caminho e não uma lista de opções como na proposta original da arquitetura CCN [Jacobson et al. 2009] e o processamento não é instanciado por interface mas um único processo atende todas as interfaces do CR; o *flooding* por todas as interfaces pode ser evitado pelo aprendizado e identificação das melhores interfaces (cf. [Yi et al. 2012]), seja pelo tempo ou percentual de respostas; inserção do mecanismo de resolução de nomes e obtenção do *metadata* e uso de um *proxy* para aplicações legadas; inserção de políticas de *caching*; otimizações nas estruturas dos Bloom filters.

**Aplicação em cenários de HomeNets:** instanciar uma versão do protótipo numa rede domiciliar combinando ambiente sem fio e cabeado onde o Super CR atua como *gateway*

para o mundo IP e onde qualquer dispositivo pode atuar como CR provendo ampla capacidade de *cached* e colaborativamente oferecendo um sistema de arquivos distribuído para os usuários da HomeNet.

## Referências

- Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., and Ohlman, B. (2012). A survey of information-centric networking. *IEEE Communications Magazine*, 50(7):26–36.
- Ahlgren et al. (2008). Design considerations for a network of information. In *ACM CoNEXT '08*, pages 1–6, Madrid, Spain.
- Choi et al. (2011). A Survey on content-oriented networking for efficient content delivery. *IEEE Communications Magazine*, 49(3):121–127.
- Creswell, J. W. (2009). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. Sage, Los Angeles, third edit edition.
- de Brito, G. M., Velloso, P. B., and Moraes, I. M. (2012). Redes Orientadas a Conteúdo: Um Novo Paradigma para a Internet. In *SBRC 2012*.
- Detti, A., Blefari Melazzi, N., Salsano, S., and Pomposini, M. (2011). CONET: a content centric inter-networking architecture. In *Proceedings of the ACM SIGCOMM Workshop on Information-Centric Networking - ICN '11*, pages 50–55.
- Ghodsi, A., Shenker, S., Koponen, T., Singla, A., Raghavan, B., and Wilcox, J. (2011). Information-centric networking: Seeing the forest for the trees. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, pages 1–6, Cambridge, USA.
- Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L. (2009). Networking named content. In *CoNEXT'09*, pages 1–12, Rome, Italy.
- 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. *ACM SIGCOMM CCR*, pages 181–192.
- Perino, D. and Varvello, M. (2011). A reality check for content centric networking. In *Proceedings of the ACM SIGCOMM workshop on Information-centric networking - ICN '11*, pages 44–49, New York, New York, USA. ACM Press.
- Tarkoma, S., Ain, M., and Visala, K. (2009). The Publish/Subscribe Internet Routing Paradigm (PSIRP): Designing the Future Internet Architecture. In *Towards the Future Internet - A European Research Perspective, 2009*, pages 102–111.
- Tarkoma, S., Rothenberg, C. E., and Lagerspetz, E. (2012). Theory and Practice of Bloom Filters for Distributed Systems. *IEEE Communications Surveys & Tutorials*, 14(1):131–155.
- Velloso, P., Cunha, D., Amodei Jr, A., Rubinstein, M., and Duarte, O. (2004). Redes domiciliares: Princípios e desafios das tecnologias sem novos fios. *Minicursos SBRC2004*, pages 221–268.
- Wong, W., Giralddi, M., Magalhaes, M. F., and Kangasharju, J. (2011). Content Routers: Fetching Data on Network Path. In *IEEE ICC*, pages 1–6. IEEE.
- Yi, C., Afanasyev, A., Wang, L., Zhang, B., and Zhang, L. (2012). Adaptive forwarding in named data networking. *ACM SIGCOMM CCR*, 42(3):62–67.