Capítulo 5

Redes Orientadas a Conteúdo: Um Novo Paradigma para a Internet

Gabriel M. de Brito, Pedro B. Velloso e Igor M. Moraes

Laboratório MídiaCom, PGC-TCC/Instituto de Computação
Universidade Federal Fluminense

Abstract

Content-Oriented Networks (CONs) are a new communication paradigm to the Internet. This new paradigm focuses on the content delivery to users regardless of the location of the content rather than the current Internet architecture that focuses on the communication between end systems. In CONs, the network infrastructure also actively contributes to content caching and distribution. The main advantage of this new paradigm is to increase the efficiency of content delivery and also the content availability. Furthermore, CONs simplifies the solution of current Internet problems, such as mobility and content security. This chapter presents the basic concepts of CONs, describes the main architecture proposals for these networks, and discusses the main challenges for its development. The challenges include naming, routing, and caching on the network-core elements and also several aspects of content security, users' privacy, and issues to implement CONs in practice.

Resumo

As Redes Orientadas a Conteúdo (Content-Oriented Networks), ou simplesmente ROCs, representam um novo paradigma de comunicação para a Internet. Nesse novo paradigma, o foco é a entrega de um dado conteúdo para os usuários independentemente da localização desse conteúdo, ao contrário da arquitetura atual da Internet em que o foco é a comunicação entre sistemas finais. Nas ROCs a infraestrutura da rede também participa ativamente do armazenamento e da distribuição dos conteúdos. A principal vantagem desse novo paradigma, portanto, é aumentar a eficiência da entrega e a disponibilidade de conteúdos. Além disso, as ROCs simplificam soluções para problemas da Internet atual, como a mobilidade e a segurança dos conteúdos. Este capítulo apresenta os conceitos básicos das ROCs, descreve as principais propostas de arquiteturas para tais redes e discute os principais desafios para seu desenvolvimento. Entre esses desafios estão a nomeação de conteúdos, o roteamento e o uso de caches de conteúdo nos elementos do núcleo da rede e também questões relativas à segurança de conteúdos, à privacidade dos usuários e aos aspectos práticos para implantação das ROCs.

5.1. Introdução

No início do desenvolvimento da Internet, o principal problema a ser resolvido em termos de comunicação entre computadores era o compartilhamento de recursos [Jacobson et al. 2009a, Jacobson et al. 2012]. Estações interconectadas necessitavam trocar informações, como arquivos e registros em bancos de dados, e também permitiam acesso a equipamentos remotos, como impressoras. O objetivo, portanto, era a comunicação eficiente entre estações. Hoje, novas tecnologias de rede de núcleo e de acesso proporcionam um aumento significativo da banda passante disponível e redução dos custos de conexão de usuários. Essa popularização do acesso à Internet estimula a criação de novas aplicações, como sistemas de publicação de vídeos e redes par-a-par (*peer-to-peer* - P2P) de compartilhamento de arquivos, remodelando completamente a forma de uso da Internet pelos usuários finais. Portanto, a distribuição de conteúdo na Internet passou por um processo de evolução, afastando-se da definição de sistema de informação textual em direção à de um sistema de informação multimídia, no qual dados, serviços e aplicações são consumidos como conteúdos [Plagemann et al. 2005]. Esse novo modelo enfatiza o interesse pelo conteúdo, independente de sua localização física ou lógica. Conteúdo, nesse contexto, consiste de dados codificados ou dados multimídias, como vídeo, áudio, documentos, imagens e páginas *web*, por exemplo, e metadados, isto é, dados a respeito dos dados que possibilitam sua localização, sua interpretação e seu gerenciamento [Plagemann et al. 2005]. Essa nova caracterização implica a existência de uma infraestrutura que possibilite a requisição e transmissão de conteúdo de forma eficiente, segura e com alta disponibilidade. Apesar dessa mudança na característica de uso das aplicações, os protocolos mais utilizados para a obtenção de conteúdo na Internet são, ainda, orientados à localização.

Atualmente, ainda não existe uma solução única que contemple todos os requisitos da distribuição de conteúdo. Basicamente, existem técnicas que tentam atender parcialmente esses requisitos e contornar as limitações da atual arquitetura da Internet. Redes P2P e redes de distribuição de conteúdos (*Content Distribution Networks* - CDNs) são soluções largamente adotadas para o atendimento desse objetivo, como comprova o sucesso de aplicações como o BitTorrent e de provedores de CDNs como a Akamai. Porém, a arquitetura atual da Internet remete a problemas de persistência, disponibilidade e segurança dos conteúdos, uma vez que tais aplicações fazem uso de soluções específicas e/ou proprietárias. O uso de redirecionamentos HTTP (*HyperText Transfer Protocol*) e DNS (*Domain Name System*) dinâmico em CDNs, por exemplo, não garante a persistência das informações. O reposicionamento dos dados na rede também requer consultas a estruturas centralizadas, aumentando o tempo total de entrega do conteúdo [Koponen et al. 2007]. Portanto, se faz necessária uma mudança radical na arquitetura atual da Internet. Deve-se levar em conta aspectos para aumentar a eficiência da localização e da entrega e a disponibilidade de conteúdos. Esses requisitos são atendidos pelas redes orientadas a conteúdo (ROCs).

As redes orientadas a conteúdo introduzem um novo paradigma de comunicação para a Internet. As ROCs enfatizam o acesso à informação independentemente de sua localização. Diferentemente da abordagem tradicional da Internet, centrada na identificação e localização de estações, as ROCs utilizam conceitos inovadores como conteúdo nomeado, roteamento baseado em nomes, segurança aplicada diretamente a conteúdos e

armazenamento de dados nos elementos do núcleo da rede [Jacobson et al. 2009a, Koponen et al. 2007, Visala et al. 2009] Tais conceitos permitem criar uma arquitetura mais eficiente para a distribuição de conteúdo, evitando assim todos os remendos necessários à arquitetura vigente da Internet, como o IP Multicast, o uso do DNS, IPSec etc. A arquitetura baseada em conteúdo pode, então, prover de forma nativa novas funcionalidades, como o compartilhamento eficiente de recursos e de dados, mecanismos para aumentar a disponibilidade dos conteúdos, suporte à segurança intrínseca de conteúdos, suporte à mobilidade etc.

Neste capítulo são apresenta os conceitos básicos das ROCs, apontando seus principais diferenciais em relação à arquitetura tradicional da Internet centrada na comunicação entre usuários. São também descritas e analisadas as principais propostas de arquitetura para as ROCs, bem como os projetos de pesquisa de maior destaque. Por fim, são apresentados os principais desafios para o desenvolvimento das ROCs, como a nomeação e o roteamento de conteúdos a segurança de conteúdos e a privacidade de usuários, o uso de *caches* de conteúdo nos elementos do núcleo da rede e aspectos práticos para a implantação das ROCs em larga escala.

O restante deste capítulo é organizado em quatro seções. A Seção 5.2, de forma sucinta, caracteriza a atual arquitetura da Internet e discute suas limitações para o desenvolvimento de aplicações de distribuição de conteúdo. Descreve-se, ainda, as principais técnicas empregadas para contornar essas limitações, como a comunicação multidestinatária, as redes par-a-par, os sistemas *publish/subscribe* e as redes de distribuição de conteúdo. A Seção 5.3 apresenta o novo paradigma introduzido pelas ROCs, descrevendo suas principais características, propostas de arquitetura existentes e projetos mais relevantes. A Seção 5.4 discute alguns dos principais problemas ainda em aberto no desenvolvimento das ROCs, como a nomeação, o roteamento, o armazenamento (*caching*), a segurança do conteúdo e a viabilidade técnica. Por fim, a Seção 5.5 apresenta algumas considerações sobre o tema e as perspectivas futuras para desenvolvimento desse novo paradigma de comunicação para a Internet.

5.2. A Distribuição de Conteúdo na Internet

No início, as aplicações na Internet eram baseadas em informações textuais. Usuários apenas trocavam mensagens de correio eletrônico, transferiam arquivos via FTP (*File Transfer Protocol*) e acessavam servidores remotamente. Hoje, a Internet é considerada um complexo sistema de informação multimídia baseado na distribuição de conteúdos. Entende-se por conteúdo dados textuais, codificados ou multimídias, como documentos, páginas *web*, arquivos de áudio e vídeo, e também metadados, que são usados para localizar, interpretar e gerenciar os próprios conteúdos [Plagemann et al. 2005]. Essa nova caracterização da Internet implica a existência de uma infraestrutura de rede que possibilite a requisição e transmissão de conteúdos de forma eficiente. Para tanto, alguns requisitos básicos devem ser atendidos. O primeiro requisito é garantir a persistência dos conteúdos. A persistência é uma propriedade dos identificadores[1] de conteúdos que define que eles devem ser únicos e válidos enquanto o conteúdo associado permanecer válido. Atualmente, mesmo usuários leigos conseguem facilmente publicar conteúdos na

[1]Neste capítulo os termos *identificadores* e *nomes* são utilizados como sinônimos.

Internet, o que confere um caráter bastante volátil às informações geradas. Por isso, o uso persistente de identificadores representa um grande desafio à distribuição de conteúdo na Internet. O segundo requisito é a escalabilidade, ou seja, os mecanismos de localização e encaminhamento de conteúdos devem ser eficientes para o número de usuários e de conteúdos disponibilizados na escala da Internet. O acesso seguro a conteúdos é o terceiro requisito, que visa garantir a autenticidade e o controle de acesso aos conteúdos disponibilizados. Atualmente não existe uma única solução que contemple todos esses requisitos. Existem apenas técnicas que tentam atendê-los de forma parcial e contornar as limitações da atual arquitetura da Internet.

5.2.1. A Arquitetura Centrada em Sistemas Finais e suas Limitações

Três características da arquitetura atual da Internet são os principais fatores que impedem que os requisitos das aplicações de distribuição de conteúdo sejam atendidos satisfatoriamente: a ausência de garantias de (i) qualidade de serviço e (ii) segurança fim-a-fim e de (iii) mecanismos de encaminhamento escaláveis.

A Internet é uma rede de comutação de pacotes de escala global na qual os pacotes são encaminhados segundo o modelo de melhor esforço oferecido pelo protocolo IP (*Internet Protocol*). Não há reserva de recursos para cada usuário da rede, nem diferenciação no tratamento dos pacotes encaminhados pelos roteadores. Assim, não há garantias de desempenho ou segurança fim-a-fim para a distribuição de conteúdo na Internet atual. Além disso, o foco da arquitetura atual é a comunicação entre sistemas finais.Nesse contexto, a comunicação entre estações na Internet é centrada em sistemas finais, já que a estação de origem precisa indicar no cabeçalho dos pacotes o endereço IP da estação de destino com a qual deseja se comunicar. O encaminhamento de tais pacotes é realizado por cada um dos nós (*hops*) existentes no caminho entre as estações de origem e destino, utilizando como único balizador o endereço IP da estação de destino final. Esse paradigma atende as aplicações iniciais da Internet, cujo objetivo era o compartilhamento de recursos. Porém, ele não é eficiente para a distribuição de conteúdo, pois, entre outros fatores, exige que o usuário conheça a localização e não somente a identificação do conteúdo.

Atualmente existem soluções paliativas, ou “remendos" para aumentar a eficiência da distribuição de conteúdo na Internet. Um exemplo é o redirecionamento HTTP usado para tratar a localização de conteúdos em virtude da volatilidade desses conteúdos. Objetos *web* são requisitados através de localizadores de recursos, denominados URLs (*Uniform Resource Locator*), presentes no cabeçalho das mensagens HTTP. Os redirecionamentos HTTP são eventos disparados pelo servidor que originalmente hospedava os objetos, enviando como resposta mensagens de redirecionamento com a nova URL em seu cabeçalho. Como tal solução é dependente da localização do conteúdo, se faz necessário desenvolver mecanismos que garantam o acesso persistente a esses conteúdos, independentemente de localização, propriedade ou qualquer outra característica pertinente associada a eles. Esse exemplo também ilustra os conceitos do modelo cliente-servidor adotado por muitas das aplicações de distribuição de conteúdo na Internet atual. Nesse modelo, um canal de comunicação ponto-a-ponto é estabelecido entre um determinado cliente e um servidor. Assim, diversos usuários que solicitam simultaneamente um mesmo conteúdo são atendidos através do estabelecimento de múltiplos canais ponto-a-ponto e do envio de uma cópia do mesmo conteúdo em cada canal. Nesse cenário, quanto mais pop-

ular um conteúdo, maior a ineficiência no uso dos recursos, principalmente, em termos de banda passante. Assim, a adoção de soluções em larga escala requer o desenvolvimento de técnicas de encaminhamento eficientes que confiram escalabilidade à distribuição de conteúdo na Internet.

As aplicações atuais de distribuição de conteúdos buscam, ainda, garantir a autenticidade das informações e segurança na comunicação sobre a Internet. Para tal, utilizam mecanismos que visam assegurar o canal de comunicação ao invés de utilizar segurança explicitamente sobre o conteúdo. Isso gera uma camada adicional de complexidade e sobrecarga de mensagens e processamento [Smetters e Jacobson 2009]. O IPSec (*Internet Protocol Security*) é outro exemplo de remendo utilizado no tratamento de segurança. Através da inserção de cabeçalhos de autenticação (*Authentication Headers* - AH), de criptografia aplicada às informações e encapsulamento do pacote original (*Encapsulating Security Payloads* - ESP) e de mecanismos de administração de chaves, o IPSec permite estabelecimento de conexões confiáveis entre os nós. Essa abordagem orientada a conexão amarra a segurança do conteúdo à confiança na estação que o armazena e à conexão estabelecida entre as partes, impossibilitando aumento de escalabilidade com o compartilhamento do conteúdo entre sistemas distintos. É necessário o estabelecimento de múltiplas conexões seguras entre fontes de conteúdos e os diversos usuários, impossibilitando o armazenamento e uso posterior de conteúdo previamente requisitados [Smetters e Jacobson 2009].

5.2.2. Comunicação Multidestinatária

A comunicação multidestinatária (*multicast*) foi uma das primeiras alternativas propostas a fim de aumentar a eficiência da distribuição de conteúdo na Internet. Uma proposta para implementá-la na camada de rede é o serviço IP Multicast [Deering 1989]. Esse serviço permite o envio de datagramas para múltiplos sistemas finais, cujos interesses comuns de recepção de dados permitem agregá-los em grupos. Esses grupos são identificados por um único endereço IP. Assim, se uma estação enviar um datagrama ao endereço IP do grupo, todas as estações que fazem parte desse grupo o receberão. É responsabilidade da rede encaminhar e replicar, quando necessário, esse datagrama por toda a árvore de distribuição que cobre os receptores interessados. Apesar de suas vantagens e de ter sido proposto na década de 90, a implantação do IP Multicast em larga escala na Internet ainda não ocorreu, em virtude da complexidade para configurar e gerenciar o conjunto de protocolos exigidos por esse serviço. Essa complexidade é introduzida pelo próprio modelo de serviço IP Multicast, que define um grupo como uma conversação aberta muitos-para-muitos [Costa e Duarte 2003].

5.2.3. Redes Par-a-Par

As redes par-a-par (*peer-to-peer* - P2P) também buscam aumentar a eficiência da distribuição de conteúdo. Dada a dificuldade da adoção da comunicação multidestinatária na camada de rede, por exigir modificação no núcleo da rede, soluções que implementam tal comunicação na camada de aplicação são propostas [Moraes et al. 2008]. Esse é o caso das redes P2P, as quais são formadas por nós que, através do compartilhamento de recursos computacionais e de conteúdos, dividem a carga de provimento de serviços entre si de forma cooperativa. Cada par contribui com parte de seus recursos e usufrui do

serviço distribuído prestado pela rede [Passarella 2012]. Dessa forma, cada novo par que adentra a rede utiliza uma parcela da capacidade total enquanto disponibiliza seus próprios recursos aos demais pares. Isso faz da escalabilidade uma característica intrínseca às redes P2P. Consequentemente, quanto mais nós pertencerem à rede P2P, maior será a capacidade da rede em atender seus usuários, o que pode reduzir o tempo de entrega e aumentar a disponibilidade de conteúdos [Moraes et al. 2008].

Outra característica fundamental é a de que um usuário está interessado em receber um dado conteúdo, seja ele um arquivo ou fluxo multimídia, sem se importar com quem o envia. No BitTorrent, por exemplo, um par, ao entrar na rede P2P, seleciona aleatoriamente seus parceiros, com os quais trocará pedaços do conteúdo. Esses parceiros são sorteados de um subconjunto dos pares que se interessam por um mesmo conteúdo e nenhuma informação sobre localização ou identificação dos pares é levada em consideração nesse processo de escolha. O sucesso das redes P2P de compartilhamento de arquivos e de distribuição de vídeo, que possuem milhões de usuários, é um forte indicativo da mudança de paradigma das aplicações da Internet e que sustenta o principal fundamento das ROCs, como será visto na Seção 5.3: usuários cada vez mais estão interessados no conteúdo e não mais em quem o envia.

Apesar de serem soluções escaláveis para a distribuição de conteúdo na Internet, as redes P2P apresentam problemas cruciais de segurança e incentivo à colaboração para o seu uso na distribuição de conteúdos. Uma vez que a rede é distribuída e suas funcionalidades são executadas de forma colaborativa por todos os nós, a confiança nos dados encaminhados pelos nós passa a ser um ponto crítico, o qual as redes P2P devem tratar. Outro ponto crítico é a robustez à entrada e saída de nós da rede (*churn*) que acarreta problemas de eficiência de distribuição e disponibilidade de conteúdos, uma vez que não existe uma infraestrutura dedicada para gerenciar esses eventos.

5.2.4. Redes de Distribuição de Conteúdo

As redes de distribuição de conteúdo (*Content Distribution Networks* - CDNs) foram propostas para o aumentar a eficiência e a escalabilidade do modelo cliente-servidor empregado por aplicações de distribuição de conteúdos na Internet [Passarella 2012]. As CDNs são formadas por um conjunto de servidores distribuídos, interconectados pela Internet, que operam de forma cooperativa na distribuição de conteúdo [Buyya et al. 2008]. Os aumentos de disponibilidade e da eficiência na distribuição de conteúdos se dão através da replicação de conteúdos em diferentes localidades e, sempre que possível, em diferentes provedores de acesso à Internet. A idéia principal é de redirecionar as requisições de conteúdo para uma das réplicas de acordo com regras de seleção/redirecionamento. Assim, o armazenamento dos conteúdos em servidores mais próximos aos clientes torna a entrega desses conteúdos mais eficiente, aumentando as taxas de transferência de dados devido à diminuição dos gargalos nas redes de acesso e reduzindo a latência no acesso dada a proximidade entre os sistemas finais envolvidos.

O conceito básico de uma arquitetura de CDN pode ser representado por dois blocos funcionais: um serviço de distribuição e replicação de conteúdos e um serviço de redirecionamento de requisições de conteúdo [Passarella 2012]. O serviço de distribuição e replicação de conteúdos é intimamente ligado aos produtores de conteúdo e trata questões

relativas à localização de servidores, à alocação de espaço de armazenamento e à alocação de conteúdos nos servidores. O serviço de redirecionamento de requisições, interface da CDN com os consumidores de conteúdos, é responsável pelo recebimento de requisições de conteúdo e encaminhamento aos nós da rede CDN mais adequados para atender à demanda. Uma CDN típica é composta por dois tipos de servidores: o servidor de origem e o servidor de réplica. O servidor de origem é o responsável pelo armazenamento, atribuição de identificadores e divulgação do conteúdo. O servidor de réplica, por sua vez, é responsável por encaminhar o conteúdo para um dado cliente. Requisições enviadas para o servidor de origem são redirecionadas para o servidor de réplica mais próximo do cliente, uma vez que o conteúdo desejado esteja disponível em tal servidor, processo ilustrado na Figura 5.1. Dessa forma, os mecanismos de redirecionamento são extremamente importantes para o correto funcionamento das CDNs.

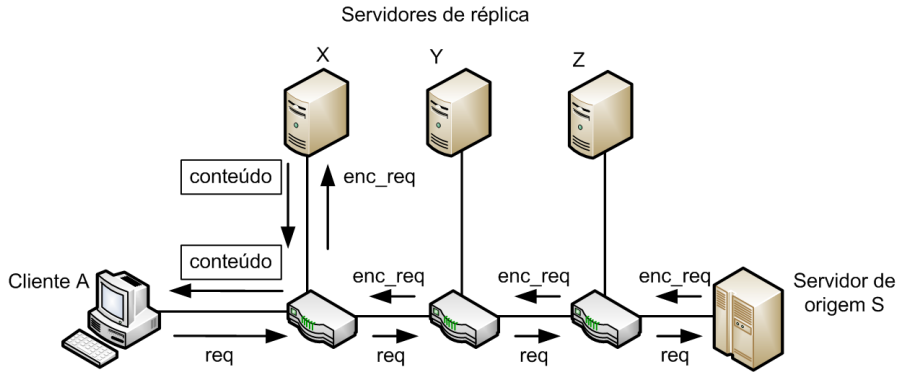


Figure 5.1. O funcionamento de uma CDN [Moraes et al. 2008].

A técnica mais simples de redirecionamento é utilizar o HTTP, em que todas as requisições de objetos *web* são feitas através de um navegador de Internet executado nos sistemas finais dos usuários. Ao receber a requisição, o servidor de origem envia mensagens de redirecionamento HTTP com o endereço do servidor de réplica adequado. Tal técnica delega ao servidor de origem todo o processamento de requisições, tornando-o um gargalo potencial e ponto singular de falha. Outra técnica bastante utilizada consiste em aplicar configurações dinâmicas do DNS que, quando consultado sobre o nome associado ao servidor de origem, envia como resposta o endereço do servidor de réplica mais adequado. Ambas as técnicas podem utilizar informações relativas à distância do servidor de réplica em relação ao cliente, como quantidade de saltos ou tempos de ida e volta (*Round-Trip Time* - RTT), bem como informações de utilização dos servidores que podem ser coletadas periodicamente pelos sistemas. Ainda que permitam o encaminhamento de requisições a servidores diferentes dos originalmente endereçados, tais técnicas não conferem persistência efetiva aos dados. Alterações de propriedade, domínio e outras características de determinado conteúdo podem inviabilizar sua obtenção à partir da URL previamente conhecida. Adicionalmente, o reposicionamento dos dados na rede requer consultas a estruturas centralizadas, aumentando o tempo total de entrega do conteúdo [Koponen et al. 2007].

As CDNs, porém, não são projetadas visando a interoperabilidade de aplicações, e sim o atendimento de demandas específicas, como é o caso da obtenção de objetos *web*.

Por isso, são baseadas em implementações proprietárias a cada aplicação consumidora de conteúdos. Uma vez que não propicia um substrato genérico o suficiente para o tratamento do problema de distribuição de conteúdo, as CDNs não se adaptam totalmente às demandas dos assinantes. Adicionalmente, decisões sobre posicionamento de servidores, dimensionamento da capacidade de armazenamento de cada nó, bem como políticas de atualização de *cache* são fundamentais para e eficiência das CDNs [Passarella 2012]. Evidentemente, tais processos são altamente custosos, tanto do ponto de vista computacional, uma vez que algoritmos para decisão e distribuição de réplicas devem ser executados em tempo real, quanto do ponto de vista financeiro, já que a distribuição geográfica de recursos computacionais e de rede reflete altos investimentos por parte dos provedores dos serviços de CDN.

Existem diversos exemplos de provedores de CDNs tanto acadêmicas, como a CoDeeN [Wang et al. 2004], quanto comerciais, como a Limelight [2] e a Akamai [3], sendo a última a mais popular. Estima-se que a Akamai possua mais de 100 mil servidores espalhados pela Internet, com pontos de presença em 72 países, suportando trilhões de interações por dia [Akamai Technologies 2012].

5.2.5. Sistemas *Publish/Subscribe*

Os sistemas *publish/subscribe* (*pub/sub*), assim como as redes P2P, podem ser vistos como um dos principais indicativos da mudança de paradigma das aplicações da Internet. Em ambos os casos, os usuários das aplicações estão interessados em receber o conteúdo de interesse, não importando quem o envie.

Em sistemas *pub/sub*, as informações de interesse dos usuários são denominadas *eventos*, enquanto o ato de entrega dessas informações é denominado de *notificação*. Assim, assinantes (*subscribers*) são capazes de expressar seus interesses em eventos ou padrões de eventos definidos pelos publicadores. Uma vez tendo manifestado interesse, um assinante será notificado sempre que for gerado um evento por quaisquer publicadores (*publishers*) que casem com seu interesse.

Os sistemas *pub/sub* dissociam assinantes e publicadores tanto no espaço quanto no tempo [Eugster et al. 2003]. Um assinante, por exemplo, pode manifestar o interesse em um evento ainda não publicado ou em um instante no qual o publicador desse evento não está em funcionamento. A dissociação é uma característica desejável, pois proporciona escalabilidade ao sistema, uma vez que permite aos participantes do sistema operarem de forma independente [Eugster et al. 2003]. Produtores publicam conteúdos apenas injetando informações no sistema usando a função `publish()`, enquanto consumidores expressam seus interesses através de assinaturas declarativas, usando a função `subscribe()`, delegando ao sistema *pub/sub* a responsabilidade do armazenamento de assinaturas e da entrega dos conteúdos a todos os assinantes interessados, como ilustrado na Figura 5.2. Essa característica permite a disseminação de informações entre milhões de produtores e assinantes, uma vez que produtores não necessitam manter estados relativos a todos os interesses dos assinantes, enquanto estes recebem as informações sem conhecer, especificamente, o produtor que as enviou [Majumder et al. 2009]. Esse é mais

[2] `http://www.limelight.com/`
[3] `http://www.akamai.com/`

um exemplo claro de que usuários estão interessados nos conteúdos e não mais em quem os envia.

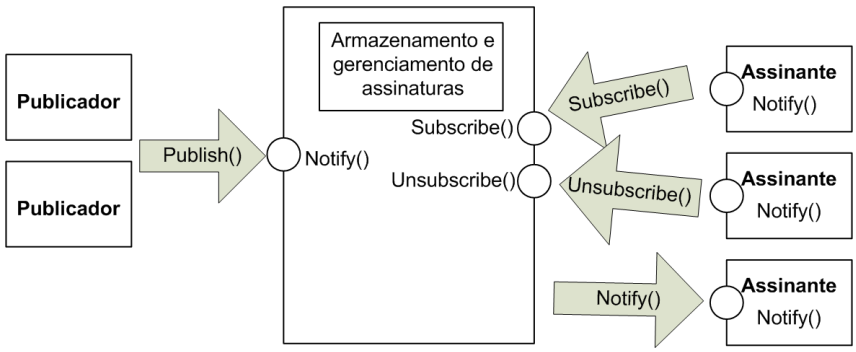


Figure 5.2. Serviço de assinatura e notificação de eventos em arquiteturas *pub/sub* [Eugster et al. 2003].

Os primeiros sistemas *pub/sub* propostos eram baseados em tópicos identificados por palavras-chave (*Topic-based* Pub/Sub). Sistemas de integração de aplicações corporativas, de monitoração do mercado de ações, de alimentação RSS (*Really Simple Sindycation*), plataformas de jogos *on-line*, são exemplos desse tipo de sistema [Chockler et al. 2007]. Em sistemas baseados em tópicos, os usuários assinam e publicam eventos através do tópico, que possui um conceito bastante similar ao conceito de comunicação multidestinatária, visto na Seção 5.2.2. Cada tópico é visto como um serviço de eventos único, identificado por um nome único, que fornece interfaces para a utilização das funções de publicação e assinatura.

Também foram propostos sistemas *pub/sub* baseados em conteúdo (*Content-based* Pub/Sub) como uma evolução dos sistemas de tópicos. Esses sistemas permitem a assinatura de eventos baseada em propriedades dos próprios eventos e não em características estáticas e previamente definidas como a identificação de tópicos. Usuários podem especificar filtros para definição de suas assinaturas, utilizando restrições baseadas em pares de atributos-valores (*Attribute-Value Pairs* - AVPs) e operadores lógicos comparativos básicos, como $=$, $<$, $>$, $\leq$ e $\geq$. Restrições podem, ainda, ser combinadas logicamente através de operadores `AND`, `OR`, etc., formando padrões complexos de assinaturas. Tais padrões são utilizados na identificação de eventos de interesse de determinado assinante, além de serem usados no encaminhamento das notificações de eventos por toda a rede. Os filtros conferem aos sistemas baseados em conteúdo uma maior liberdade na declaração de interesses em relação aos sistemas baseados em tópicos, muito embora tenha como contrapartida um grande sobrecarga de comunicação dada à grande sobreposição de eventos que pode existir no caso de um interesse parcialmente declarado. Siena [Carzaniga et al. 2001] e Kyra [Cao et al. 2004] são exemplos de sistemas *pub/sub* baseados em conteúdos.

Independente da forma com a qual usuários especificam os eventos de interesse, a arquitetura dos sistemas *pub/sub* pode ser classificada em centralizada ou distribuída [Eugster et al. 2003]. A arquitetura centralizada opera de forma que produtores de eventos enviam mensagens a uma entidade específica que as armazena e redireciona sob demanda aos assinantes. Aplicações baseadas em arquiteturas centralizadas possuem requisitos estritos em relação à disponibilidade e à consistência dos dados, já que se estruturam so-

bre uma única entidade central. A arquitetura distribuída, por sua vez, não possui uma entidade central responsável por tratar interesses e notificações, distribuindo tais responsabilidades entre todos os nós do sistema. Arquiteturas distribuídas são adequadas para a entrega rápida e eficiente de dados, uma vez que utilizam mecanismos de comunicação multidestinatária. Assim, sistemas *pub/sub* baseados em tópicos podem beneficiar-se do conceito de comunicação em grupo, porém a comunicação multidestinatária eficiente em sistemas *pub/sub* baseados em conteúdos é um grande desafio já que seu desempenho é altamente afetado pelo custo computacional da filtragem de eventos para distribuição, o qual depende diretamente da quantidade de assinaturas no sistema.

5.3. Redes Orientadas a Conteúdo

As redes orientadas a conteúdo mudam radicalmente o paradigma de comunicação da Internet. Apresentando uma nova abordagem de comunicação baseada apenas no conteúdo, as ROCs enfatizam o acesso à informação independente de sua localização, tornando a arquitetura da rede adequada para a distribuição de conteúdo. As ROCs utilizam alguns conceitos inovadores como conteúdo nomeado, roteamento baseado em nomes, segurança aplicada diretamente a conteúdos e armazenamento de dados nos nós intermediários da rede [Jacobson et al. 2009a]. Por isso, este novo enfoque traz uma série de desafios ao desenvolvimento das ROCs, como métodos para nomeação e roteamento de conteúdos, técnicas para proteção de conteúdos e usuários, planejamento e utilização de *cache* no núcleo da rede, entre outros. Nesta seção são apresentados os conceitos básicos relativos às ROCs, ressaltando suas vantagens e desvantagens. São também descritas as principais arquiteturas propostas e projetos de maior destaque.

5.3.1. Nomeação de Conteúdos

Como visto na Seção 5.2.1, a obtenção de conteúdo na arquitetura atual da Internet, centrada em sistemas finais, implica o conhecimento de todas as partes envolvidas nas transferências de dados através de seus endereços IP. Dessa forma a obtenção de um determinado conteúdo requer o conhecimento *a priori* da sua localização na topologia da rede, ou seja, o conhecimento do endereço do sistema final que o hospeda, para o estabelecimento posterior de uma conexão a fim de que seja possível requisitar uma cópia do conteúdo. Esta característica amarra de forma estrita os conceitos de identificação e localização de conteúdos.

Em se tratando de identificação de conteúdos, a abordagem das ROCs baseia-se em uma premissa bastante diferente da abordagem tradicional da arquitetura centrada em estações. Por tratar os conteúdos como primitiva básica de rede, as ROCs tornam possível a obtenção de conteúdo através, apenas, de sua identificação ou nome, utilizando esquemas de nomeação de conteúdos, com propriedades bastante específicas. Um esquema ideal para nomeação de conteúdos deve apresentar as seguintes características:

- **Unicidade:** Garantir a identificação do conteúdo de forma única, sem falsos positivos ou negativos.
- **Persistência:** Validar o nome do conteúdo em sincronismo com a validade do próprio conteúdo.

- **Escalabilidade:** Permitir sua adoção em uma variedade de cenários, servindo a espaços de nomes de pequeno a grande porte, não impondo limitações quanto à sua natureza, local de armazenamento ou qualquer outra característica.

Para tal, são utilizados esquemas de nomeação que permitem identificar o conteúdo e requisitar sua distribuição à infraestrutura de rede de forma eficiente, segura e com alta disponibilidade. São empregadas três técnicas básicas de nomeação em redes orientadas a conteúdo: nomeação plana, nomeação hierárquica e nomeação por atributos.

5.3.1.1. Nomeação Plana

Os nomes planos podem ser entendidos por cadeias de *bits* de aparência aleatória, utilizados na identificação de conteúdos. Os esquemas de nomeação plana aplicados à identificação de conteúdos utilizam diferentes abordagens de mapeamento de conteúdos em identificadores planos, sendo o mais comum a utilização de funções *hash* de criptografia. Devido ao fato de não possuírem semântica, isto é, regras para formatação ou codificação de informações nos identificadores, os nomes planos são persistentes, pois não há relação direta entre localização, propriedade ou qualquer outra característica além do vínculo entre o conteúdo em si e seu nome. Por exemplo, a função *hash* SHA-1[4] mapeia palavras originais menores que 2^{64} bits em chaves *hash* de 160 bits, utilizando diversas operações booleanas em blocos de bits da palavra original [Wang et al. 2005]. Tal mapeamento depende somente do conteúdo da palavra original, retornando uma palavra de tamanho fixo para diferentes comprimentos de palavras originais, composta de caracteres sem correlação. A unicidade também é, de certo modo, garantida, dado que as funções *hash* devem conferir uma baixa probabilidade de colisão em seu mapeamento [Peyravian et al. 1998]. Uma vez que as funções *hash* de criptografia retornam cadeias de *bits* de comprimento fixo a partir de blocos de dados arbitrários, uma característica comum às propostas de nomeação plana é o comprimento fixo de seus identificadores.

Um conceito importante, possibilitado pela utilização de nomes planos, é a autocertificação de conteúdos e seus identificadores. Utilizando-se pares de *hashes* criptográficos no formato `P:L`, no qual `P` representa o *hash* criptográfico da chave pública do publicador [Koponen et al. 2007] ou do conteúdo em si [Dannewitz et al. 2010] e `L` representa um rótulo arbitrário escolhido pelo publicador, os usuários possuem meios para verificar a validade da chave utilizada na codificação do conteúdo e do vínculo entre este e seu nome [Ghodsi et al. 2011b]. Dessa forma, havendo um vínculo entre nome e chave de criptografia dos publicadores, basta aos usuários reconhecerem o vínculo entre nome e a identidade real do publicador para uma certificação completa dos conteúdos. Uma vez que sistemas robustos e escaláveis de distribuição de chaves representam problema bastante conhecido, o uso de mecanismos semelhantes de confiança externos poderia ser aplicado nas ROCs [Ghodsi et al. 2011b]. Adicionalmente, uma vez que chaves criptográficas são utilizadas na formação dos nomes planos, os mesmos tornam-se não-amigáveis ao usuário final, sendo necessária a adoção de mecanismos externos adicionais, como serviços de busca e recomendação, para a resolução e mapeamento entre nomes e conteúdos em espaços de nomes privados de cada usuário [Koponen et al. 2007].

[4]Disponível em http://http://www.xorbin.com/tools/sha1-hash-calculator/.

A utilização de nomes planos traz consigo uma característica indesejável a qualquer identificador de objetos em rede que é a impossibilidade de agregação direta, o que conforma um grave problema de escalabilidade para os protocolos de roteamento. Não sendo agregáveis, é necessário dispor de uma entrada para cada conteúdo nas tabelas de encaminhamento e roteamento, fato que é visivelmente prejudicial à eficiência destes protocolos e suas implementações.

5.3.1.2. Nomeação Hierárquica

Estruturas hierárquicas para atribuição de nomes também foram propostas para utilização em ROCs. Através da concatenação de diferentes componentes hierárquicos de nome, identificadores únicos podem ser formados para atribuição a conteúdos. Em oposição aos nomes gerados em um sistema de nomeação plana, os nomes hierárquicos possuem uma característica semântica, uma vez que suas estruturas e cada um dos componentes que as compõem refletem alguma informação à respeito da natureza do conteúdo: propriedade, versão, formato etc. Dessa forma, estruturas semelhantes a identificadores uniformes de recursos (*Uniform Resource Identifiers* - URIs) [Mealling e Denenberg 2002] podem ser utilizados na representação de nomes hierárquicos.

Para obter conteúdo gerado dinamicamente é necessário que os usuários sejam capazes de construir, de forma determinística, os nomes dos dados desejados sem qualquer conhecimento prévio do nome ou conteúdo em si [Ghodsi et al. 2011b]. A utilização de nomes parciais e requisições relativas é um recurso que permite determinar sequências de nomes de forma simplificada, explorando as relações hierárquicas entre os componentes do nome. Um usuário pode requisitar o conteúdo `br.uff/video/intro.avi`, por exemplo, baseado na composição representada na Figura 5.3 e receber um pedaço (*chunk*) específico desse conteúdo, denominado `br.uff/video/intro.avi/1/1`. Num segundo instante, esse pedaço recebido pode ser utilizado para selecionar e requisitar outros pedaços do conteúdo de forma relativa ao primeiro segmento obtido, como por exemplo o pedaço 3, cujo nome é `br.uff/video/intro.avi/1/3`.

Figure 5.3. Nome hierárquico estruturado como URI.

Uma consequência direta da utilização de nomes hierárquicos é possibilidade de agregá-los através da utilização de mapeamento de prefixos longos, de forma análoga à agregação de rotas realizada pelos protocolos de roteamento IP. Dessa forma grande parte dos mecanismos já propostos para o tratamento de endereços IP podem ser adaptados para o tratamento de nomes hierárquicos, facilitando o processo de adoção gradativa das ROCs e reduzindo a carga sobre os protocolos de roteamento [Jacobson et al. 2009a]. Justamente por refletir propriedades dos conteúdos de forma estrita, os nomes hierárquicos não possuem uma característica forte de persistência. A semântica por trás destes nomes não permite o uso persistente dos mesmos, uma vez que qualquer mudança hierárquica,

como a transferência de propriedade ou de entidade publicadora do conteúdo deve ser refletida nos componentes do nome.

5.3.1.3. Nomeação por Atributos

Diferentemente dos demais esquemas de nomeação apresentados, nomeação por atributos não provê uma identificação estrita a cada um dos conteúdos. Pares de atributos, no formato `[atributo = valor]`, chamados de pares valor-atributo (*attribute-value pairs* - AVPs), são atribuídos aos conteúdos e tornam possível a sua identificação. Por exemplo, ao invés de ser requisitado através de um nome explícito, um conteúdo nomeado por atributos poderia apresentar a identificação `[classe = "alerta", severidade = 6, dispositivo = "web-server", tipo = "falha de hardware]"` e ser requisitado através das restrições `[severidade >2 ∧ classe = "alerta"]` [Carzaniga e Wolf 2003]. Aos conjuntos de restrições que podem ser utilizadas para identificação de conteúdos dá-se o nome de predicados [Carzaniga et al. 2000]. Há um mapeamento direto entre os predicados, seus conjuntos de restrições e os conteúdos por eles representados, ao qual dá-se o nome de cobertura. Determinado predicado cobre outro predicado se e somente se todos os conteúdos obtidos pelo último estão contidos no conjunto obtido pelo primeiro.

Consequência direta da cobertura de predicados de identificação é a possibilidade de agregação de nomes. Uma vez que as restrições que os definem são compostas apenas por operadores lógicos e AVPs, pode-se facilmente obter predicados agregados cuja cobertura inclui diversos subconjuntos de conteúdos, aliviando consideravelmente a carga sobre os protocolos de roteamento. Outra facilidade propiciada pela nomeação por atributos é possibilidade de realizar busca por conteúdo diretamente na rede, sem a necessidade de aplicações ou mecanismos externos para esse fim. Uma vez que os conteúdos não são nomeados explicitamente, pode-se especificar predicados que atendam a determinados interesses do usuário sendo o único objetivo verificar quais conteúdos seriam obtidos dessa forma.

O uso de pares de atributos e conjuntos de restrições para identificar conteúdos implica alguns problemas às ROCs. O primeiro deles é a dificuldade em expressar determinado conjunto de restrições mínimo necessário para a correta identificação do conteúdo, gerando problemas de unicidade. Uma vez que não consegue explicitar, exatamente, o conteúdo desejado, o usuário deve tratar o excesso ou falta de conteúdos disponibilizados após a solicitação, prejudicando o desempenho de suas aplicações. No caso de disponibilização de conteúdo em excesso há, ainda, uso ineficaz dos recursos de rede, que entregam conteúdos não-desejados.

5.3.2. Roteamento de Conteúdos

Diferentemente das redes centradas na comunicação entre estações, as ROCs devem ser capazes de entregar os conteúdos requisitados por nome sem qualquer informação referente à localização, tanto de usuários quanto de armazenamento de conteúdos. Para tal, os nós da ROC necessitam obter informações a respeito dos conteúdos existentes na rede a fim de encaminhar, da melhor forma possível, as requisições de conteúdo até cópias

dos conteúdos requisitados. O roteamento de conteúdos deve apresentar as seguintes características:

- **Orientação a conteúdo:** Endereçamento de pacotes através da utilização dos nomes de conteúdos, sem informações ou indicações de remetente e destinatário.
- **Robustez:** Tolerância a falhas e recuperação rápida em situações de descontinuidade, evitando encaminhamento de dados a nós falhos.
- **Eficiência:** Baixo impacto na quantidade de tráfego na rede devido a informações de controle.
- **Escalabilidade:** Permitir sua adoção em diferentes cenários, servindo a topologias de pequeno a grande porte.

Esse tipo de roteamento é denominado baseado em nomes e possui uma série de características particulares no que tange a forma como as informações de roteamento são trocadas entre os nós e como tais informações são armazenadas na rede. Os mecanismos podem ser divididos em dois grandes grupos: roteamento não-hierárquico e roteamento hierárquico.

5.3.2.1. Roteamento Não-Hierárquico

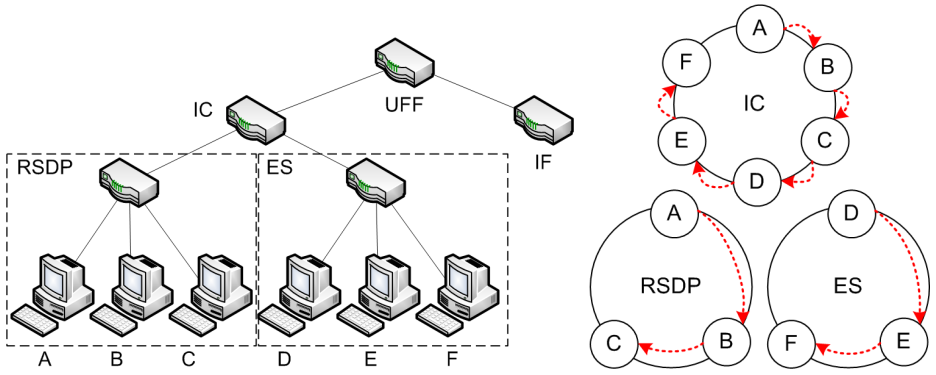
O roteamento não-hierárquico, ou não-estruturado, não apresenta estruturas dedicadas ao armazenamento de informações de roteamento ou estruturas hierárquicas para organização dos roteadores. Baseado no estabelecimento de enlaces sob demanda entre os nós, de acordo com necessidades instantâneas de entrega de dados, o roteamento não-hierárquico permite que todos os nós sejam capazes de obter conteúdos válidos. Como não há um nó para armazenamento centralizado das informações de roteamento e nem fluxo determinístico dos pacotes, as informações de roteamento devem ser difundidas entre os nós em escala global, permitindo que todos calculem as melhores rotas para entrega dos conteúdos, seja qual for o critério.

Esse tipo de roteamento torna possível a utilização de múltiplos caminhos, uma vez que o conhecimento do mapa completo da rede permite o cálculo de rotas livres de *loops*, além de aumentar a disponibilidade da rede como um todo, pois não existe um ponto único de falha. Os protocolos de roteamento tradicionalmente utilizados na Internet são, de modo geral, não-hierárquicos. Dessa forma, grande parte dos problemas encontrados para esses protocolos já foi identificada e soluções não-hierárquicas adotadas nas ROCs podem se aproveitar desse conhecimento prévio [Jacobson et al. 2009a, Carzaniga et al. 2004].

5.3.2.2. Roteamento Hierárquico

No roteamento hierárquico, ou estruturado, o mecanismo de roteamento assume que os roteadores da rede seguem uma estrutura hierárquica, garantindo um fluxo determinístico de informações de roteamento e de dados. Baseados na premissa de que os

roteadores são organizados em diversos níveis, os protocolos de roteamento hierárquicos são capazes de reduzir a quantidade de informações de controle trafegadas na rede, pois se aproveitam das relações hierárquicas existentes entre os roteadores. As ROCs apresentam, basicamente, duas propostas de roteamento hierárquico: baseado em árvores hierárquicas e baseado em tabelas *hash* distribuídas (*Distributed Hash Tables* - DHT). O uso de árvores hierárquicas como topologia de rede, como ilustra a Figura 5.4(a), implica a definição de relacionamentos entre os roteadores, dependente de suas posições na hierarquia. Conceitos como filiação, paridade, superioridade e inferioridade são intrínsecos às estruturas hierárquicas, os quais podem ser aplicados ao roteamento de conteúdo nomeado. Nós pais são ditos aqueles que possuem conexão com um ou mais nós filhos, configurando a raiz da subárvore à qual o nó filho pertence. Nós pares são todos os nós pertencentes ao mesmo nível hierárquico em relação a um nó raiz em comum. Concentrando as informações de roteamento em nós pais, raízes de subárvores, o agrupamento dos nós da rede em níveis hierárquicos permite que cada roteador seja capaz de encaminhar dados para elementos de seu nível de forma direta, através de rotas explícitas, sendo necessário recorrer a um elemento de nível hierárquico superior somente quando houver necessidade de encaminhamento para fora do nível [Koponen et al. 2007]. Essa característica permite agregar a carga de roteamento de toda a rede em nós pais, diminuindo a quantidade de informações utilizadas por cada nó no cálculo de rotas e, como consequência, reduzindo seus requisitos computacionais, como processamento e memória. Não é necessário que cada nó tenha um mapa completo da rede, sendo necessário armazenar somente as informações de roteamento dos nós filhos. Evidentemente, o nó pai concentra todas as informações de roteamento de seus nós filhos, consistindo em um ponto único de falha, podendo eventualmente causar a remoção de ramos inteiros da árvore de distribuição.



(a) Árvore hierárquica de domínios.

(b) DHTs específicas a cada domínio.

Figure 5.4. Árvore hierárquica e uma H-DHT sobreposta à topologia física.

As DHTs são estruturas adotadas para a distribuição de uma tabela de códigos *hash* criptográficos entre os nós participantes. A responsabilidade da manutenção do mapeamento entre valores e chaves é dividida entre os nós, formando uma estrutura plana para a distribuição uniforme de carga, garantindo proteção contra pontos únicos de falhas [Ganesan et al. 2004]. Os mecanismos baseados em DHTs hierárquicas (*Hierarchical*

Distributed Hash Tables - H-DHT) permitem, ainda, o arranjo dos nós em redes sobrepostas, encaminhando eficientemente mensagens em direção a determinadas chaves *hash*, sob responsabilidade de nós específicos.As estruturas de H-DHTs garantem que todos os nós em determinados domínios e subdomínios façam parte de uma DHT exclusiva, de modo que níveis hierárquicos superiores são compostos por fusões de níveis mais baixos. Por exemplo, na topologia apresentada na Figura 5.4(a), diversos níveis hierárquicos, ou subdomínios, compõem o domínio `UFF`. Os nós pertencentes a esta rede participam da cada um dos subdomínios existentes, como `RSDP` e `ES`, utilizando recursos à manutenção de cada uma das distintas DHTs, como na Figura 5.4(b). Assim, o nó A comunica-se tão somente com os nós B e C na DHT correspondente ao domínio RSDP, enquanto na DHT correspondente ao domínio IC é possível comunicar-se com os nós B, C, D, E e F.

A estrutura hierárquica de uma H-DHT confere isolamento de falhas ao mecanismo de roteamento uma vez que interações entre nós de um domínio não podem ser afetadas por falhas de nós externos. Adicionalmente, a distribuição de carga e funcionalidade entre os nós em uma DHT confere aos domínios uma tolerância a falhas internas aumentada, já que falhas em nós afetam apenas uma fração do espaço de chaves, permitindo uma rápida recuperação e redistribuição de chaves entre os nós participantes [Ganesan et al. 2004].

5.3.3. Armazenamento de Conteúdos (*Caching*)

Em semelhança às propostas de redes para distribuição de conteúdo na Internet apresentadas anteriormente, as ROCs também representam cenários promissores para a aplicação de técnicas de armazenamento de conteúdo nos elementos de rede (*in-network caching*). Baseado na característica de acesso a conteúdos na Internet, no qual uma pequena parcela de conteúdos populares contribuem com a maior parte do tráfego na rede [Breslau et al. 1999], a replicação de conteúdos e disponibilização dos mesmos por nós da rede mais próximos aos usuários implica um grande potencial de redução de tráfego e melhoria nos níveis de qualidade de serviço [Wang 1999]. Roteadores de conteúdo podem ter suas funcionalidades estendidas para prover uma infraestrutura distribuída de armazenamento, nos mesmos moldes das tradicionais CDNs. À medida que encaminha conteúdos a diferentes nós da rede, um roteador pode armazenar os conteúdos mais frequentemente acessados em memória, operando como um *cache* de rede [Jacobson et al. 2009a].

A abordagem para armazenamento de conteúdos nas ROCs diverge da adotada pelas soluções tradicionais de CDNs. Nas CDNs, além de avaliar a popularidade dos conteúdos através da quantidade de requisições feitas por usuários em escala global, os nós operam de forma orquestrada com um gerenciamento centralizado para otimizar a distribuição de réplicas e a utilização de recursos. O processo decisório de armazenamento de conteúdo nas ROCs baseia-se somente nas informações locais de conteúdo, isto é, os nós utilizam apenas as requisições e conteúdos em trânsito na determinação do armazenamento. Em essência, qualquer nó da rede, incluindo as estações de usuários, pode atuar como um *cache*, a qualquer instante, possibilitando estender as vantagens hoje proporcionadas por CDNs privadas a uma rede verdadeiramente pública e global de armazenamento e distribuição de conteúdos.

Apesar de não tratar a localização do conteúdo de forma direta, como visto anteriormente, a utilização de armazenamento em rede acaba por distribuir cópias dos conteúdos para nós distantes um dos outros, em termos de topologia de rede. Este cenário pode configurar um problema para os protocolos de roteamento uma vez que a agregação de rotas pode tornar-se uma questão bastante complexa, impactando a distribuição ótima de informações de roteamento.

5.3.4. Arquiteturas e Projetos em Desenvolvimento

Nesta seção algumas das principais arquiteturas propostas para as ROCs são apresentadas, dando-se enfoque à forma de aplicação dos conceitos apresentados anteriormente em cada uma das propostas, bem como os projetos que as implementam.

5.3.4.1. *Content-Based Networking/Combined Broadcast and Content-Based*

A arquitetura CBN(*Content-Based Networking*) [Carzaniga et al. 2000] é uma das propostas pioneiras para o desenvolvimento das ROCs. Fortemente baseada em conceitos provenientes dos sistemas de notificação de eventos *publish/subscribe*. CBN implementa uma arquitetura na qual conteúdos são publicados sem endereços explícitos de destinatários, entregando-os aos usuários com interesse de recebimento declarado através de *predicados*.

Em CBN cada nó anuncia à rede um predicado que define mensagens de interesse de recebimento, chamado de predicado de receptor (*receiver predicate*, ou *predicado-r*). Mensagens são identificadas por AVPs, apresentados na Seção 5.3.1.3, unicamente identificados por tipo, nome e valor. Por exemplo, um conjunto de AVPs válido seria `[string companhia = PET, int preço = 30]`. Predicados são usualmente representados através de uma conjuntos de restrições, ou filtros, sobre tais AVPs. O predicado `[string companhia = PET ∧ int preço <40]`, por exemplo, caracteriza um predicado válido correspondente à mensagem representada anteriormente. Além do *predicado-r*, pode-se divulgar um predicado de emissor (*sender predicate*, ou *predicado-s*), definindo mensagens que o usuário tem a intenção de enviar. Melhorou Um *predicado-r*, ao definir o interesse por mensagens de determinado nó, pode ser interpretado como seu endereço de rede baseado em conteúdo, já que estabelece o estado necessário na rede para o recebimento de conteúdos, estabelecendo o conceito de assinatura. Dessa forma, as restrições declaradas pelo *predicado-r* funcionam como filtro de mensagens publicadas e difundidas pela rede.

Sob uma ótica geral, o mecanismo de encaminhamento em CBN é organizado como uma estrutura de mapeamento entre atributos, restrições e interfaces do roteador. O mecanismo de encaminhamento implementa o fluxo representado na Figura 5.5, no qual todos os atributos encontrados nos predicados recebidos pelo roteador encontram-se à esquerda, na entrada do processo. Sobre tais atributos incidem as restrições impostas pelos predicados, conectadas às interfaces através de operadores booleanos. Tais operadores implementam as conjunções de restrições estabelecidas pelos predicados [Carzaniga e Wolf 2003], estabelecendo o estado necessário para o encaminhamento de mensagens pelas interfaces de saída. A construção da tabela de encaminhamento é propiciada pelo

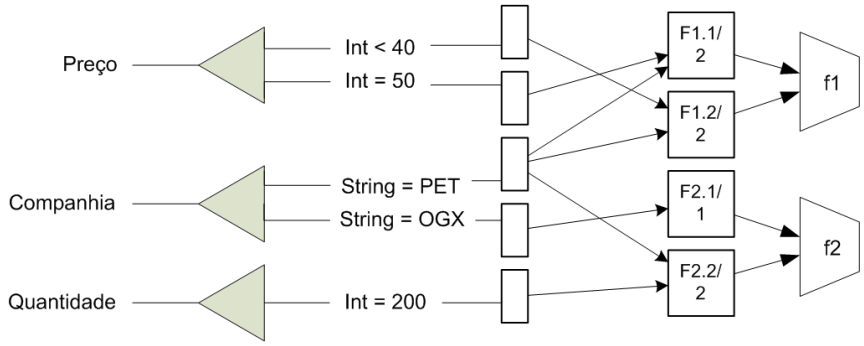


Figure 5.5. Mecanismo simplificado do encaminhamento [Carzaniga e Wolf 2003].

esquema de roteamento *Combined Broadcast and Content-Based* (CBCB), sobrepondo uma camada baseada em conteúdo sobre uma camada de difusão [Carzaniga et al. 2004]. A camada de difusão trata cada mensagem como uma mensagem para difusão global, enquanto a camada baseada em conteúdo poda dinamicamente os caminhos de distribuição, limitando a propagação de cada mensagem baseada nos predicados declarados. A camada de difusão garante que toda mensagem flua do nó emissor a todos os receptores através de caminhos sem laços e possivelmente mais curtos. Tal camada pode ser implementada utilizando-se mecanismos conhecidos para a construção de topologias sem laços, como mecanismos de *spanning trees*, árvores centradas em fontes (*per-source trees*) e outros mecanismos de difusão, como encaminhamento por caminho reverso (*reverse-path forwarding*).

As informações de roteamento em CBCB são propagadas através de dois mecanismos distintos: o envio de anúncios pelos receptores (*receiver advertisements* - RAs) e a requisição de emissores (*sender requests* - SRs). RAs são emitidos por todos os nós, periodicamente ou toda vez que o mesmo altera seu *predicado-r*. RAs transportam os novos predicados dos receptores, propagando tais informações a todos os potenciais emissores, estabelecendo os estados de encaminhamento necessários para a correta distribuição de mensagens aos nós de interesse. Toda vez que recebe um RA em dada interface, o roteador de conteúdos verifica se o endereço anunciado é coberto pelo predicado da interface receptora do RA. Em caso positivo, o RA e o filtro anunciado são descartados. Em caso negativo, o roteador computa o conjunto de interfaces pertencentes à árvore centrada no nó emissor do RA, encaminhando o RA a tais interfaces e estabelecendo caminhos para o fluxo de mensagens. Uma última etapa envolve a atualização da tabela de roteamento em que o filtro apontado pelo RA é incluído ao predicado da interface receptora através da conjunção lógica dos endereços.

Os SRs são utilizados pelos roteadores para buscar informações sobre todos os receptores a fim de atualizar sua tabela de roteamento. Ao receber um SR, os nós o respondem com *update replies* (URs), contendo todos os predicados de suas interfaces. O recebimento de um SR implica o encaminhamento imediato do mesmo a todas as interfaces participantes na árvore de difusão centrada no emissor do SR. Assim, os nós que replicaram o SR somente enviam o UR em resposta ao emissor original após o recebimento de todos os URs pelas interfaces utilizadas na retransmissão ou com o estouro de um contador de tempo. O protocolo permite, ainda, que os roteadores façam o armazena-

mento e reuso de URs em situações específicas, porém bastante comuns, como no cenário em que todos os nós receptores de determinada interface sejam os mesmos nós receptores da árvore de difusão centrada no nó emissor do SR original. Essa modificação permite reduzir a quantidade de tráfego de controle e de processamento nos roteadores.

As primeiras implementações da arquitetura, desenvolvidas para a avaliação de desempenho dos conceitos propostos em CBN [Carzaniga e Wolf 2003, Carzaniga et al. 2004], indicam resultados promissores dos mecanismos de encaminhamento e roteamento. O mecanismo de busca e encaminhamento de conteúdos apresenta desempenho aceitável mesmo na presença de milhões de restrições possíveis para aplicação. Os resultados experimentais obtidos com o protocolo de roteamento CBCB indicam sua capacidade em entregar, efetivamente, todo conteúdo de interesse dos usuários. A identificação de conteúdos através de AVPs, como visto na Seção 5.3.1.3, gera uma dificuldade de expressão de nomes de conteúdos com unicidade, dada a possibilidade de cobertura dos predicados. Dessa forma, as implementações apresentaram uma taxa de falsos positivos aceitável, que geram o *overhead* constante de aproximadamente 10% do tráfego de mensagens. Adicionalmente, propriedades interessantes foram observadas, como intensidade de tráfego de controle proporcional à taxa de mudança de predicados e diminuição dos requisitos de memória dos nós devido ao uso de armazenamento e reuso de URs, conferindo alguma escalabilidade à arquitetura.

5.3.4.2. *Data-Oriented Network Architecture*

DONA (*Data-Oriented Network Architecture*) [Koponen et al. 2007] é uma arquitetura baseada nos conceitos *clean-slate*[5] de nomeação e localização de conteúdos para a distribuição persistente e confiável dos mesmos em uma rede hierárquica. DONA propicia persistência e autenticidade através da utilização de nomes planos e autocertificadores. A alta disponibilidade é conferida pelo mecanismo de localização de conteúdos, guiando as requisições de conteúdos às cópias com menor custo de obtenção, evitando nós falhos ou sobrecarregados.

Em DONA todo nome é gerado por um outorgante, entidade associada a um par de chaves público-privadas as quais são utilizadas na identificação dos conteúdos. Essa associação é fundamental na formação dos nomes de conteúdos em DONA. Nomes apresentam o formato `P:L`, em que `P` representa o *hash* criptográfico da chave pública do outorgante e `L` é um rótulo arbitrário escolhido pelo mesmo, de modo que o nome de cada conteúdo seja único em seu domínio. Os outorgantes possuem papel de publicadores e administradores de conteúdo, uma vez que somente nós autorizados pela chave `P` podem prover acesso a objetos nomeados do tipo `P:L`. Uma vez que todo usuário, ao requisitar um conteúdo `P:L` irá receber como resposta o conteúdo composto pelos dados, chave pública de `P`, o rótulo `L`, metadados e uma assinatura do conteúdo [Ghodsi et al. 2011b], pode-se imediatamente checar a autenticidade do publicador dos dados verificando-se que o *hash* da chave pública é, de fato, `P` e que a mesma foi utilizada na assinatura do conteúdo. A utilização de nomes planos acarreta a dificuldade de associação dos nomes de

[5] O termo em inglês refere-se à proposição de ideias inovadoras, desconsiderando técnicas e/ou conceitos pré-concebidos e rompendo com as tecnologias atuais.

conteúdo pelos usuários. A arquitetura considera como premissa o fato de que usuários obtém nomes através de diversos mecanismos externos à rede, como sistemas de busca, comunicação privada, serviços de recomendação, e demais sistemas de recomendação e confiança utilizados pelos usuários.

O mecanismo de resolução de nomes, isto é, de roteamento de requisições de conteúdo nomeado, é implementado em nós denominados manipuladores de registros (*register handlers* - RHs), que implementam um protocolo bastante simples e eficaz. Pacotes `FIND(P:L)` são enviados ao RH local para localizar determinado objeto `P:L`, que por sua vez encaminha a solicitação às cópias mais próximas do conteúdo. Mensagens `REGISTER(P:L)`, por sua vez, estabelecem o estado necessário para que RHs encaminhem os pacotes FINDs de forma eficaz. Nós devidamente autorizados pelo outorgante também podem enviar pacotes `REGISTER(P:*)` a seu RH local. Dessa forma, independente do rótulo L utilizado, toda requisição de conteúdos sob reponsabilidade do outorgante `P` será encaminhada pelo RH local do nó que registrou `P:*`. Os RHs mantém entradas separadas na tabela de registros para `P:*` e `P:L`, mapeando de forma distinta os próximos saltos para cada uma das entradas. A existência de entradas na tabela de registros é crucial no encaminhamento dos pacotes `FIND` às cópias mais próximas do conteúdo. Não havendo uma entrada pertinente na tabela o RH encaminha o pacote `FIND` ao RH de nível hierárquico superior, eventualmente encontrado uma entrada válida na tabela de registros, uma vez que RHs de níveis hierárquicos superiores concentram as informações de roteamento de seus RH filhos (ou subdomínios), funcionando como uma espécie de *gateway* padrão. DONA, em semelhança à arquitetura apresentada em CBN, na Seção 5.3.4.1, não prevê o rompimento por completo com o IP. O pacote `FIND` é caracterizado por sua inserção entre os cabeçalhos IP e da camada de transporte, limitando-se à resolução dos endereços de conteúdos. Assim, os mecanismos convencionais de transporte são acionados para a realização da entrega dos conteúdos à partir de cópias armazenadas na rede, apenas orientando tais mecanismos a nomes, sem maiores mudanças nos protocolos e infraestrutura que os suportam.

A seleção automática de servidores, uma das funcionalidades desejadas em qualquer sistema de distribuição de conteúdos, pode ser obtida de forma nativa em DONA. RHs podem optar pelo encaminhamento de `FINDs` a vizinhos de menor custo, segundo a métrica de roteamento em DONA. *Multi-homing* e mobilidade são, também, características intrínsecas a DONA. `FINDs` podem ser encaminhados a mais de um RH por um nó *multi-homed*, resultando na utilização de múltiplos caminhos para obtenção de conteúdo. O protocolo de registro de conteúdos, baseado nas mensagens `REGISTER` e `UNREGISTER`, é o responsável por conferir mobilidade aos sistemas finais, já que antes da mudança de posicionamento do *host* na topologia de rede, o mesmo deve cancelar os registros de seus endereços de conteúdo e registrá-los novamente em sua nova localidade. Dessa forma, assim que os novos registros tiverem sido divulgados e estabelecido o estado de encaminhamento necessário, todos os `FINDs` serão roteados a essa nova localidade. A distribuição de conteúdo no formato *multicast* é, também, realizada de forma nativa já que a utilização de identificadores `P:L`, com o rótulo arbitrário `L` representando a identificação de um grupo *multicast* do tipo (S,G), permitem a criação de grupos centrados em fontes específicas, similar à implementação da comunicação *multicast* SSM (*Source-Specific Multicast*) [Bhattacharyya 2003, Holbrook e Cain 2006].

Algumas extensões opcionais a DONA foram propostas, intensificando seu impacto na distribuição de conteúdos. A utilização de *cache* nos RHs estendem suas funcionalidades, implementando uma infraestrutura genérica e sempre disponível nos caminhos de distribuição para o armazenamento de conteúdos, conferindo ganhos de qualidade de serviço no acesso de conteúdos. Mecanismos de assinatura de conteúdos e notificação de atualizações através de `FINDs` de longa duração, isto é, adicionados de TTLs (*time-to-live*). Enquanto o `FIND` for válido, atualizações pertinentes ao conteúdo desejado serão enviadas em direção ao *host* interessado. Outra funcionalidade desejável é a capacidade de se evitar servidores falhos ou sobrecarregados no provimento de conteúdos. Ao enviar `REGISTERs` à rede, tais nós podem incluir informações a respeito de carga e processamento, facilitando a tomada de decisões relativas a encaminhamento de `FINDs` pelos RHs.

5.3.4.3. *Content-Centric Networking/Named-Data Networking*

CCN (*Content-Centric Networking*) [Jacobson et al. 2009a] é uma arquitetura de ROC baseada na utilização do conteúdo como objeto elementar da rede, tratando questões como alta disponibilidade e segurança de conteúdos, independente da localização. CCN, em analogia às propostas já apresentadas, preserva alguns dos conceitos do TCP/IP que o tornaram simples, robusto e escalável, estendendo-os para prover uma camada de rede flexível, com poucas restrições à camada de enlace.

Uma das principais características em CCN é a divisão dos conteúdos em pedaços (*chunks*), estruturas nomeadas com identificadores únicos e hierárquicos, requisitados de forma individual. Os nomes são compostos por variável número de componentes, exatamente como apresentado na Seção 5.3.1.2. Cada componente é formado por um número arbitrário de octetos, sem qualquer significado à camada de transporte, podendo ser, inclusive, criptografados. Uma vantagem direta da utilização de nomes hierárquicos é a possibilidade de agregar nomes fazendo-se referência aos nós raízes da árvore hierárquica. A estruturação do nome em forma hierárquica permite, ainda, a expressão do posicionamento dentro da estrutura hierárquica, definindo inclusive o posicionamento relativo a outros nós da árvore de nomes. Em uma árvore de nomes, como a representada na Figura 5.3, pode-se requisitar conteúdo por relacionamentos. Caso um usuário queira a versão anterior do conteúdo, por exemplo, o usuário pode solicitá-la através do identifcador `br.uff/video/intro.avi/1/anterior`. Se quer o próximo pedaço, pode solicitá-lo usando `br.uff/video/intro.avi/1/1/posterior`. Dados satisfazem ao interesse declarado à rede caso o nome do conteúdo no pacote de interesse for um prefixo do nome do pacote de dados. Isso equivale a dizer que o pacote de dados está na subárvore de nomes especificados pelo pacote de interesse.

A arquitetura CCN é baseada em duas primitivas básicas: a declaração do interesse por determinado *chunk* e o envio deste em resposta ao interesse. Usuários requisitam conteúdo diretamente à rede difundindo seu interesse por determinado *chunk*, na forma de pacote de interesse (*interest packet*, ou *I-packet*), em todas as interfaces disponíveis. Nós vizinhos respondem ao *I-packet* enviando como resposta o pacote de dados (*data packet*, ou *D-packet*) caso o tenham armazenado em memória. Caso contrário encaminham o

I-packet aos seus vizinhos até que, eventualmente, o interesse encontre um nó com o dado armazenado. Dados são somente enviados como resposta a interesses, consumindo o interesse pendente equivalente em cada nó no caminho reverso, estabelecendo uma espécie de balanço entre requisições e atendimento.

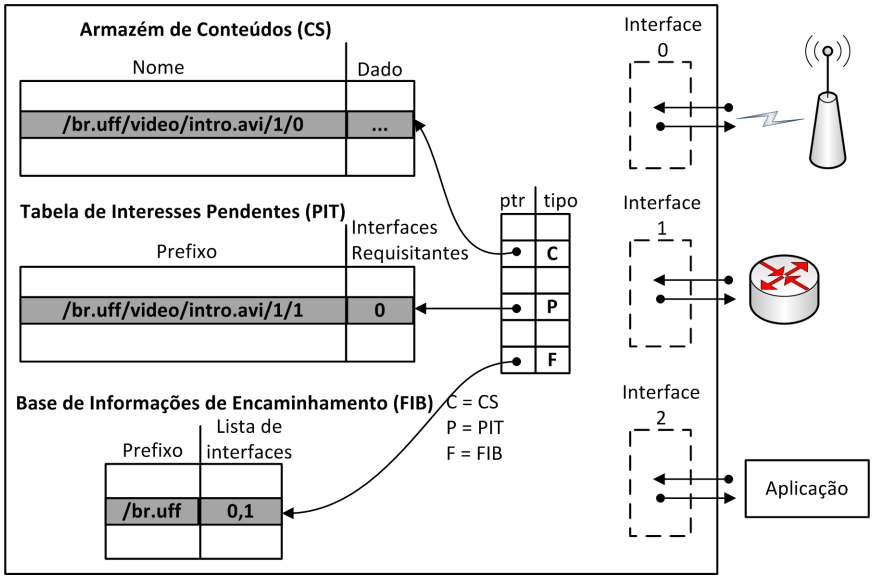


Figure 5.6. Nó CCN e estruturas do mecanismo de encaminhamento [Jacobson et al. 2009a].

O encaminhamento de pacotes realizado pelos nós em CCN é fortemente derivado do IP, consistindo do básico mapeamento entre nome do conteúdo e interface associada à árvore de distribuição do pacote, com algumas características especiais. Cada roteador utiliza três estruturas distintas nas operações de encaminhamento de pacotes: o CS (*Content Store*), a PIT (*Pending Interest Table*) e a FIB (*Forwarding Information Base*), ilustrados na Figura 5.6. A FIB é uma base de dados utilizada no armazenamento de informações de encaminhamento de pacotes, realizando o mapeamento entre nomes de conteúdo e uma ou mais interfaces para encaminhamento, permitindo a utilização de múltiplas fontes de forma nativa. CS é a estrutura de *cache* do roteador em CCN, área de armazenamento de *chunks* pelo tempo mais longo possível, utilizando políticas de atualização de *cache* similares a LFU ou LRU [Podlipnig e Böszörmenyi 2003]. A PIT é uma tabela na qual são armazenados os interesses encaminhados adiante, registrando a interface de origem para que os dados enviados como resposta possam ser encaminhados em direção ao solicitante. Ao receber um *I-packet*, o roteador primeiro checa em seu CS a existência de uma entrada para o nome requisitado. Em caso positivo, envia como resposta o *D-packet* relativo. No caso de não haver uma entrada armazenada no CS o roteador verifica se já há um interesse pendente na PIT. Em caso positivo, a interface de recebimento do *I-packet* é adicionada a lista de interfaces para encaminhamento do conteúdo na PIT e o *I-packet* é descartado. Caso não haja entrada na PIT, o roteador encaminha o pacote de acordo com as regras de sua FIB, criando na PIT um registro da interface de origem. Caso não haja entrada na FIB para determinado conteúdo é realizado o descarte do interesse, uma vez que não há rota válida para o mesmo. Esse encaminhamento difuso visa encontrar, eventualmente, um nó que possa atender à solicitação e enviar o pacote de dados no caminho reverso, sinalizado

pelas entradas das PITs. Somente uma entrada na PIT valida a admissão de *D-packet* pelo roteador, com todos os outros cenários levando ao descarte do pacote. É necessário que as fontes de dados registrem sua intenção de prover determinados conteúdos, através de uma primitiva de registro (*register*), criando o estado de encaminhamento inicial necessário para o envio de interesses às fontes.

A camada de estratégia implementa o mecanismo decisório de encaminhamento de pacotes que atua na FIB, determinando dinamicamente a forma como um roteador encaminha os pacotes de interesse. Diferentemente do TCP, em CCN cabe à camada de estratégia do receptor requisitar conteúdos não entregues ou corrompidos. O controle de fluxo também é implementado pela camada de estratégia, uma vez que o envio de múltiplos pacotes de interesses em paralelo endereçados a *chunks* sequenciais possui função equivalente à janela TCP, controlando a quantidade de tráfego que pode ser inserida na rede pelas fontes de dados.

Dadas as características de nomeação e encaminhamento presentes em CCN, qualquer esquema de roteamento por estado de enlace válido para IP possui uso potencial em CCN. O mecanismo de encaminhamento de CCN não impõem restrições quanto ao uso de múltiplas fontes ou destinos, uma vez que a utilização da PIT impede a formação de laços na rede. A utilização de nomes hierárquicos, com semântica semelhante à dos endereços IP, confere agregabilidade aos endereços de rede em CCN, aplicando-se o mesmos mecanismo de casamento do maior prefixo (*longest prefix match*).

CCN aplica conceitos de segurança diretamente aos conteúdos, independente dos mecanismos de segurança adotados pelos meios de transporte. A autenticação do vínculo entre nome e dados é obtido pela assinatura do publicador do conteúdo sobre o nome e os dados do conteúdo. O vínculo entre nome e conteúdo permite que publicadores atribuam nomes arbitrários às suas publicações e as torna facilmente autenticáveis, pois qualquer nó da rede pode avaliar se o vínculo entre nome e conteúdo foi assinado por determinada chave. A determinação do mecanismo de autenticação de vínculo pode variar entre diferentes conjuntos de publicadores e usuários, criando uma flexibilidade de adequação dos recursos computacionais de acordo com a necessidade de cada aplicação. Pode-se, ainda, distribuir a carga computacional de autenticação entre vários pacotes, apesar de pacotes serem pensados como autenticáveis individualmente. A validação do vínculo é simplesmente sintática, isto é, valida-se que a chave foi utilizada na assinatura do conteúdo sem inserir nenhum significado a ela, como propriedade ou critérios para confiança na chave.

A arquitetura proposta em CCN é a base para o desenvolvimento do projeto NDN (*Named Data Networking*) [Zhang et al. 2010]. O projeto NDN visa desenvolver as técnicas complementares necessárias à plena adoção das ROCs, abordadas na proposição da CCN. Questões como o roteamento global, o encaminhamento eficiente de conteúdo nomeado, o desenvolvimento de aplicações, as técnicas de segurança e privacidade, entre outras, fazem parte da agenda de pesquisa do projeto. O projeto NDN possui um *testbed* com 11 nós distribuídos pelos principais centros de pesquisas dos Estados Unidos. Usuários conectam-se ao *testbed* via túneis UDP, estendendo o alcance da ROC a todo o *campus* e às redes domiciliares dos usuários.

5.3.4.4. *Publish-Subscribe Internet Routing Paradigm/Publish-Subscribe Internet Technologies*

O projeto *Publish-Subscribe Internet Routing Paradigm* (PSIRP) [Lagutin et al. 2010], de encontro às demais propostas de ROCs, especifica uma arquitetura de ROC sem aplicar tecnologias de conectividade e transporte existentes, como o TCP/IP. Fortemente baseada nas primitivas de redes *publish/subscribe*, PSIRP define as publicações (como são chamados os conteúdos) como associações persistentes entre identificadores e dados criados pelo publicador. Identificadores autocertificáveis são utilizados neste vínculo entre nome e conteúdo, na forma de *hashes*, uma vez que cada publicação possui um único publicador lógico.

PSIRP utiliza o conceito de *rendezvous*[6] [Visala et al. 2009] para implementar a resolução de nomes de conteúdos. Publicadores anunciam conteúdo em redes de *rendezvous* locais, que realizam a associação de fontes de dados e assinantes interessados no conteúdo armazenado. Publicadores anunciam os escopos (*Scope ID* - S_{id}), identificadores relacionados aos conteúdos que autorizam sua distribuição por outras fontes de dados. Fontes de dados são nós de armazenamento, localizadas nas extremidades da rede que utilizam o sistema de *rendezvous* para a divulgação dos conteúdos existentes, o identificador de *rendezvous* (*Rendezvous ID* - R_{id}). Dessa forma, todo conteúdo deve ser solicitado através do uso da dupla de identificadores S_{id}, que direciona a forma de distribuição autorizando determinadas fontes, e R_{id}, indicando a publicação desejada neste escopo. S_{id} e R_{id} utilizam pares de identificadores `"P:L"` em que `P` é a chave pública do proprietário do espaço de nomes e `L` é um rótulo arbitrário da publicação. As redes locais de *rendezvous* são interligadas através da interconexão de *rendezvous* (*Rendezvous Interconnect* - RI), uma DHT hierárquica com presença em todos os domínios da arquitetura PSIRP, que permite resolver S_{id}s e R_{id}s de conteúdos disponíveis em domínios distintos.

A resolução dos identificadores de uma publicação na RI devolve ao usuário a indicação de nós de ramificação (*Branching Nodes* - BNs) da árvore de distribuição relativa à publicação. A requisição de assinatura da publicação é, então, encaminhada através dos diversos domínios de PSIRP, utilizando rotas para os BNs corretos, como na Figura 5.7. O sistema de roteamento é responsável pela determinação e manutenção da árvore de entrega de cada publicação e pelo processo de armazenamento de conteúdos populares na rede. Os BNs selecionam as rotas para novas assinaturas baseados nas informações de topologia da rede e de métricas obtidas a partir de medidas de intensidade de tráfego, obtidas por um sistema distribuído de gerenciamento topológico, além de gerenciar grandes *caches* e realizar a ramificação das árvores de distribuição quando há múltiplas requisições, semelhante ao *multicast*.

O sistema de encaminhamento [Jokela et al. 2009] encarrega-se da entrega das publicações aos assinantes através de árvores de entrega eficientes. Os nós de encaminhamento (*Forwarding Nodes* - FNs) entregam pacotes através do mapeamento da árvore de distribuição da publicação e um identificador de encaminhamento (*Forwarding ID* - F_{id}) e utilizam de tais F_{id}s no encaminhamento. F_{id}s são identificadores baseados em filtros de *Bloom* construídos pelo sistema, numa espécie de roteamento pela fonte. Uma vez

[6]Termo em francês para a palavra “encontro”.

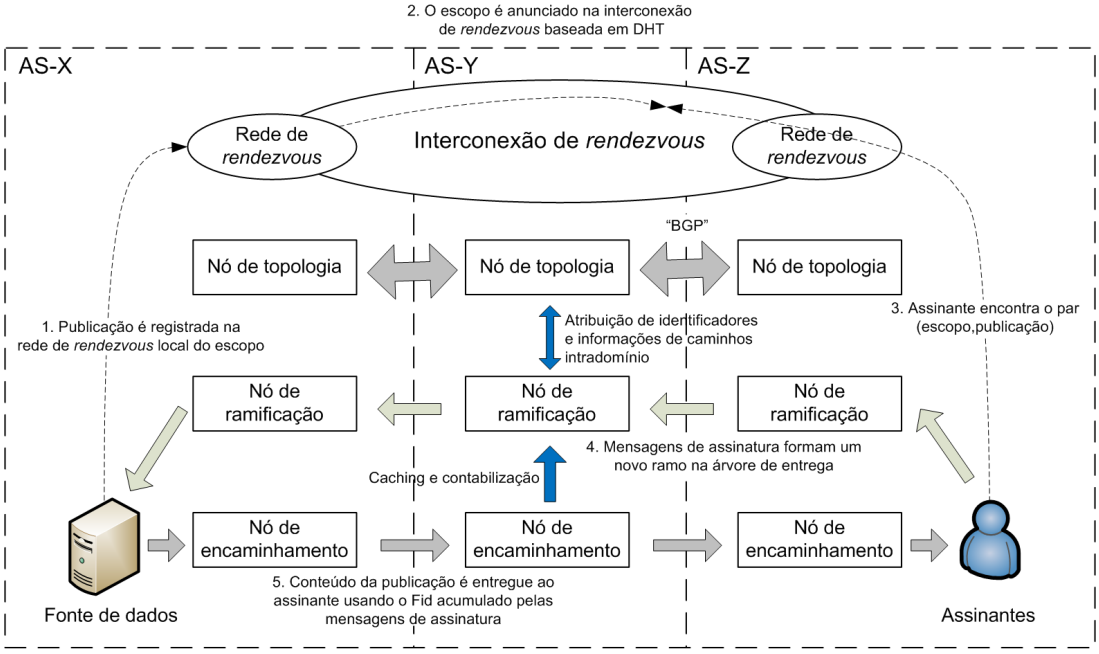


Figure 5.7. Arquitetura simplificada adotada em PSIRP [Lagutin et al. 2010].

que todos os enlaces possuem uma identificação (*Link ID*), também um filtro de *Bloom*, é possível codificar a árvore de entrega através de um filtro e utilizá-lo como F_{id} para encaminhamento. Cada FN na árvore de entrega do conteúdo realiza uma simples operação lógica `AND` no *link* ID de saída e no filtro de *Bloom* no cabeçalho do pacote. Se o resultado da operação for o próprio *link* ID, assume-se que o mesmo faz parte da árvore de entrega e o pacote é encaminhado pela interface correspondente. Esse mecanismo é sujeito a falsos positivos na identificação de interfaces de encaminhamento, causando o envio do pacote a nós que, na verdade, não participam da árvore. Um exemplo deste mecanismo é representado na Figura 5.8.

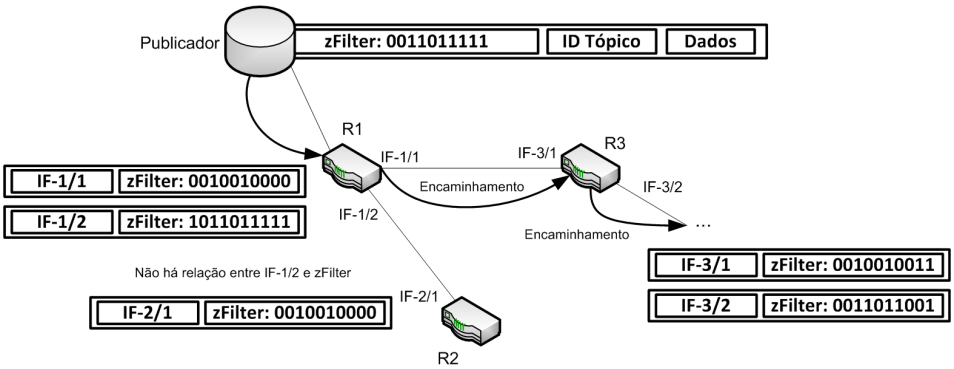


Figure 5.8. Exemplo do encaminhamento de dados adotado em PSIRP [Jokela et al. 2009].

Baseado nos conceitos levantados pela proposição da arquitetura PSIRP, o projeto PURSUIT (*Publish / Subscribe Internet Technologies*) [Fotiou et al. 2010] foi criado com o objetivo de explorar e refinar ainda mais a proposta da PSIRP. Mobilidade, pri-

vacidade, armazenamento em rede e contabilização são temas de grande importância no desenvolvimento do projeto. Um dos principais objetivos do projeto PURSUIT é o desenvolvimento de soluções e mecanismos aplicáveis à criação e oferta de serviços inovadores, aproveitando todo o potencial dessas novas estruturas de informação. A identificação algorítmica de conteúdos é um conceito explorado em PURSUIT através de novas técnicas de fragmentação do conteúdo e armazenamento distribuído eficiente. A natureza da arquitetura em PURUSIT, recursiva e hierárquica, permite gerenciar mobilidade de forma nativa, orientada ao nó móvel, de forma transparente ao núcleo da rede. Assim, suporte a mobilidade é um dos tópicos de interesse no desenvolvimento do projeto. Mecanismos de autenticação e contabilização devem ser estendidos para suportar os sistemas de PURSUIT. Questões relacionadas ao gerenciamento de politicas de topologia, roteamento e encaminhamento, em todos os níveis da arquitetura, requerem o desenvolvimento de uma plataforma de gerenciamento global. O projeto PURSUIT possui um *testbed* com servidores dedicados localizados em diversas instituições americanas e europeias, executando mais de 25 nós da rede PURSUIT simultaneamente.

5.3.4.5. Demais arquiteturas

Além das arquiteturas apresentadas, outras também foram propostas visando a implementação das ROCs. TRIAD [Cheriton e Gritter 2000], NetInf [Ahlgren et al. 2008] e MultiCache [Katsaros et al. 2011] são bons exemplos. TRIAD foi a arquitetura pioneira a propor a comunicação baseada em conteúdos, utilizando as URLs presentes nas requisições HTTP como nome de conteúdos. TRIAD realiza o redirecionamento das requisições para cópias mais próximas, conceitualmente bastante próxima das CDNs, implementado a camada de conteúdo e armazenamento em todos os roteadores de conteúdo. NetInf, proposta pelo projeto 4WARD[7], utiliza conceitos em comum com DONA e PSIRP/PURSUIT, como nomes planos e roteamento baseado em DHT. NetInf propicia o uso de identificadores persistentes, como apresentado na Seção 5.4.1 e a separação entre publicação e dados, permitindo a utilização de múltiplas versões da mesma publicação. MultiCache explora a comunicação multidestinatária como meio de distribuição de conteúdo, implementando uma rede sobreposta baseada em Pastry [Rowstron e Druschel 2001] para o armazenamento de conteúdo em rede, localização de réplicas na rede e distribuição do conteúdo em si.

5.3.4.6. Comparativo Geral

A Tabela comparativa 5.1 apresenta as principais características de cada uma das arquiteturas apresentadas. CBN, por utilizar AVPs na nomeação de conteúdos, permite a busca em rede e assinatura prévia de conteúdos ainda não publicados. Porém, como todos os atributos existentes podem ou não ser utilizados na identificação de conteúdos, as tabelas de roteamento acabam por sofrer impacto considerável, da ordem de 2^A, onde A representa a quantidade total de atributos, diretamente relacionada com expressividade. Adicionalmente, todas as mudanças de predicados não cobertos devem ser difundidas por

[7] http://www.4ward-project.eu/

Table 5.1. Tabela comparativa geral.

Característica	CBN	DONA	CCN/NDN	PSIRP/PURSUIT
Nomeação Plana		X		X
Nomeação Hierárquica			X	
Nomeação por AVPs	X			
Roteamento Estruturado		X		X
Roteamento Não-estruturado	X		X	
Caching em rede		X	X	X
Segurança intrínseca de conteúdos		X		X

toda a rede, impactando consideravelmente na quantidade de tráfego de controle.

Em DONA, por sua vez, utiliza-se nomeação plana de conteúdos, o que confere persistência e unicidade de identificadores. Como a arquitetura utiliza uma estrutura hierárquica de roteamento, definindo fluxos sistemáticos e centralização das informações de roteamento, o impacto do tráfego de controle na rede não tende a ser significativo. O uso de identificadores não-agregáveis, porém, implica na utilização de uma entrada por conteúdo nas tabelas de roteamento, caracterizando um ponto crítico no que tange a escalabilidade.

Já os projetos CCN/NDN são baseados em estruturas hierárquicas de nomeação, conferindo agregabilidade aos nomes de conteúdo. O mecanismo de difusão de interesses, porém, impacta consideravelmente na quantidade de tráfego de controle na rede. O uso de *cache*, ainda, distribui os conteúdos mais próximos da borda da rede, dificultando o processo de agregação de rotas. Ao adotar princípios de *design* semelhantes ao IP, CCN/NDN permitem a adoção de mecanismos legados, comprovadamente funcionais, para uma adoção gradual e eventual substituição de tecnologia.

PSIRP/PURSUIT, em semelhança a DONA, nomeia seus conteúdos com identificadores planos, conferindo as mesmas propriedades de persistência e unicidade. A estrutura de DHT hierárquica do sistema de *rendezvous* permite distribuir a carga de roteamento entre todos os domínios participantes, normalizando os requisitos computacionais dos nós da rede. Porém ao romper totalmente com os conceitos provenientes do IP, PSIRP/PURSUIT institui uma arquitetura *clean slate*, dificultando sua adoção em escala global.

5.4. Desafios

A pesquisa relativa às ROCs, apesar de bastante recente, apresentou uma série de avanços e proporcionou uma gama de soluções para diferentes questões. Porém, ainda há diversos problemas em aberto e aspectos práticos de implementação que necessitam de definições mais bem estudadas e detalhadas. Esta seção discute alguns desses desafios e indica possíveis rumos para os quais o desenvolvimento das ROCs pode ir ao encontro.

5.4.1. Nomeação de Conteúdos

Todos os conceitos propostos para as ROCs tornaram-se viáveis graças a uma premissa básica: a utilização de conteúdos nomeados como primitiva básica de rede. Por ser

o alicerce de todas as arquiteturas vistas na Seção 5.3.4, o desenvolvimento dos mecanismos de identificação ou nomeação de conteúdos representam um grande desafio para a adoção das ROCs. Mecanismos distintos para nomeação de conteúdos foram propostos, como visto na Seção 5.3.1, podendo ser agrupados 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, como persistência, escalabilidade e inteligibilidade ao usuário final. É o caso, por exemplo, da nomeação hierárquica que é escalável, mas possui restrições ao uso persistente dos nomes. Existem, ainda, questões relativas à segurança das ROCs que permeiam a nomeação dos conteúdos, porém tais aspectos são tratados na Seção 5.4.4.

A principal crítica feita à nomeação plana é o fato de que a ausência de hierarquia acarreta problemas de escalabilidade. Ghodsi *et al.*, entretanto, alegam que a falta de hierarquia dos nomes planos não impede a agregação explícita de nomes [Ghodsi et al. 2011b]. Utilizando-se de nomes planos únicos, no já citado formato `P:L`, pode-se formar identificadores agregados através da concatenação de nomes na forma $\texttt{nome}_1\texttt{.nome}_2\texttt{.nome}_3\texttt{...nome}_n$, em que cada componente $\texttt{nome}_i$ é um nome plano individual. Dessa forma, não são necessárias quaisquer mudanças nas tabelas de roteamento. As tabelas possuem entradas individuais para cada um dos nomes, possibilitando o encaminhamento individual de cada conteúdo pelo roteador. Porém, quando confrontado com concatenações de nomes, o roteador realiza o encaminhamento baseado no casamento mais específico (*deepest match*). Assim, a busca por entradas casadas na tabela de roteamento inicia-se pelo nome de nível mais baixo, da direita para a esquerda. Suponha um nome concatenado na forma `A.B.C`, no qual `A`, `B` e `C` são nomes planos individuais. Nesse caso, a primeira entrada a ser buscada é relativa ao nome `C`). Não havendo uma rota específica para `C`, o mecanismo executa o mesmo procedimento para `B` e, em última caso, utiliza-se a rota de nível mais alto para `A`. De forma análoga aos nomes hierárquicos, a concatenação de nomes planos pode ser utilizada para representar conteúdo estruturado. Por exemplo, no identificador agregado `A.B.C`, `A` pode representar todo o conteúdo agregado de determinado publicador, `B` pode representar um grande arquivo e `C`, por sua vez, um determinado pedaço (*chunk*) que o compõe. A agregação explícita de nomes planos confere uma característica de "hierarquia virtual" ao mecanismo de nomeação plana. Ela também permite que um mesmo nome seja agregado de diversas formas distintas ao contrário da agregação característica da nomeação hierárquica que permite somente a agregação de nomes em prefixos de níveis hierárquicos maiores.

Se o desafio para mecanismos de nomeação plana é ser capaz de agregar nomes, o desafio para os mecanismos de nomeação hierárquica é garantir a persistência dos nomes. A capacidade de agregação intrínseca dos nomes hierárquicos tanto contribui para a escalabilidade quanto dificulta o uso persistente de nomes nas arquiteturas baseadas em nomes hierárquicas. A escalabilidade é proporcionada pela potencial redução do tamanho e dos tempos de atualização das tabelas de roteamento. Também há unicidade de prefixos, garantindo o roteamento por prefixos através de mecanismos de casamento do maior prefixo, como o usado por protocolos de roteamento da Internet atual. A persistência é prejudicada justamente porque os nomes hierárquicos refletem propriedades dos conteúdos de forma explícita. Assim, qualquer mudança hierárquica como, por exemplo, transferência de propriedade ou de entidade publicadora do conteúdo, deve ser refletida nos compo-

nentes do nome.

Os nomes planos, no formato `P:L`, em geral, também não conferem persistência plena aos nomes, pois a chave pública do publicador é utilizada na geração do nome. Dessa forma, o nome está atrelado à propriedade do conteúdo. Assim, uma mudança de proprietário implica mudança da chave `P` e consequentemente mudança de nome do conteúdo. A arquitetura NetInf especifica uma solução para a utilização de nomes planos efetivamente persistentes. Para tanto, a NetInf desatrela os nomes do conceito de propriedade [Dannewitz et al. 2010]. São propostas duas abordagens para prover tal independência. Na abordagem mais simples, os nomes planos, diferentemente do apresentado na Seção 5.3.1.1, possuem a componente de *hash* constituída por uma chave pública PK_{IO}, atrelada ao conteúdo em si e não a um publicador ou outorgante, resultando no identificador $\mathtt{PK}_{IO}\mathtt{:L}$. Sempre que houver uma mudança sobre a posse do conteúdo, a chave secreta SK_{IO} deve ser passada ao novo proprietário através de um canal seguro, permitindo que as publicações sejam assinadas utilizando a mesma chave. A abordagem mais complexa prevê a utilização de novos pares PK/SK sempre que o proprietário do conteúdo for alterado. Para que o nome continue válido, isto é, mantenha sua persistência, o *hash* da chave pública original PK_{IO} permanece inalterado no nome do conteúdo. Porém, o novo proprietário assina o conteúdo com sua chave privada SK_{latest}, inserindo a informação em metadados verificáveis através de PK_{latest}. O par $\mathrm{PK}_{latest}/\mathrm{SK}_{latest}$ é autorizado pelo par de chaves originais $\mathrm{PK}_{IO}/\mathrm{SK}_{IO}$ através de uma cadeia de certificados [Clarke et al. 2001]. A cadeia de certificados consiste da autorização de um par de chaves público-privada através da utilização de um certificado pai. A relação entre os diversos níveis hierárquicos de certificados estabelece uma cadeia de confiança, denominada caminho de certificação. Para garantir a validade da assinatura digital inserida nos metadados e da autenticação do vinculo entre PK_{latest} e SK_{IO}, todo o caminho de certificação deve ser verificado para a obtenção do certificado raiz original, assinado por PK_{IO}.

Existe um consenso em relação às características desejáveis dos mecanismos de nomeação, mais especificamente em relação à unicidade, à autocertificação e à inteligibilidade pelos usuários, informalmente conhecido como Triângulo de Zooko[8]. Embora não haja uma prova formal, tal consenso afirma que qualquer mecanismo de nomeação pode apresentar até duas das três características citadas, porém não as três simultaneamente. Os conceitos apresentados na Seção 5.3.1 corroboram tal visão, uma vez que nenhum dos mecanismos de nomeação aplicados às ROCs apresentam todas as características simultaneamente. Até mesmo os nomes hierárquicos, que eventualmente espelham as características das URLs, podem ser baseados em componentes não-amigáveis aos usuários. O uso de nomes não-amigáveis implica o uso de mecanismos externos para a resolução e obtenção de identificadores, como mecanismos de busca e redes de recomendação [Koponen et al. 2007].Como alternativa aos mecanismos centralizados para resolução de endereços, são propostos mecanismos de nomeação pessoal, atuando como abstrações amigáveis aos diversos mecanismos de nomeação utilizados na Internet, como o *Pnames* [Allman 2007]. Allman afirma que a utilização de um espaço de nomes pessoal permite a identificação de recursos através de apelidos locais, amigáveis ao usuário, sem maiores requisitos quanto à unicidade. Nomes podem ter distintos sig-

[8]Disponível em `ttp://www.zooko.com/distnames.tml`

nificados em cada um dos espaços de nomes, uma vez que são diretamente administrados pelos usuários. Tais espaços inserem uma camada de abstração de nomes, cujo mecanismo básico é o mapeamento de identificadores de recursos de rede em espaços de nomes distintos, próprios ou não, dependendo do contexto. O compartilhamento de porções dos espaços de nomes pode ser direto, através de trocas entre indivíduos, ou através de uma base centralizada que é aberta a consultas e edição das entradas públicas e privadas e que pode ser acessada por todos os usuários. Outra abordagem que visa a não-utilização de mecanismos externos de resolução é a utilização de nomes hierárquicos atribuídos pelos provedores de acesso [Zhang et al. 2010], muito semelhante à atribuição de endereços IP, porém sem a limitação no tamanho dos identificadores. Dessa forma, um recurso disponibilizado por um usuário atendido por determinado Provedor A poderia ser nomeado `/provedor/users/usuario/conteudo`. Apesar de facilitar a administração dos provedores e de aumentar o potencial de agregação de rotas dados os identificadores do provedor, tal solução tende a atrelar os identificadores à localização dos recursos. Dessa forma, a resolução de nomes nas ROCs é, ainda, um desafio bastante importante a ser estudado visto que não há consenso sobre qual é a abordagem com mais pontos positivos do que negativos.

5.4.2. Roteamento de Conteúdos

Outro importante desafio no desenvolvimento das ROCs é o roteamento de conteúdos baseado em nomes. Esse tipo de roteamento rompe com o atual paradigma na Internet, no qual o caminho mais curto a ser percorrido por uma pacote é determinado pelo endereço IP de destino que ele carrega [Saltzer et al. 1984]. Por utilizar nomes como identificadores em nível de rede a invés do endereços IP, os protocolos de roteamento de conteúdos baseados em nomes são suscetíveis às características particulares dos diferentes mecanismos de nomeação de conteúdos. A utilização de mecanismos de nomeação não agregáveis, por exemplo, configura um grave problema de escalabilidade para os protocolos de roteamento. Não sendo agregáveis, o impacto da manutenção de diversas entradas para conteúdos distintos nas tabelas de roteamento é visivelmente prejudicial à eficiência desses protocolos e, no pior caso, pode levar à explosão das tabelas de roteamento. Além do impacto direto das diversas técnicas de nomeação, o roteamento apresenta desafios intrínsecos às diversas arquiteturas discutidas. Por ser um tema de pesquisa recente, muitos dos desafios relacionados ao roteamento baseado em nomes encontram-se ainda sem tratamento adequado, de modo que a literatura disponível é, ainda, limitada. Assim, muitas das abordagens ilustradas nesta seção tratam de abordagens utilizadas em outros cenários, podendo ser facilmente adaptadas às ROCs.

O roteamento baseado em nomes quebra a semântica sobrecarregada do IP, ou seja, separa-se a localização física da identificação das estações que na arquitetura atual é feita por um único endereço IP [Campista et al. 2010] . Essa dissociação entre localizador e identificador facilita a mobilidade das estações. Na arquitetura atual da Internet, quando uma estação muda a sua localização, ela necessariamente deve mudar seu endereço IP se muda de rede. Assume-se, porém, que os endereços IP não se alteram durante toda a comunicação entre duas estações. Se há mudança de endereço, perdem-se as comunicações estabelecidas com o endereço IP anterior. Nas ROCs esse problema não ocorre visto que os nomes, em geral, não carregam informações sobre a localização de quem os publica.

Por isso, diz-se que o problema da mobilidade é simplificado nas ROCs.

Anteriormente às ROCs, algumas propostas de roteamento baseado em nomes surgiram, principalmente para resolver o problema da mobilidade de estações descrito anteriormente. Alguns exemplos são o TRIAD [Cheriton e Gritter 2000], que utiliza as URLs enviadas nas requisições HTTP como nomes hierárquicos de conteúdo, o ROFL [Caesar et al. 2006], que utiliza técnicas oriundas das DHTs para roteamento de nomes planos em redes físicas e diversos sistemas *pub/sub*, como os vistos na Seção 5.2.5, que encaminham notificações de eventos baseadas em interesses identificados através de AVPs, sem qualquer informação adicional sobre identificação ou localização de assinantes [Martins e Duarte 2010]. Dessa forma, como a viabilidade dos mecanismos de roteamento baseado em nomes já foi estudada, o desenvolvimento complementar desses mecanismos para ROCs deve priorizar o tratamento de questões relativas à eficiência dos protocolos em termos da quantidade de mensagens de controle trafegadas na rede, da escalabilidade devido ao tamanho das tabelas de roteamento e demais estruturas utilizadas no roteamento e ao "esticamento" de rotas em relação à rota de menor caminho entre fonte e destino. Rotas são consideradas esticadas sempre que possuem um caminho mais longo que o caminho físico ótimo, de acordo com alguma métrica de proximidade do roteamento, como número de saltos ou tempos de ida e volta (RTT). Rotas que apresentam esticamento implicam a perda na qualidade do serviço de entrega de conteúdos dinâmicos e sensíveis a atraso, como a distribuição de vídeo, devido principalmente à latência.

De forma geral, o uso de estruturas hierárquicas tem sido a técnica mais usada para prover escalabilidade aos protocolos de roteamento. As estruturas hierárquicas permitem o roteamento em agregados de níveis hierárquicos maiores até que seja atinjido o nível do destino, apresentando agregados de maior granularidade. Estruturas hierárquicas requerem, normalmente, endereços dependentes de localização, o que contraria os princípios das ROCs e podem resultar em consideráveis esticamentos de rotas dada a rigidez topológica [Singla et al. 2010]. Uma solução trivial para esse problema de roteamento é a utilização de difusão de informações para todos os nós. Tal difusão pode ser implementada através da inundação (*flooding*) que, em termos de roteamento, não requer o armazenamento de estados, adapta-se a modificações de topologia e aumenta a disponibilidade das informações [Martins e Duarte 2010]. A adoção de mecanismos de roteamento não-hierárquicos traz o problema de escalabilidade devido ao excesso de mensagens de controle de roteamento trafegadas na rede em virtude da inundação.

Uma forma de reduzir a quantidade dessas mensagens é utilizar algoritmos em que roteadores necessitem somente de informações locais [Zhang et al. 2010], ou seja, somente as próprias informações e as de vizinhos diretos. Uma técnica que pode ser empregada para esse fim é o uso de roteamento de requisições (*query routing*). Em S-BECON [Rosensweig e Kurose 2009], "migalhas de pão" (*breadcrumbs* - BC) são estruturas de dados voláteis que armazenam, em cada roteador, informações relativas ao recebimento e encaminhamento de conteúdos. O mecanismo utiliza o seguinte princípio probabilístico: dado que determinado conteúdo foi recebido e encaminhado recentemente pelo roteador, pode-se afirmar que o encaminhamento de uma nova requisição do conteúdo em direção à fonte (interface de recebimento, ou *upstream*) ou em direção aos usuários servidos (interface de encaminhamento, ou *downstream*) irá, com alguma probabilidade, encontrar uma cópia válida do conteúdo. O uso de políticas do tipo LRU e encamin-

hamento *downstream* podem proporcionar maiores ganhos, uma vez que armazena-se por mais tempo os conteúdos mais recentes e aumenta-se a probabilidade de atendimento à requisição. Resultados de simulações mostram que, quando o tamanho do *cache* é relativamente menor que a variedade de conteúdos, o uso de BCs torna possível obter conteúdos com grande eficácia, mostrando que restrições no tamanho do *cache* nos roteadores afetam muito mais os tempos necessários para obtenção de conteúdos do que sua disponibilidade absoluta. Outra abordagem semelhante é utilizada em SCAN [Lee et al. 2011], que é um mecanismo escalável de roteamento baseado em nomes planos. O SCAN utiliza, entre outras técnicas, a troca periódica de informações de roteamento entre nós vizinhos e permite que qualquer roteador, ao receber uma requisição de conteúdo, realize o direcionamento dessas requisições aos vizinhos, cujas informações de roteamento permitam atender à requisição. Esse processo de busca do conteúdo baseado nas informações de vizinhos é denominado roteamento de varredura (*scan routing*) e utiliza a mesma lógica de S-BECON. Tal mecanismo será resumidamente descrito mais adiante.

Outra questão fundamental quando se utiliza o roteamento baseado em nomes de conteúdos ao invés do roteamento baseado em endereços é o crescimento das tabelas de roteamento. Como a primitiva de rede deixa de ser baseada em nós, na ordem dos bilhões atualmente na Internet, passando a ser baseada em conteúdos, estimados na ordem de centenas de trilhões, devem ser adotadas técnicas que favoreçam a redução das tabelas de roteamento. Lida-se, usualmente, com esse desafio através da agregação de endereços, o que reduz a quantidade de memória para armazenar a tabela e o tempo de processamento da busca na tabela por um dado conteúdo. Porém, em muitas das soluções propostas para as ROCs, a agregação é impossível ou bastante difícil, sendo necessária a utilização de estruturas de dados compactas para a representação das tabelas de roteamento, capazes de reduzir os requisitos de armazenamento e utilização destas informações.

O SCAN, como visto anteriormente, prevê a troca de informações de roteamento de conteúdos entre nós vizinhos para a realização de varreduras. Tais informações são provenientes da tabela de roteamento de conteúdos (*content routing table* - CRT), que são estruturas compactas para o armazenamento de informações relativas ao armazenamento de conteúdos nos nós. Como a quantidade de entradas numa tabela desse tipo pode crescer consideravelmente, o SCAN comprime tais informações utilizando filtros de Bloom [Broder e Mitzenmacher 2002]. Filtros de Bloom são estruturas de dados probabilísticas que permitem verificar se determinado elemento faz parte de um conjunto. Um filtro de Bloom é, usualmente, formado por uma cadeia de m bits, todos inicialmente preenchidos com 0. Quando se deseja inserir um elemento no filtro, aplica-se a esse elemento k funções *hashes* independentes, que retornam inteiros entre 0 e $m-1$. Dessa forma, os bits correspondentes aos k resultados das funções *hash* são carregados com 1. Dessa forma, para realizar a verificação se determinado elemento pertence ao filtro, basta checar se os k elementos registram o valor 1 (ou, no caso de aceitar-se aproximações, se uma quantidade de bits menor que k significante suficiente registram 1) [Lee et al. 2011]. Cabe notar que filtros de Bloom são estruturas probabilísticas e, dessa forma, os mapeamentos possibilitados por tais estruturas estão sujeitos a falsos positivos. Em termos de sua aplicação em redes, essa característica se traduz em envio de dados a nós errados, caracterizando um aumento na sobrecarga de controle, que pode ser minimizada escolhendo-se adequadamente o tamanho em bits do filtro e a quantidade de funções *hash* utilizadas.

Carzaniga *et al.* propõem o B-DRP, um mecanismo de roteamento para sistemas *pub/sub* baseados em conteúdos que utiliza particionamento dinâmico de receptores (*dynamic receiver partitioning* - DRP) e filtros de Bloom na representação de predicados [Carzaniga et al. 2009]. O B-DRP requer que cada roteador conheça os predicados anunciados por todos os roteadores, o que caracteriza um severo obstáculo à escalabilidade. Por isso, o B-DRP também prevê a compactação dessas informações através da codificação desses predicados em estruturas baseadas em filtros de Bloom. Algumas propostas [Kumar et al. 2005,Lee et al. 2011] utilizam o conceito de filtro de Bloom com decaimento exponencial (*exponential decay Bloom filter* - EDBF), que representa a inversão de bits 1 para 0 com probabilidade aumentando exponencialmente com a distância do nó emissor do EDBF. Dessa forma, as informações de roteamento locais são compartilhadas, porém possuem muito mais relevância e validade num pequeno raio em torno do nó, diminuindo também o tráfego de controle na rede.

Por fim, o potencial de esticamento de rotas propiciado pelas soluções de roteamento baseadas em nomes deve-se, em grande parte, ao rompimento entre identificação e localização das ROCs. Nós em estruturas hierárquicas de nomes ou objetos com chaves de um espaço de nomes circular de uma DHT podem não possuir o estado necessário para estabelecimento de rotas com o caminho físico mais curto ainda que possuam vizinhança no plano de nomes ou da aplicação. O ROFL, por exemplo, possui, na prática, um esticamento alto, tendendo ao ilimitado em topologias genéricas [Singla et al. 2010]. A forma mais comum de lidar com esses esticamentos de rotas é introduzir nos mecanismos de roteamento aspectos ligados à proximidade física dos nós, como o roteamento por proximidade (*proximity routing*) e a seleção de vizinhos por proximidade (*proximity neighbour selection*) [Ratnasamy et al. 2002]. Essas duas soluções amplamente adotadas em redes P2P. O roteamento por proximidade consiste em levar em consideração a proximidade dos nós de próximo salto existentes na tabela de roteamento. Dessa forma, o roteamento mantém o equilíbrio entre o avanço em direção ao identificador de destino e a seleção de nós mais próximos da origem. São exemplos de aplicação de roteamento por proximidade as propostas de redes P2P CAN [Ratnasamy et al. 2001] e Chord [Stoica et al. 2003], que utilizam medições de RTT de seus vizinhos (CAN) e listas de nós mais próximos que possibilitam saltos para regiões específicas do plano de nomes (Chord), e utilizando o nó com menor RTT (ou o mais próximo) dentre os que possibilitam avanço em direção ao destino. Tal abordagem confere um potencial de redução do esticamento ao mecanismo de roteamento, porém como existem muito mais nós na rede sobreposta próximos à origem do que vizinhos, a redução dos caminhos obtidos pode ser bastante limitada. O mecanismo de seleção de vizinhos por proximidade consiste em construir tabelas de roteamento levando-se a topologia em consideração. Toda entrada na tabela de roteamento de um nó é referente a um nó próximo, de acordo com alguma métrica de proximidade, como contiguidade no espaço de nomes e RTT. Isso permite uma redução na distância percorrida pelas mensagens, sem aumento significativo na quantidade de saltos [Castro et al. 2002]. MultiCache [Katsaros et al. 2011] é uma proposta de arquitetura sobreposta para ROCs, que implementa o Pastry [Rowstron e Druschel 2001] como substrato de roteamento. O Pastry utiliza a seleção de vizinhos por proximidade, gerando rotas mais curtas e convergentes para fluxos com origens próximas e mesmos destinos. Dessa forma, a consideração de proximidade no estabelecimento de enlaces com vizinhos confere a estes sistemas um

melhor desempenho, já que os caminhos utilizados na distribuição de conteúdos tendem a apresentar atrasos menores.

5.4.3. Armazenamento na Rede (*caching*)

Um dos principais fundamentos de uma ROC é o armazenamento de conteúdo na rede através do uso de *cache* em todos os roteadores da rede [Ghodsi et al. 2011a]. O principal objetivo consiste no aumento do desempenho da rede na distribuição dos conteúdos. Esse aumento de desempenho é proporcionado pela diminuição do atraso percebido pelo usuário, pelo aproveitamento mais eficiente da banda passante no núcleo da rede, pela disponibilidade de múltiplas cópias evitando assim o ponto único de falha e pela diminuição do tráfego próximo à fonte de conteúdo, que reduz a carga de processamento. O armazenamento de conteúdo e o uso de *cache* são temas bastante estudados, sobretudo a partir do surgimento da *Web*. Recentemente, o emprego de CDNs também incitou um grande número de trabalhos, principalmente sobre o problema de distribuição dos *caches* na rede bem como a distribuição dos conteúdos nos *caches* de maneira a minimizar o atraso e otimizar o uso da banda passante na rede. No entanto, o uso do *cache* em redes orientadas a conteúdo possui características bem distintas das mencionadas anteriormente. A primeira característica importante é o emprego do *cache* em todos os componentes da rede, formando uma rede de *caches*[9], ao contrário do *web-caching*, o qual apresenta uma topologia hierárquica em árvore, e das CDNs, nas quais os *caches* são posicionados em pontos específicos da rede. Essa característica proporciona uma grande flexibilidade para alocação de conteúdo, expandindo o acesso a tal funcionalidade a todos os nós da rede e não somente aos servidores de conteúdo, como acontece nas CDNs. A segunda característica relevante é que, dependendo da arquitetura de ROC, os *caches* armazenam pedaços de mesmo tamanho (*chunks*) dos conteúdos ao invés de armazenar o conteúdo inteiro. O tamanho dos conteúdos pode ter um grande impacto no desempenho do sistema de armazenamento de conteúdo [Podlipnig e Böszörmenyi 2003]. Em arquiteturas, como a CCN, o fato de haver um tamanho único para os pedaços armazenados simplifica o problema.

Os principais temas de pesquisa nesta área podem ser resumidos por: (i) modelos analítico para redes de *cache*, (ii) estratégias de descarte de conteúdo e (iii) políticas de armazenamento. Todos os trabalhos presentes na literatura realizam algum tipo de avaliação do desempenho da rede de *cache* a fim de investigar o modelo proposto, a política de *cache*, a política de armazenamento, o modelo de segurança, entre outros aspectos funcionais. As métricas mais utilizadas para medir o desempenho das redes de *caches* são (i) a taxa de acerto (*hit ratio*), que define a fração de pedidos de conteúdo que foram encontrados no *cache*, (ii) o atraso percebido pelo usuário, que mede o tempo de espera para receber um dado conteúdo após o envio da respectiva requisição e (iii) número médio de saltos que um pedido atravessa antes de ser atendido. Todos os trabalhos utilizam a distribuição Zipf-like ($1/i^{\alpha}$) para modelar a popularidade dos conteúdos [Breslau et al. 1999]. Porém, os trabalhos utilizam diferentes valores para o parâmetro α.

Em seguida são apresentados os principais desafios e trabalhos propostos em cada um dos temas de pesquisa mencionados anteriormente.

[9]O termo usado em inglês é *in-network caching*.

5.4.3.1. Modelos Analíticos para Redes de *Cache*

O maior desafio nesta área de pesquisa é a grande complexidade que representa uma rede de *caches* do tamanho da Internet. Existem muitos trabalhos que propõem novos modelos na tentativa de compreender a dinâmica e analisar o desempenho das redes de *caches* [Rosensweig et al. 2010, Psaras et al. 2011, Carofiglio et al. 2011a, Carofiglio et al. 2011b, Fricker et al. 2012]. Existem alguns trabalhos que modelam sistemas de *web-caching* hierárquicos, com topologias específicas nas quais o servidor de origem do conteúdo está conectado ao topo da hierarquia, como nas topologias em árvore [Che et al. 2001, Che et al. 2002]. Além disso, devido a complexidade desses modelos, esses são usualmente empregados em topologias pequenas, como em árvores de dois níveis [Rosensweig et al. 2010].

Em um dos primeiros esforços para modelar as redes de *caches*, Rosensweig *et al.* propõem um algoritmo aproximativo para caracterizar o comportamento de redes de *caches* com topologias genéricas [Rosensweig et al. 2010]. A abordagem utilizada consiste em definir um modelo para apenas um roteador com um único *cache*. Assim, os pedidos que chegam a esse roteador são todos aqueles vindos diretamente para o roteador somados aos pedidos não encontrados nos *caches* dos vizinhos. Considera-se, nesse caso, que após um pedido não ser encontrado, ele é reencaminhado pelo menor caminho até a origem dos conteúdos. Em um processo interativo, o algoritmo atualiza os pedidos que chegam em cada roteador até que toda a rede convirja para um estado estacionário. Os autores afirmam que o modelo proposto serve para qualquer topologia independentemente da escala. Porém, a complexidade do algoritmo é dada por $O(KB)$, o que prejudica o uso desse modelo para redes com muitos conteúdos (K) e com tamanhos grandes de *cache* (B) [Psaras et al. 2011]. A fim de resolver este problema, Psaras *et al.* propõem um modelo mais simples para o *cache* de apenas um roteador [Psaras et al. 2011]. Esse modelo utiliza uma cadeia de Markov contínua e homogênea, na qual cada estado da cadeia representa a posição atual do conteúdo no *cache*. Os autores consideram que sempre que um conteúdo é requisitado, ele é armazenado na primeira posição do *cache* (topo). No caso de ser um conteúdo que não esteja no *cache*, todos os outros conteúdos são movidos uma posição para baixo. Caso contrário, o conteúdo requisitado é movido para o topo, e apenas os conteúdos armazenados acima do conteúdo encontrado no *cache* mudam de posição. Assim, considerando N o tamanho do *cache*, um elemento no estado $N+1$ da cadeia está fora do *cache*. O modelo é, também, estendido para uma rede de *caches*. O maior objetivo é calcular o tempo em que um conteúdo permanece no *cache*. Por fim, os autores estendem o modelo para dois roteadores em cascata e analisam o desempenho de uma rede de *caches* em árvore.

Utilizando uma abordagem diferente, Carofiglio *et al.* propõem um modelo para transferências de pedaços de conteúdo (*chunks*) em redes de *caches* [Carofiglio et al. 2011b]. Primeiramente, é proposto um modelo probabilístico de falha na busca por um determinado conteúdo (probabilidade de erro) em apenas um roteador. O processo de requisições é modelado em dois níveis: conteúdo e pedaço de conteúdo. O processo de chegada de requisições é modelado a partir de um processo [1]MMRP (*Markov Modulated Rate Process*). Em seguida, os autores estendem o modelo para uma rede de *caches* com e sem agregação de pedidos, baseado em uma topologia em árvore ou em cascata. Por

fim, é apresentada uma fórmula fechada para a média da vazão estacionária em relação a diversos parâmetros, tais como, probabilidade de acerto, popularidade do conteúdo, tamanho do conteúdo e do *cache*. Em um segundo trabalho, Carofiglio *et al.* estendem o modelo anteriormente proposto para considerar limitações de banda passante e de capacidade de armazenamento [Carofiglio et al. 2011a]. Dessa maneira, os autores apresentam uma fórmula fechada para o atraso médio de entrega de um conteúdo considerando estas limitações acrescentadas ao modelo.

Com o intuito de avaliar o impacto de diferentes tipos de tráfego no desempenho de redes de *caches*, Fricker *et al.* modelam uma rede de *caches* estruturada em árvore, considerando quatro tipos de conteúdos diferentes: objetos *web*, compartilhamento de arquivos, conteúdos gerados por usuários e vídeo sob demanda [Fricker et al. 2012]. Essas categorias apresentam características bem distintas em termos de popularidade, população e tamanho de conteúdo. O modelo é baseado em uma aproximação do desempenho das políticas LRU (*Least Recently Used*) e LFU (*Least Frequently Used*) [Che et al. 2002]. Os autores mostram que o compromisso entre o tamanho do *cache* e a banda passante utilizada está fortemente relacionado com as características do conteúdo armazenado.

5.4.3.2. Estratégias de Descarte de Conteúdo

As políticas de descarte de pacotes estão associadas à substituição de conteúdos quando o *cache* atinge sua capacidade máxima de armazenamento e um novo conteúdo, ainda não armazenado no *cache*, é requisitado. Por isso, é necessária uma estratégia para escolher qual conteúdo será descartado para dar lugar ao último requisitado. As duas políticas mais utilizadas são a LRU e a LFU. A primeira, a mais simples, descarta o conteúdo acessado menos recentemente enquanto a segunda descarta o conteúdo usado menos frequentemente. Segundo Podlipnig e Böszörmenyi, apesar de haver uma quantidade significativa de diferentes propostas de políticas de descarte de conteúdo, existe um forte argumento alegando que o estudo de novas políticas mais eficientes é menos importante já que a capacidade dos dispositivos de armazenamento apresentava uma forte tendência de crescimento, aliada à diminuição constante dos preços [Podlipnig e Böszörmenyi 2003]. Portanto, as políticas de descarte mais simples apresentam desempenho semelhante às políticas mais complexas. No entanto, nas redes de *cache* a realidade é outra. O tamanho dos *caches* nos roteadores são limitados a quantidades relativamente pequenas de armazenamento se comparado ao universo de conteúdos a ser armazenado, em razão de custo e desempenho [Perino e Varvello 2011]. Contudo, mesmo com a limitação de capacidade de armazenamento dos roteadores, existe ainda um compromisso entre a complexidade da política de descarte e a capacidade de processamento do roteador. Por isso, a maioria dos trabalhos analisam apenas o desempenho das políticas LFU e LRU, sendo esda última considerada um limite superior de velocidade para as outras políticas [Rossi e Rossini 2011a]. Ainda assim, muitos trabalhos avaliam o desempenho das redes de *caches* considerando diferentes políticas de descarte de conteúdo.

Em uma das primeiras tentativas de investigar o desempenho das redes de *caches*, Choi *et al.* avaliam por meio de simulações dois tipos específicos de topologia para redes orientadas a conteúdo [Choi et al. 2009]. Nesse trabalho, são consideradas a topologia em

árvore e a topologia baseada em DHTs. Os autores analisam o tamanho do *cache*, o atraso de transferência e a robustez a falhas aleatórias de roteadores em ambas as topologias.

Rossi e Rossini analisam o desempenho das redes de *caches* baseado em características da infraestrutura da rede [Rossi e Rossini 2012]. Os autores consideram diversas métricas de centralidade, tais como grau, proximidade (*closeness*) e intermediação (*betweenness*), entre outras. Essas métricas refletem, de maneiras diferentes, a importância de um nó na rede. Assim, eles utilizam algumas topologias reais, com diferentes características estruturais. Por fim, são propostos dois critérios para atribuição de tamanhos de *cache* para cada nó da rede, proporcional às métricas de centralidade. O resultado mais importante desse trabalho mostra que o impacto da topologia na definição de *caches* de tamanhos diferentes é insignificante, sobretudo em relação a complexidade acrescentada pelo mecanismo proposto. Em outro trabalho, Rossi e Rossini analisam o desempenho de redes de *cache* com diversas políticas de descarte de conteúdo [Rossi e Rossini 2011a]. Entre as políticas analisadas estão a LRU, FIFO, na qual o conteúdo armazenado mais antigo é descartado, UNIF, onde um conteúdo é escolhido aleatoriamente com distribuição uniforme, e BIAS, na qual dois conteúdos são sorteados e o mais popular é descartado. Em um terceiro trabalho, Rossi e Rossini além de diferentes políticas de descarte, consideram ainda modelos de localidade e estratégias de encaminhamento da arquitetura CCN [Rossi e Rossini 2011b].

Carofiglio *et al.* apresentam duas novas estratégias para o descarte de conteúdos a fim de prover qualidade de serviço [Carofiglio et al. 2011c]. Na primeira estratégia, os autores sugerem a utilização da técnica de particionamento de *cache* [Lu et al. 2004], na qual uma fração da memória é alocada para cada aplicação. Pelo fato da alocação proposta ser estática, surge o problema de subutilização do *cache* quando uma determinada aplicação preenche toda a sua fração da memória e começa a usar a política de descarte, enquanto ainda há espaço livre na memória reservado a outras aplicações. A segunda estratégia consiste em definir categorias de prioridades para as aplicações e realizar o gerenciamento do descarte de acordo com essas prioridades, escolhendo assim qual categoria terá um pedaço descartado. Em um dos algoritmos propostos, um determinado pedaço só poderá ser descartado para dar lugar a um outro pedaço de prioridade igual ou superior a sua.

5.4.3.3. Políticas de Armazenamento

As políticas de armazenamento definem que conteúdos devem ser armazenados no *cache*. A política mais utilizada na literatura consiste em armazenar todos os conteúdos que não foram encontrados no *cache*, indiscriminadamente. Rossi e Rossini avaliam o desempenho, através de simulações, de diferentes políticas de armazenamento [Rossi e Rossini 2011a]. A primeira é a política de armazenar tudo que chega nos roteadores sem discriminação. A segunda política analisada [Arianfar et al. 2010b] armazena o conteúdo aleatoriamente com uma probabilidade fixa P. Por último, é analisada a política LCD (*Leave a Copy Down*), na qual os conteúdos que não são encontrados no *cache* de um determinado roteador são armazenados apenas no roteador seguinte (próximo salto) no caminho para o usuário que o requisitou.

Cho *et al.* propõem um esquema de armazenamento no qual o número de pedaços a ser armazenado depende da popularidade do conteúdo [Cho et al. 2012]. Além disso, os roteadores indicam aos seus vizinhos se um dado pedaço deve ser armazenado ou não, marcando um bit de armazenamento. Dessa maneira, evita-se que muitos nós do caminho armazenem os mesmos conteúdos. Nesse esquema, os roteadores tendem a "empurrar" os conteúdos mais populares na direção dos roteadores mais próximos dos usuários que fizeram o pedido.

Existem também pesquisas de novas políticas de armazenamento e posicionamento de *caches* em redes baseadas no paradigma *pub/sub*. Diallo *et al.* introduzem novas políticas de armazenamento [Diallo et al. 2011]. As diferentes políticas priorizam diferentes tipos de conteúdo tais como conteúdos que têm um pedido pendente ou que já foram expedidos. Os autores também analisam o desempenho das políticas propostas usando diferentes políticas de descarte. Sourlas *et al.* propõem um algoritmo para definir o posicionamento de *caches* e a atribuição de réplicas de conteúdo a fim de possibilitar o acesso a conteúdos cujo "publicador" já se desconectou da rede [Sourlas et al. 2011].

5.4.4. Segurança

Um dos principais problemas de segurança nas ROCs é a confiança no conteúdo, isto é, como saber que um determinado conteúdo é realmente aquele que está sendo requisitado. Na arquitetura atual, esse problema é resolvido a partir do conhecimento da origem do conteúdo e das características da comunicação. O conhecimento da origem pressupõem confiança no *site* de busca, bem como no sistema DNS [Smetters e Jacobson 2009]. Portanto, existem três problemas básicos a serem resolvidos para se prover confiança no conteúdo: (i) verificar a integridade do documento, (ii) verificar a proveniência do conteúdo e (iii) verificar a relevância do conteúdo obtido em relação ao requisitado. Muitos trabalhos [Koponen et al. 2007, Walfish et al. 2004, Popescu et al. 2005] focam em resolver [2]esses problemas com nomes autocertificadores (*self-certifying names*), no qual cada nome é gerado através de uma operação criptográfica com o próprio conteúdo. No entanto, essa abordagem dá origem a um espaço de nomes plano, no qual a busca por conteúdos é significativamente mais complexa que usar a nomeação baseada em DHTs. Além disso, essa técnica de nomes autocertificadores resolve apenas o problema de integridade. Uma outra abordagem consiste em realizar uma operação criptográfica com a chave que foi usada para assinar o conteúdo. Essa abordagem possibilita que a proveniência seja verificada [Popescu et al. 2005]. No entanto, o maior problema de ambas as técnicas é a necessidade de haver um mapeamento entre os nomes gerados a partir das funções *hash* e nomes amigáveis para os usuários. Esse problema exige a existência de um mecanismo que possa prover confiança no mapeamento. Para evitar esse mapeamento, Koponen *et al.* propõem a concatenação de um rótulo escolhido pelo publicador do conteúdo, que represente um nome amigável, com o resultado do *hash* da chave pública do publicador [Koponen et al. 2007]. Entretanto, o rótulo não é assinado e, por isso, permite que um atacante associe um conteúdo válido e assinado a um rótulo qualquer de sua escolha. Dannewitz *et al.* possuem uma abordagem similar à anterior para prover nomes com autocertificação [Dannewitz et al. 2010]. Smetters e Jacobson propõem autenticar a relação entre nomes e conteúdos, ao invés de autenticar cada um deles [Smetters e Jacobson 2009]. Assim, os nomes teriam a função apenas de identificador dos conteúdos de

forma amigável e de localizador dos mesmos, enquanto a autenticação seria atingida por meio da validação do mapeamento de um conteúdo e seu respectivo nome.

Ainda com relação à nomeação, outro problema identificado é a falta de privacidade. O uso de nomes dos conteúdos para realizar as operações básicas da rede introduz uma falha na privacidade dos usuários. Nas ROCs, os roteadores têm acesso direto à requisição de conteúdo dos usuários bastando, portanto, que um dos roteadores da rede seja comprometido para permitir que um atacante seja capaz de monitorar os pedidos enviados pelos usuários. Além disso, a censura de conteúdos, prática adotada por alguns países, é bastante facilitada, pelo acesso direto a informações de conteúdo requisitado. Arianfar *et al.* propõem um mecanismo para aumentar a privacidade dos usuários, dificultando o trabalho dos atacantes de monitorar os pedidos realizados [Arianfar et al. 2011]. O esquema proposto visa dificultar a identificação de uma requisição a um conteúdo proibido, bem como dificultar que um conteúdo recuperado seja identificado como um conteúdo proibido. A técnica utilizada consiste em misturar blocos de conteúdos proibidos com conteúdos normais de maneira que um pedaço de conteúdo seja composto por mais de um bloco. Assim, o usuário deve requisitar um número de pedaços que o permita recuperar mais de um conteúdo, e não somente o proibido. Dessa maneira, o trabalho do atacante para descobrir o que está sendo requisitado baseado apenas nos pedidos de pedaços é bastante dificultado.

Outro problema importante de segurança presente nas ROCs é a vulnerabilidade à poluição de *cache* [Gao et al. 2006]. Esse ataque consiste em enviar pedidos de conteúdo aleatórios com o intuito de alterar a popularidade dos conteúdos, provocando o armazenamento de conteúdos de pouca popularidade nos *caches* dos roteadores. Uma variação desse ataque é fazer apenas requisições de conteúdos pouco populares. No entanto, é necessário o conhecimento prévio da popularidade dos conteúdos. Um caso mais grave desse ataque consiste em fazer pedidos de conteúdos falsos, criados apenas para poluir os *caches* e que estão fora do universo de conteúdos válidos. Xie *et al.* propõem um mecanismo de armazenamento que reduz o efeito de um ataque de poluição [Xie et al. 2012]. O mecanismo proposto armazena apenas os conteúdos pedidos com maior frequência. Os autores mostram através de simulações a eficiência do mecanismo proposto para ataques de envio de requisições à conteúdos legítimos mas de forma aleatória, uniformemente distribuída.

Apesar de a segurança ser uma área importante com muitas questões em aberto, os esforços empreendidos no sentido de resolver estes problemas são ainda bastante restritos e insipientes.

5.4.5. Aspectos Práticos

Diferentes arquiteturas, como as descritas na Seção 5.3.4, são avaliadas em plataformas de teste experimentais e possuem protótipos das suas pilhas de protocolos, como é o caso do CCNx[10]. Para adoção dessas propostas em larga escala, no entanto, é necessário definir questões práticas de implementação das ROCs. Entre essas questões estão a definição de um modelo econômico para uma nova Internet baseada em ROCs, a implementação de roteadores de conteúdo e o desenvolvimento e a avaliação de aplicações

[10]http://www.ccnx.org.

tipicamente conversacionais. A seguir, essas questões práticas são discutidas.

5.4.5.1. Modelo econômico

Para a adoção das ROCs em larga escala, é fundamental definir um modelo que estimule os atuais operadores de rede e de serviços a migrarem para uma arquitetura de ROCs e que garanta a remuneração de suas atividades, mesmo com a adoção de novas tecnologias [Feamster et al. 2007].

O modelo de negócios adotado atualmente na Internet é baseado na conectividade. Em geral, os usuários pagam aos seus ISPs locais para terem acesso à rede [Trossen e Biczók 2010]. A função de um ISP, nesse caso, é simplesmente encaminhar pacotes. Como os serviços de rede são prestados fim-a-fim e a Internet é organizada em sistemas autônomos, os ISPs também pagam para enviar seu tráfego para outros ISPs.

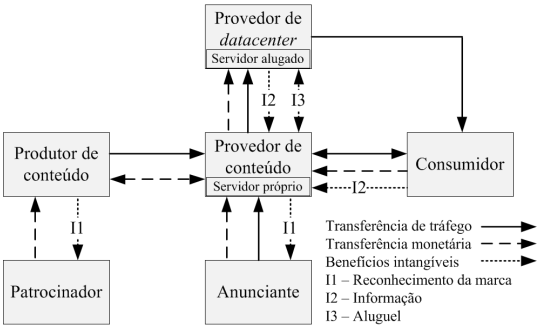
Os serviços básicos de entrega estão se tornando *commodities* e, por isso, os ISPs procuram por novos serviços que aumentem suas receitas [Feamster et al. 2007]. Atualmente, já existem serviços diferenciados que oferecem distribuição de voz e vídeo e usuários que pagam mais por tal diferenciação [Trossen e Biczók 2010]. Outro exemplo nesse sentido são as CDNs que adotam um modelo de negócios particular. Um provedor de CDN oferece seus serviços para produtores que pagam para distribuir seus conteúdos aos consumidores de forma mais eficiente. O provedor de CDN possui uma plataforma central que coordena o provisionamento de serviços, oferece garantias de nível de serviço e é responsável pela tarifação dos usuários. Entretanto, esse modelo não é eficiente para as ROCs, uma vez que é fortemente baseado em um elemento central e, por isso, compromete a escalabilidade de tais redes.

Ainda não há um modelo econômico definido e bem aceito para as ROCs, mas já existem algumas propostas [Trossen et al. 2010, Zhang et al. 2011, Biraghi et al. 2011]. Zhang *et al.* definem os atores de um modelo de negócios proposto para ROCs [Zhang et al. 2011]. São eles os *produtores de conteúdo*, que criam conteúdos, os *provedores de conteúdo*, que são portais que agregam conteúdos de diferentes produtores e os *consumidores*, que solicitam o conteúdo. Os autores também consideram a existência de *provedores de datacenter*, que oferecem recursos de processamento e armazenamento e de ISPs, que são divididos em provedores de redes de acesso (*Internet Access network Providers* - IAPs) e provedores de *backbone* (*Internet Backbone Providers* - IBPs). Por fim, existem *anunciantes*, que correlacionam sua marca com os conteúdos durante o processo de distribuição, e *patrocinadores*, que adicionam suas marcas aos conteúdos durante a fase de produção. Esses atores atuam nas duas camadas propostas para a distribuição de conteúdos: a camada de produção de conteúdo e a camada de interconexão.

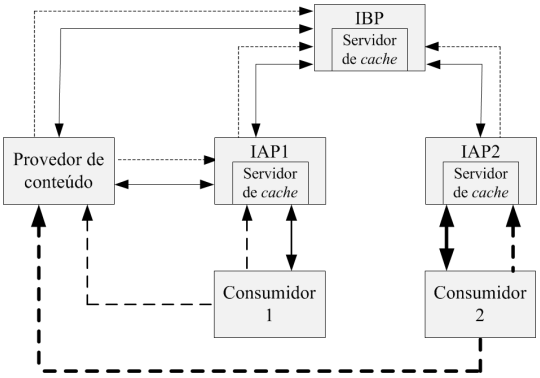
A camada de produção de conteúdo é a de mais alto nível, na qual a unidade de dados são objetos digitais como vídeos, músicas, arquivos, etc. A Figura 5.9(a) mostra uma rede de valor[11] simplificada para a camada de produção, na qual estão ilustrados os processos de (i) criação e publicação de conteúdo, (ii) as diferentes possibilidades

[11]Uma rede de valor é um método usado para analisar modelos de negócio que descrevem recursos técnicos e sociais e ilustra as interligações entre os diferentes atores do modelo.

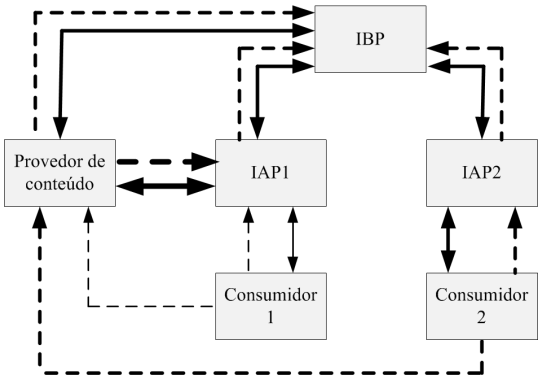
de armazenamento desse conteúdo e (iii) as diferentes fontes de receita do provedor de conteúdo. Os nós representam os atores e as setas que interconectam os nós representam as transferências de tráfego, monetárias e os benefícios intangíveis entre os atores.



(a) ROC - camada de produção de conteúdo.



(b) ROC - camada de interconexão.



(c) Cliente-servidor - camada de interconexão.

Figure 5.9. As redes de valor para uma ROC e para o modelo cliente-servidor.

A camada de interconexão provê a infraestrutura de rede necessária para o encaminhamento e o armazenamento do conteúdo. A unidade de dados nessa camada são os bits. A Figura 5.9(b) ilustra uma rede de valor simplificada para a camada de interconexão, na qual a espessura das setas indica o volume relativo de tráfego entre os atores. É importante notar que a distribuição se dá entre os consumidores e provedores de conteúdo e não envolve os produtores. A diferença básica dessa rede de valor para a do modelo cliente-servidor, ilustrada na Figura 5.9(c), é a presença do servidor de *cache* nos IAPs e IBPs [Zhang et al. 2011]. Um dos benefícios em virtude da introdução do *cache* nesses atores é a redução do tráfego entre os próprios IAPs e IBPs e entre tais provedores e os provedores de conteúdo, como indicam a setas de menor espessura na Figura 5.9(b). Outro detalhe é que o custo do Consmuidor 2 é maior do que o do Consumidor 1, pois ele se encontra em um IAP diferente do provedor de conteúdo.

A possibilidade de armazenamento de conteúdos nos elementos de rede proporciona redução de tráfego e, consequentemente, do custo de encaminhamento. Entretanto, introduz novos desafios relativos à tarifação e ao gerenciamento das ROCs. A redução do custo de encaminhamento, por exemplo, é obtida em troca do armazenamento na rede. Atualmente essa troca é interessante visto que o custo do armazenamento decai mais rap-

idamente do que o de transmissão. Porém, a comparação entre os custos de ambos é difícil, pois o armazenamento e transmissão são tarifados de maneira diferente [Biraghi et al. 2011]. Em geral, paga-se por uma determinada capacidade de armazenamento que é reservada e pode ser reutilizada múltiplas vezes. Um mesmo conteúdo, por exemplo, pode ser acessado múltiplas vezes no *cache* e a cobrança não se dá por esse número de acessos, mas sim pelo espaço ocupado pelo conteúdo. Por outro lado, o encaminhamento é um serviço do tipo pague-pelo-uso, ou seja, a cada reenvio se fatura o mesmo pacote.

Outra questão primordial é se o *cache* deve ser transparente ou baseado em acordos comerciais. Para a maioria das arquiteturas de ROCs, o *cache* é transparente, ou seja, todos os roteadores armazenam conteúdos sem distinção de quem os publica ou solicita. Entretanto, atualmente nas CDNs ele é feito baseado em acordos comerciais com grande sucesso [Biraghi et al. 2011]. Além disso, ainda não há uma definição sobre quem deve decidir sobre o armazenamento de um dado conteúdo em um elemento da rede: o ISP e/ou o provedor de conteúdo. Em caso de ambos terem controle, podem existir conflitos e, por isso, é necessário desenvolver mecanismos de controle multipartite. Por exemplo, IAPs podem atualizar seu *cache* com uma frequência mais baixa do que a desejada pelo provedor de conteúdo. Medidas contratuais e incentivos monetários podem resolver esse conflito. Nesse cenário também pode existir o regulador de conteúdo, que é o elemento responsável pela resolução de conflitos de interesse entre os atores da rede, sejam eles consumidores, ISPs ou provedores de conteúdo. O regulador pode determinar, por exemplo, se um conteúdo é sensível e, por isso, não deve ser armazenado pelos elementos da rede ou ainda se os elementos que armazenam um dado conteúdo são qualificados para tanto. Outro ponto ainda sem definição é se o *cache* é feito apenas pelos elementos da rede, como roteadores e servidores de *cache*, ou também pelos sistemas finais. As duas soluções são implementadas atualmente pelas CDNs e pelas redes P2P, respectivamente. Da mesma forma, não há definição se os elementos de rede estão sob a mesma administração ou se são "abertos". A propriedade do local de armazenamento define quem é o responsável por pagar pelo *cache* e quem se beneficia dele.

Uma questão fundamental para adoção das ROCs é estimular os ISPs a mudarem o atual modelo de negócios. Hoje, como indica o modelo adotado pelas CDNs, os provedores de conteúdo estão mais dispostos a pagar por serviços diferenciados de entrega de conteúdo do que os ISPs [Trossen e Biczók 2010]. Entretanto, os ISPs têm papel fundamental para as ROCs, visto que eles são os locais onde os *caches* devem ser implementados. Os IAPs podem ser estimulados pela redução de tráfego interdomínio e de custos com a adoção de *caches*. Atualmente, os IBPs sofrem com a queda de preço das trocas de tráfego e, por isso, buscam novas fontes de receita [Zhang et al. 2011]. Alguns se tornaram também provedores de CDN, o que pode indicar que os IBPs estão abertos às mudanças introduzidas pelas ROCs.

Roteadores de Conteúdo

A implementação dos roteadores de conteúdo é outro grande desafio para o desenvolvimento das ROCs [Ahlgren et al. 2008, Arianfar et al. 2010a, Perino e Varvello 2011]. Esses roteadores, diferentemente dos roteadores tradicionais, não são responsáveis apenas pelo encaminhamento dos pacotes e cálculo das tabelas de rotas. A operação chave desses elementos é o armazenamento dos conteúdos, que passa a ser operação interna dos

roteadores e não mais um procedimento coordenado por outros elementos da rede [Arianfar et al. 2010a]. Assim, cada roteador de conteúdo deve ser capaz de encaminhar, armazenar e procurar conteúdos em seu *cache* com base nos identificadores dos conteúdos e não mais nos endereços de destino, bem como encaminhar e processar pacotes de controle. Além disso, também deve manter estados para encaminhar de volta um dado conteúdo para um usuário que demonstrou interesse nesse conteúdo.

A introdução das operações associadas ao armazenamento de conteúdos requer modificações de *hardware* e *software* nos roteadores de pacotes atuais. O encaminhamento de conteúdos baseados em nomes e o *cache* na granularidade de pacotes exigem, por exemplo, velocidades de processamento maiores do que as dos roteadores atuais. Outra modificação é o aumento do número de estados a serem armazenados pelos roteadores em virtude da mudança de espaço de endereçamento. Atualmente, esse espaço consiste de um bilhão de endereços IP, entretanto, passará a ser de pelo menos um trilhão de nomes de conteúdos [Perino e Varvello 2011]. Por outro lado, a adoção de um *cache* local pode reduzir o número de operações de encaminhamento realizadas por um roteador.

Arianfar *et al.* avaliam a viabilidade da implementação de roteadores de conteúdos com base nas configurações de *hardware* e *software* de roteadores atuais [Arianfar et al. 2010a]. Para tanto, os autores definem um modelo de referência para um roteador de conteúdo e propõem mecanismos para implementar suas funcionalidades. O modelo definido considera a implementação do armazém de conteúdos (*Content Store* - CS) definido para os nós da arquitetura CCN, ilustrado na Figura 5.6, e a hierarquia de memória em três níveis adotada por roteadores-padrão atuais que usam CAM (*Content-Addressable Memory*), SRAM (*Static Random-Access Memory*) e DRAM (*Dynamic Random-Access Memory*). O CS é composto por dois componentes principais: o armazém de pacotes e a tabela de indexação. Cada entrada da tabela contém o identificador do pacote, seu endereço de memória no armazém de pacotes e informações de estado. As entradas da tabela de indexação são divididas entre a DRAM e a SRAM para reduzir a quantidade de memória SRAM usada e, consequentemente, o custo do equipamento, como pode ser verificado pela Tabela 5.2. O armazém usa apenas memória DRAM.

Table 5.2. Parâmetros característicos dos tipos de memória [Perino e Varvello 2011].

Tipo	Tempo de acesso (ms)	Tamanho máximo	Custo (USD/MB)	Consumo energético (W/MB)
TCAM	4	$\approx$ 20 Mb	200	15
SRAM	0,45	$\approx$ 210 Mb	27	0,12
RLDRAM	15	$\approx$ 2 Gb	0,27	0,027
DRAM	55	$\approx$ 10 GB	0,016	0,023
High-speed SSD	1.000	$\approx$ 10 TB	0,03	0,00005
SSD	10.000	$\approx$ 1 TB	0,003	0,00001

Entre as funcionalidades de um roteador de conteúdo definidas por Arianfar *et al.* estão a inserção, a remoção e a busca de pacotes de conteúdo no CS e a identificação, o enfileiramento e o encaminhamento de pacotes. A partir dessas definições, estimam-se os recursos computacionais exigidos por cada funcionalidade individualmente. Os recursos avaliados são poder de processamento, tempo de acesso à memória, quantidade

de memória exigida para armazenamento de conteúdos e consumo de energia.

O processamento não é um fator crítico. As operações relacionadas ao CS envolvem a busca, a verificação e a atualização da tabela de indexação, o que requer poucos ciclos de relógio considerando o uso de ASICs e FPGAs. Por outro lado, o tempo de acesso à memória pode ser um gargalo. O tempo de acesso típico de uma DRAM é da ordem de 50 ns e os intervalos entre a chegada de pacotes de 40 bytes em enlaces de 10 Gb/s (OC-192) e 40 Gb/s (OC-768) são, respectivamente, 32 ns e 8 ns. Nesse caso, a solução mais simples para evitar o gargalo é usar bancos de memória em paralelo ou substituir as atuais DRAM por memórias mais rápidas como as RLDRAMs (*Reduced Latency* DRAMs), que possuem tempo de acesso da ordem de 15 ns, como indica a Tabela 5.2. Para avaliar a quantidade de memória necessária para o armazém de conteúdos, os autores assumem que todo o tráfego que passa por uma interface de rede de 10 Gb/s em um intervalo de 10 s é armazenado. Isso exige uma quantidade de memória DRAM de aproximadamente 100 Gb com custo de até USD 300, atualmente. Além disso, calcula-se que devem ser mantidas 9 milhões de entradas de 36 bits na porção da tabela de indexação armazenada na SRAM. Logo, são necessários 324 Mb com custo atual aproximado de USD 500. No total, o custo com memória por porta do roteador é de USD 800,00. Atualmente, um roteador comercial de 10 Gb/s possui custo por porta entre USD 1.500 e 2.500. Portanto, a adaptação de um roteador convencional para um roteador de conteúdo aumentaria em no máximo 30% o seu custo por porta. O aumento do consumo de energia por cada interface em virtude do aumento de memória seria da ordem de 100 kWh/ano, o que representa um pequena parcela do consumo total do equipamento e um aumento de custo de USD 20. Portanto, os autores concluem que a transformação dos roteadores convencionais em roteadores de conteúdo é viável em termos da quantidade de recursos necessária e dos custos desses recursos adicionais.

Perino e Varvello também analisam a viabilidade da implementação de roteadores de conteúdo usando as configurações de *hardware* e *software* de roteadores atuais [Perino e Varvello 2011]. No entanto, utilizam um modelo de referência mais complexo para os roteadores de conteúdo, que leva em consideração os demais componentes de um roteador CCN além do CS: a FIB (*Forwarding Information Base*) e a PIT (*Pending Interest Table*). Basicamente, cada componente é modelado individualmente de acordo a taxa de chegada de pacotes de interesse e de dados e da taxa de serviço de cada componente. A partir desse modelo e considerando diferentes métricas, indicam-se que tipos de memória devem ser usados por cada componente do roteador de conteúdo. O CS é avaliado em função da taxa de acerto do *cache* de conteúdos. Intuitivamente, quanto maior a probabilidade de encontrar um conteúdo requisitado no CS, menor a quantidade de pacotes encaminhados para a FIB. No entanto, esse comportamento só é garantido se a memória usada para armazenar a tabela de indexação é suficientemente rápida para processar todos os pacotes que chegam ao CS. Do contrário, pacotes serão encaminhados diretamente para a FIB. Assim, para garantir uma taxa de acerto de 90%, que é similar a de uma CDN atual, devem ser usadas memórias RLDRAM e SRAM para armazenar a tabela de indexação. A PIT é avaliada em função da taxa de chegada de pacotes de interesse, que no pior caso, é igual a taxa de chegada ao CS. Assim, o tamanho estimado para a PIT é de 1,4 Gb considerando um enlace de entrada de 100 Gb/s e por isso pode-se usar memória RLDRAM. Por fim, a busca na FIB é avaliada em função do número de prefixos de nomes

de conteúdo. Verifica-se que até 20 milhões de prefixos são suportados para garantir que seja feita apenas uma busca na memória externa (*off-chip*), considerando uma memória interna (*in-chip*) de 200 Mb. Esse número de prefixos corresponde a 25% dos nomes de estações ativas na Internet atual. Em virtude desses resultados, a principal conclusão dos autores é que os atuais roteadores ao serem adaptados para roteadores de conteúdo podem operar na escala de uma CDN ou de um ISP e ainda não na escala global da Internet.

Os autores também apresentam duas possíveis configurações para roteadores de conteúdo de núcleo e borda. A quantidade adicional de memória para transformar cada roteador é determinada em função da taxa de chegada de pacotes de interesse por segundo. O objetivo da análise é estimar o aumento do custo e do consumo de energia do equipamento. O modelo usado como roteador de borda é um Cisco 7507 que possui 5 interfaces de rede Gigabit Ethernet. Para que ele se torne um roteador de conteúdo e suporte uma taxa de 15 Mpck/s devem ser usados 1 TB de memória High-speed SSD para implementar o armazém de pacotes, 6 GB de DRAM para a tabela de indexação, 70 Mb de RLDRAM para a PIT, 200 Mb de SRAM para a memória interna da FIB e, por fim, 140 MB de memória DRAM para a memória externa da FIB. Esses recursos adicionais elevariam em 195 W o consumo de pico e em aproximadamente USD 30.000 o preço de um equipamento. O modelo do roteador de núcleo é um Cisco CRS 1 com oito interfaces de rede de 40 Gb/s. Para torná-lo um roteador de conteúdo que suporta uma taxa de chegada de 1 Gpck/s e 250 milhões de prefixos, deve-se implementar um armazém de pacotes por interface, cada um com 10 GB de memória DRAM e uma tabela de indexação de 266 Mb duplicadas em dois *chips* RLDRAM, uma PIT por interface com 560 Mb de SRAM, além de 4 Gb de SRAM para implementar a memória interna da FIB e 1,5 GB de DRAM para a memória externa. Esses recursos adicionais elevariam em aproximadamente 3000 W o consumo de pico e em USD 130.000 o preço do equipamento.

Aplicações Conversacionais e em Tempo-Real

As ROCs aumentam a eficiência de aplicações de distribuição de conteúdo. Entretanto, é necessário investigar o desempenho de aplicações tipicamente conversacionais, como o correio eletrônico e a transmissão de voz sobre IP (*Voice over IP*), nas ROCs uma vez que os conteúdos são encaminhados com base em seus nomes e não mais nos endereços IP.

Jacobson *et al.* propõem e implementam uma aplicação de telefonia sobre a arquitetura CCN, chamada de VoCCN [Jacobson et al. 2009b]. O principal objetivo é mostrar a viabilidade de mapear protocolos conversacionais, como o SIP (*Session Initiation Protocol*) e o RTP (*Real-time Transfer Protocol*), em modelos baseados em conteúdo. Para tanto, os autores identificam dois principais problemas: como iniciar uma chamada e como o receptor identifica e responde ao originador. Na Internet atual, um número de porta é usado como ponto de encontro para estabelecimento de chamadas. Para iniciar uma chamada, o originador deve solicitar o estabelecimento de conexão com o receptor, cujo endereço é conhecido, enviando para ele pacotes com esse número de porta. O receptor, por sua vez, envia de volta pacotes de confirmação da abertura, uma vez que que os pacotes recebidos possuem o endereço do originador. Na arquitetura CCN isso não é trivial. É necessário implementar um mecanismo de publicação sob-demanda para iniciar uma chamada. Como visto na Seção 5.3.4.3, a arquitetura CCN não exige que

um conteúdo seja publicado e registrado na infraestrutura antes de ser solicitado. Assim, os usuários enviam requisições para conteúdos que ainda não foram publicados. Essas requisições são encaminhadas para publicadores potenciais que, ao recebê-las, criam e publicam o conteúdo desejado em resposta. Assim, a chamada é iniciada. O segundo problema é fazer com que o publicador consiga enviar conteúdos para o consumidor, uma vez que os pacotes CCN não possuem identificação de quem solicita o conteúdo. A solução é usar nomes construíveis. A idéia é que seja possível construir o nome de um dado pedaço do conteúdo sem ter conhecimento prévio de informações desse conteúdo. Para garantir essa propriedade, é necessário usar um algoritmo determinístico que possibilite ao publicador e ao consumidor construírem o mesmo nome e que o consumidor possa solicitar conteúdos com nomes parcialmente especificados [Jacobson et al. 2009b]. Uma vez garantida a propriedade de nomes construíveis, basta que o publicador receba um pacote de interesse, crie o conteúdo nomeado de acordo com as informações contidas nesse pacote e envie um pacote de dados. Esse pacote será, então, encaminhado de volta ao consumidor pelo caminho reverso construído a partir dos rastros deixados pelo pacote de interesse. Assim, cria-se um fluxo bidirecional de dados entre consumidor e publicador. Resultados experimentais mostram que a aplicação VoCCN é mais simples, segura e escalável do que uma aplicação VoIP equivalente baseada na arquitetura atual. Vale ressaltar ainda que a VoCCN mantém a interoperabilidade com as aplicações atuais, pois usa implementações-padrão dos protocolos SIP e RTP e um *gateway* IP-para-CCN simples e sem estados.

Uma aplicação semelhante à VoCCN, chamada de VoPSI, é proposta por Stais *et al.* para a arquitetura PSIRP [Stais et al. 2011]. Outra aplicação, a ACT (*Audio Conference Tool*), é uma extensão da VoCCN que implementa funcionalidades de audioconferência, como a descoberta de conferências em andamento e de seus participantes [Zhu et al. 2011]. Tsilopoulos e Xylomenos, por sua vez, propõem mecanismos de diferenciação de tráfego para ROCs [Tsilopoulos e Xylomenos 2011]. A idéia é que conteúdos sejam também encaminhados de acordo com seu tipo e não apenas de acordo com seus nomes. Os conteúdos são classificados em documentos e canais. Pedaços de conteúdo sensíveis a perdas são classificados como documentos. Do contrário, são classificados como canais. Além disso, os documentos são divididos em dois grupos: sob demanda e de tempo real. Aplicações multimídias em geral são classificas como canais. Transferência de arquivos, vídeo sobre HTTP e correio eletrônico são documentos sob demanda. Por fim, jogos *online*, salas de bate-papo, mensageiros instantâneos são documentos de tempo real.

5.5. Considerações Finais

O acesso a conteúdos em suas mais diversas formas já representa mais da metade do tráfego atual da Internet [Sandvine 2011]. Somente no Brasil, 53% do tráfego é gerado por redes P2P de compartilhamento de arquivos e por aplicações de distribuição de áudio e vídeo [Sandvine 2011]. Ainda que tais aplicações já representem um grande sucesso em termos de usuários e, em alguns casos, de geração de receita, a distribuição de conteúdo na arquitetura atual da Internet encontra uma série de obstáculos técnicos que aumentam consideravelmente a complexidade de implementação e administração de tais aplicações. Esses obstáculos exigem uma série de "remendos" na Internet para que

as aplicações funcionem, muitas vezes baseados em soluções proprietárias e que também podem comprometer a escalabilidade da rede.

As ROCs inserem-se nesse contexto como substrato de rede alternativo e viável não somente para o desenvolvimento de aplicações de distribuição de conteúdo, mas também para aplicações tipicamente conversacionais. A principal vantagem das ROCs é prover de forma nativa o compartilhamento eficiente de recursos e de dados, mecanismos para aumentar a disponibilidade dos conteúdos, suporte à segurança intrínseca de conteúdos e à mobilidade de usuários. Em geral, as ROCs adotam soluções mais simples do que as propostas para a Internet atual.

As pesquisas em andamento se concentram principalmente na proposta de novas arquiteturas e na avaliação das políticas de *cache* para ROCs. Porém, o desenvolvimento de tais redes remete, ainda, a outros desafios não-técnicos, principalmente relacionados aos interesses das partes envolvidas na distribuição de conteúdos. Há conflitos de interesses nas relações de *peering* entre provedores, falta de estímulo para a adoção de *cache* nas redes de acesso e dificuldade do processo de contabilização no acesso aos conteúdos. A padronização e a interoperabilidade das ROCs é ainda outra questão em aberto. Atualmente, existem grupos de trabalho do IETF para a padronização do armazenamento de pacotes na rede (DECADE) [Song et al. 2012] e interconexão entre CDNs (CDNI) [Niven-Jenkins et al. 2012]. Porém não há inciativas para definir mecanismos de interoperabilidade entre arquiteturas de ROC, como CCN, DONA, etc.

Esse cenário com diversos desafios técnicos e econômico-financeiros é o que torna o desenvolvimento das ROCs um campo de pesquisa promissor e que potencialmente pode resultar em uma mudança radical do paradigma de comunicação da Internet.

Agradecimentos

Este trabalho foi realizado com recursos do CNPq, CAPES, FAPERJ, FINEP, CTIC e FUNTTEL.

Referências

[Ahlgren et al. 2008] Ahlgren, B., D'Ambrosio, M., Marchisio, M., Marsh, I., Dannewitz, C., Ohlman, B., Pentikousis, K., Strandberg, O., Rembarz, R. e Vercellone, V. (2008). Design considerations for a network of information. Em *Re-Architecting the Internet Workshop - ReARCH*, páginas 66:1–66:6.

[Akamai Technologies 2012] Akamai Technologies (2012). Akamai handles a significant portion of world wide web traffic - over a trillion interactions every day. `http://www.akamai.com/html/about/index.html`. Acessado em 12 de março de 2012.

[Allman 2007] Allman, M. (2007). Personal namespaces. Em *ACM Workshop on Hot Topics in Networks - HotNets*.

[Arianfar et al. 2011] Arianfar, S., Koponen, T., Raghavan, B. e Shenker, S. (2011). On preserving privacy in content-oriented networks. Em *ACM SIGCOMM Workshop on Information-Centric Networking - ICN*, páginas 19–24.

[Arianfar et al. 2010a] Arianfar, S., Nikander, P. e Ott, J. (2010a). On content-centric router design and implications. Em *Re-Architecting the Internet Workshop - ReARCH*, páginas 5:1–5:6.

[Arianfar et al. 2010b] Arianfar, S., Nikander, P. e Ott, J. (2010b). Packet-level caching for information-centric networking. Relatório técnico, Aalto University.

[Bhattacharyya 2003] Bhattacharyya, S. (2003). An overview of source-specific multicast (SSM). IETF Network Working Group RFC 3569.

[Biraghi et al. 2011] Biraghi, A. M., Gonçalves, J., Levä, T., Ferreira, R. J., Zhang, N., Correia, L., Sebastião, D., Ohlman, B. e Salo, J. (2011). New business models and business dynamics of the future networks. Relatório Técnico FP7-ICT-2009-5-257448-SAIL/D.A.7, Scalable and Adaptable Internet Solutions (SAIL) Project.

[Breslau et al. 1999] Breslau, L., Cao, P., Fan, L., Phillips, G. e Shenker, S. (1999). Web caching and zipf-like distributions: Evidence and implications. Em *IEEE INFOCOM*, páginas 126–134.

[Broder e Mitzenmacher 2002] Broder, A. e Mitzenmacher, M. (2002). Network applications of Bloom filters: A survey. Em *Internet Mathematics*, páginas 636–646.

[Buyya et al. 2008] Buyya, R., Pathan, M. e Vakali, A. (2008). *Content Delivery Networks*. Springer, 1a. edição.

[Caesar et al. 2006] Caesar, M., Condie, T., Kannan, J., Lakshminarayanan, K. e Stoica, I. (2006). ROFL: routing on flat labels. Em *ACM SIGCOMM*, páginas 363–374.

[Campista et al. 2010] Campista, M. E. M., Ferraz, L. H. G., Moraes, I. M., Lanza, M. L. D., Costa, L. H. M. K. e Duarte, O. C. M. B. (2010). Interconexão de redes na Internet do futuro: Desafios e soluções. Em *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC*, páginas 47–101.

[Cao et al. 2004] Cao, F., e Singh, J. P. (2004). Efficient event routing in content-based publish-subscribe service networks. Em *IEEE INFOCOM*.

[Carofiglio et al. 2011a] Carofiglio, G., Gallo, M. e Muscariello, L. (2011a). Bandwidth and storage sharing performance in information centric networking. Em *ACM SIGCOMM Workshop on Information-Centric Networking - ICN*, páginas 26–31.

[Carofiglio et al. 2011b] Carofiglio, G., Gallo, M., Muscariello, L. e Perino, D. (2011b). Modeling data transfer in content-centric networking. Em *International Teletraffic Congress - ITC*, páginas 111–118.

[Carofiglio et al. 2011c] Carofiglio, G., Gehlen, V. e Perino, D. (2011c). Experimental evaluation of memory management in content-centric networking. Em *IEEE International Communications Conference - ICC*, páginas 1–6.

[Carzaniga et al. 2009] Carzaniga, A., Carughi, G. T., Hall, C. e Wolf, A. L. (2009). Practical high-throughput content-based routing using unicast state and probabilistic encodings. Relatório Técnico 2009/06, Faculty of Informatics, University of Lugano.

[Carzaniga et al. 2000] Carzaniga, A., Rosenblum, D. S. e Wolf, A. L. (2000). Content-based addressing and routing: A general model and its application. Relatório Técnico CU-CS-902-00, Department of Computer Science, University of Colorado.

[Carzaniga et al. 2001] Carzaniga, A., Rosenblum, D. S. e Wolf, A. L. (2001). Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383.

[Carzaniga et al. 2004] Carzaniga, A., Rutherford, M. J. e Wolf, A. L. (2004). A routing scheme for content-based networking. Em *IEEE INFOCOM*, páginas 918–928.

[Carzaniga e Wolf 2003] Carzaniga, A. e Wolf, A. L. (2003). Forwarding in a content-based network. Em *ACM SIGCOMM*, páginas 163–174.

[Castro et al. 2002] Castro, M., Druschel, P., Hu, Y. C. e Rowstron, A. (2002). Exploiting network proximity in distributed hash tables. Em *International Workshop on Future Directions in Distributed Computing - FuDiCo*, páginas 52–55.

[Che et al. 2002] Che, H., Tung, Y. e Wang, Z. (2002). Hierarchical web caching systems: Modeling, design and experimental results. *IEEE Journal on Selected Areas in Communications*, 20(7):1305–1314.

[Che et al. 2001] Che, H., Wang, Z. e Tung, Y. (2001). Analysis and design of hierarchical web caching systems. Em *IEEE INFOCOM*, páginas 1416–1424.

[Cheriton e Gritter 2000] Cheriton, D. e Gritter, M. (2000). TRIAD: A new next generation Internet architecture. Relatório técnico, Computer Science Department, Stanford University.

[Cho et al. 2012] Cho, K., Lee, M., Park, K., Kwon, T. T., Choi, Y. e Pack, S. (2012). WAVE: Popularity-based and collaborative in-network caching for content-oriented networks. Em *Workshop on Emerging Design Choices in Name-Oriented Networking - NOMEN*.

[Chockler et al. 2007] Chockler, G., Melamed, R., Tock, Y. e Vitenberg, R. (2007). SpiderCast: a scalable interest-aware overlay for topic-based pub/sub communication. Em *ACM International Conference on Distributed Event-Based Systems - DEBS*, páginas 14–25.

[Choi et al. 2009] Choi, J., Han, J., Cho, E., Kwon, T., Kim, H. e Choi, Y. (2009). Performance comparison of content-oriented networking alternatives: A hierarchical tree versus a flat distributed hash table. Em *IEEE Conference on Local Computer Networks - LCN*, páginas 253–256.

[Choi et al. 2011] Choi, J., Han, J., Cho, E., Kwon, T. T. e Choi, Y. (2011). A survey on content-oriented networking for efficient content delivery. *IEEE Communications Magazine*, 49(3):121–127.

[Clarke et al. 2001] Clarke, D. E., Elien, J.-E., Ellison, C. M., Fredette, M., Morcos, A. e Rivest, R. L. (2001). Certificate chain discovery in SPKI/SDSI. *Journal of Computer Security*, 9(4):285–322.

[Costa e Duarte 2003] Costa, L. H. M. K. e Duarte, O. C. M. B. (2003). Roteamento multicast na Internet. Em *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC*.

[Dannewitz et al. 2010] Dannewitz, C., Golic, J., Ohlman, B. e Ahlgren, B. (2010). Secure naming for a network of information. Em *IEEE INFOCOM Workshops*, páginas 1–6.

[Deering 1989] Deering, S. (1989). Host extensions for IP multicasting. IETF Network Working Group RFC 1112.

[Diallo et al. 2011] Diallo, M., Fdida, S., Sourlas, V., Flegkas, P. e Tassiulas, L. (2011). Leveraging caching for Internet-scale content-based publish/subscribe networks. Em *IEEE International Communications Conference - ICC*, páginas 1–5.

[Eugster et al. 2003] Eugster, P., Felber, P., Guerraoui, R. e Kermarrec, A. (2003). The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2):114–131.

[Feamster et al. 2007] Feamster, N., Gao, L. e Rexford, J. (2007). How to lease the Internet in your spare time. *ACM SIGCOMM Computer Communication Review*, 37(1):61–64.

[Fotiou et al. 2010] Fotiou, N., Nikander, P., Trossen, D. e Polyzos, G. (2010). Developing information networking further: From PSIRP to PURSUIT. Em *ICST BROADNETS*.

[Fricker et al. 2012] Fricker, C., Robert, P., Roberts, J. e Sbihi, N. (2012). Impact of traffic mix on caching performance in a content-centric network. Em *Workshop on Emerging Design Choices in Name-Oriented Networking - NOMEN*.

[Ganesan et al. 2004] Ganesan, P., Gummadi, K. e Garcia-Molina, H. (2004). Canon in G major: Designing DHTs with hierarquical structure. Em *International Conference on Distributed Computing Systems - ICDCS*, páginas 263–272.

[Gao et al. 2006] Gao, Y., Deng, L., Kuzmanovic, A. e Chen, Y. (2006). Internet cache pollution attacks and countermeasures. Em *IEEE International Conference on Network Protocols - ICNP*, páginas 54–64.

[Ghodsi et al. 2011a] Ghodsi, A., Koponen, T., Raghavan, B., Shenker, S., Singla, A. e Wilcox, J. (2011a). Information-centric networking: seeing the forest for the trees. Em *ACM Workshop on Hot Topics in Networks - HotNets*, páginas 1:1–1:6.

[Ghodsi et al. 2011b] Ghodsi, A., Koponen, T., Rajahalme, J., Sarolahti, P. e Shenker, S. (2011b). Naming in content-oriented architectures. Em *ACM SIGCOMM Workshop on Information-Centric Networking - ICN*, páginas 1–6.

[Holbrook e Cain 2006] Holbrook, H. e Cain, B. (2006). Source-specific multicast for IP.

[Jacobson et al. 2009a] Jacobson, V., Smetters, D., Thornton, J., Plass, M., Briggs, N. e Braynard, R. (2009a). Networking named content. Em *International Conference on emerging Networking EXperiments and Technologies - CoNEXT*.

[Jacobson et al. 2009b] Jacobson, V., Smetters, D. K., Briggs, N. H., Plass, M. F., Stewart, P., Thornton, J. D. e Braynard, R. L. (2009b). VoCCN: voice-over content-centric networks. Em *Re-Architecting the Internet Workshop - ReARCH*, páginas 1–6.

[Jacobson et al. 2012] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M., Briggs, N. e Braynard, R. (2012). Networking named content. *Communications of the ACM*, 55(1):117–124.

[Jokela et al. 2009] Jokela, P., Zahemszky, A., Arianfar, S., Nikander, P. e Esteve, C. (2009). LIPSIN: Line speed publish/subscribe inter-networking. Em *ACM SIGCOMM*, páginas 195–206.

[Katsaros et al. 2011] Katsaros, K. V., Xylomenos, G. e Polyzos, G. C. (2011). MultiCache: An overlay architecture for information-centric networking. *Computer Networks*, 55(4):936–947.

[Koponen et al. 2007] Koponen, T., Shenker, S., Stoica, I., Chawla, M., Chun, B., Ermolinsky, A. e Kim, K. (2007). A data-oriented (and beyond) network architecture. Em *ACM SIGCOMM*, páginas 181–192.

[Kumar et al. 2005] Kumar, A., Xu, J. e Zegura, E. W. (2005). Efficient and scalable query routing for unstructured peer-to-peer networks. Em *IEEE INFOCOM*, páginas 1162–1173.

[Lagutin et al. 2010] Lagutin, D., Visala, K. e Tarkoma, S. (2010). Publish/subscribe for Internet: PSIRP perspective. Em *Towards the Future Internet - Emerging Trends from European Research*, chapter 8, páginas 75–84. IOS Press.

[Lee et al. 2011] Lee, M., Cho, K., Park, K., Kwon, T. e Choi, Y. (2011). SCAN: Scalable content routing for content-aware networking. Em *IEEE International Communications Conference - ICC*, páginas 1–5.

[Lu et al. 2004] Lu, Y., Abdelzaher, T. F. e Saxena, A. (2004). Design, implementation, and evaluation of differentiated caching services. *IEEE Transactions on Parallel and Distributed Systems*, 15(5):440–452.

[Majumder et al. 2009] Majumder, A., Shrivastava, N., Rastogi, R. e Srinivasan, A. (2009). Scalable content-based routing in pub/sub systems. Em *IEEE INFOCOM*, páginas 567–575.

[Martins e Duarte 2010] Martins, J. L. e Duarte, S. (2010). Routing algorithms for content-based publish/subscribe systems. *IEEE Communications Surveys and Tutorials*, 12(1):39–58.

[Mealling e Denenberg 2002] Mealling, M. e Denenberg, R. (2002). Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations. IETF Network Working Group RFC 3305.

[Moraes et al. 2008] Moraes, I. M., Campista, M. E. M., Moreira, M. D. D., Rubinstein, M. G., Costa, L. H. M. K. e Duarte, O. C. M. B. (2008). Distribuição de vídeo sobre redes par-a-par: Arquiteturas, mecanismos e desafios. Em *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC*, páginas 115–171.

[Niven-Jenkins et al. 2012] Niven-Jenkins, B., Faucheur, F. L. e Bitar, N. (2012). Content Distribution Network Interconnection (CDNI) problem statement. IETF Network Working Group Internet-Draft (Work In Progress).

[Passarella 2012] Passarella, A. (2012). A survey on content-centric technologies for the current Internet: CDN and P2P solutions. *Computer Communications*, 35(1):1–32.

[Perino e Varvello 2011] Perino, D. e Varvello, M. (2011). A reality check for content centric networking. Em *ACM SIGCOMM Workshop on Information-Centric Networking - ICN*, páginas 44–49.

[Peyravian et al. 1998] Peyravian, M., Roginsky, A. e Kshemkalyani, A. D. (1998). On probabilities of hash value matches. *Computers and Security*, 17(2):171–176.

[Plagemann et al. 2005] Plagemann, T., Goebel, V., Mauthe, A., Mathy, L., Turletti, T. e Urvoy-Keller, G. (2005). From content distribution networks to content networks - issues and challenges. *Internation Journal for the Computer and Telecommunications Industry*, 29:551–566.

[Podlipnig e Böszörmenyi 2003] Podlipnig, S. e Böszörmenyi, L. (2003). A survey of web cache replacement strategies. *ACM Computing Surveys*, 35(4):374–398.

[Popescu et al. 2005] Popescu, B. C., van Steen, M., Crispo, B., Tanenbaum, A. S., Sacha, J. e Kuz, I. (2005). Securely replicated web documents. Em *IEEE International Parallel and Distributed Processing Symposium - IPDPS*, páginas 104b–104b.

[Psaras et al. 2011] Psaras, I., Clegg, R. G., Landa, R., Chai, W. K. e Pavlou, G. (2011). Modelling and evaluation of CCN-caching trees. Em *IFIP NETWORKING'11*.

[Ratnasamy et al. 2001] Ratnasamy, S., Francis, P., Handley, M., Karp, R. e Shenker, S. (2001). A scalable content-addressable network. Em *ACM SIGCOMM*, páginas 161–172.

[Ratnasamy et al. 2002] Ratnasamy, S., Stoica, I. e Shenker, S. (2002). Routing algorithms for DHTs: Some open questions. Em *International Workshop on Peer-to-Peer Systems - IPTPS*, páginas 45–52.

[Rosensweig e Kurose 2009] Rosensweig, E. J. e Kurose, J. (2009). Breadcrumbs: Efficient, best-effort content location in cache networks. Em *IEEE INFOCOM*, páginas 2631–2635.

[Rosensweig et al. 2010] Rosensweig, E. J., Kurose, J. e Towsley, D. (2010). Approximate models for general cache networks. Em *IEEE INFOCOM*, páginas 1–9.

[Rossi e Rossini 2011a] Rossi, D. e Rossini, G. (2011a). Caching performance of content centric networks under multi-path routing (and more). Relatório técnico, Telecom ParisTech.

[Rossi e Rossini 2011b] Rossi, D. e Rossini, G. (2011b). A dive into the caching performance of content centric networking. Relatório técnico, Telecom ParisTech.

[Rossi e Rossini 2012] Rossi, D. e Rossini, G. (2012). On sizing CCN content store by exploiting topological information. Em *Workshop on Emerging Design Choices in Name-Oriented Networking - NOMEN*.

[Rowstron e Druschel 2001] Rowstron, A. e Druschel, P. (2001). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. Em *IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg - Middleware*, páginas 329–350.

[Saltzer et al. 1984] Saltzer, J. H., Reed, D. P. e Clark, D. D. (1984). End-to-end arguments in system design. *ACM Transactions on Computer Systems*, 2(4):277–288.

[Sandvine 2011] Sandvine (2011). Global Internet phenomena report. Relatório técnico, Sandvine.

[Singla et al. 2010] Singla, A., Godfrey, B., Fall, K. R., Iannaccone, G. e Ratnasamy, S. (2010). Scalable routing on flat names. Em *International Conference on emerging Networking EXperiments and Technologies - CoNEXT*, páginas 20:1–20:12.

[Smetters e Jacobson 2009] Smetters, D. e Jacobson, V. (2009). Securing network content. Relatório Técnico TR-2009-1, Xerox Palo Alto Research Center - PARC.

[Song et al. 2012] Song, H., Zong, N., Yang, Y. e Alimi, R. (2012). DECoupled Application Data Enroute (DECADE) problem statement. IETF Network Working Group Internet-Draft (Work In Progress).

[Sourlas et al. 2011] Sourlas, V., Flegkas, P., Paschos, G. S., Katsaros, D. e Tassiulas, L. (2011). Storage planning and replica assignment in content-centric publish/subscribe networks. *Computer Networks*, 55(18):4021–4032.

[Stais et al. 2011] Stais, C., Diamantis, D., Aretha, C. e Xylomenos, G. (2011). VoPSI: Voice over a publish-subscribe internetwork. Em *Future Network and Mobile Summit - FutureNetw*, páginas 1–8.

[Stoica et al. 2003] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F. e Balakrishnan, H. (2003). Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking*, 11(1):17–32.

[Trossen e Biczók 2010] Trossen, D. e Biczók, G. (2010). Not paying the truck driver: Differentiated pricing for the future internet. Em *Re-Architecting the Internet Workshop - ReARCH*, páginas 1–6.

[Trossen et al. 2010] Trossen, D., Sarela, M. e Sollins, K. (2010). Arguments for an information-centric internetworking architecture. *ACM SIGCOMM Computer Communication Review*, 40(2):26–33.

[Tsilopoulos e Xylomenos 2011] Tsilopoulos, C. e Xylomenos, G. (2011). Supporting diverse traffic types in information centric networks. Em *ACM SIGCOMM Workshop on Information-Centric Networking - ICN*, páginas 13–18.

[Visala et al. 2009] Visala, K., Lagutin, D. e Tarkoma, S. (2009). LANES: An inter-domain data-oriented routing architecture. Em *Re-Architecting the Internet Workshop - ReARCH*, páginas 55–60.

[Walfish et al. 2004] Walfish, M., Balakrishnan, H. e Shenker, S. (2004). Untangling the web from DNS. Em *USENIX/ACM Symposium on Networked Systems Design and Implementation - NSDI*, páginas 17–17.

[Wang 1999] Wang, J. (1999). A survey of web caching schemes for the Internet. *ACM SIGCOMM Computer Communication Review*, 29(5):36–46.

[Wang et al. 2004] Wang, L., Park, K. S., Pang, R., Pai, V. e Peterson, L. (2004). Reliability and security in the CoDeeN content distribution network. Em *USENIX Annual Technical Conference - ATC*, páginas 14–14.

[Wang et al. 2005] Wang, X., Yin, Y. L. e Yu, H. (2005). Finding collisions in the full SHA-1. Em *CRYPTO*, volume 3621, páginas 17–36.

[Xie et al. 2012] Xie, M., Widjaja, I. e Wang, H. (2012). Enhancing cache robustness for content-centric networking. Em *IEEE INFOCOM*.

[Zhang et al. 2010] Zhang, L., Estrin, D., Burke, J., Jacobson, V., Thornton, J., Smetters, D. K., Zhang, B., Tsudik, G., Claffy, K., Krioukov, D., Massey, D., Papadopoulos, C., Abdelzaher, T., Wang, L., Crowley, P. e Yeh, E. (2010). Named Data Networking (NDN) project. Relatório Técnico NDN-0001, Xerox Palo Alto Research Center - PARC.

[Zhang et al. 2011] Zhang, N., Levä, T. e Hämmäinen, H. (2011). Two-sidedness of internet content delivery. Em *IEEE Conference on Telecommunications Internet and Media Techno-Economics - CTTE*, páginas 16–18.

[Zhu et al. 2011] Zhu, Z., Wang, S., Yang, X., Jacobson, V. e Zhang, L. (2011). ACT: audio conference tool over named data networking. Em *ACM SIGCOMM Workshop on Information-Centric Networking - ICN*, páginas 68–73.