



Pós-Graduação em Ciência da Computação

EDSON ADRIANO MARAVALHO AVELAR

**MOBCACHE: MOBILIDADE DE PROVEDOR E
ARMAZENAMENTO EFICIENTE EM REDES SEM
FIO ORIENTADAS A CONTEÚDO**



Universidade Federal de Pernambuco

posgraduacao@cin.ufpe.br

www.cin.ufpe.br/~posgraduacao

RECIFE
2017

EDSON ADRIANO MARAVALHO AVELAR

**MOBCACHE: MOBILIDADE DE PROVEDOR E
ARMAZENAMENTO EFICIENTE EM REDES SEM FIO
ORIENTADAS A CONTEÚDO**

Este trabalho foi apresentado à Pós-Graduação em
Ciência da Computação do Centro de Informática da
Universidade Federal de Pernambuco como requisito
parcial para obtenção do grau de Doutor em Ciência da
Computação.

*ORIENTADOR(A): PROF. DR.
KELVIN LOPES DIAS*

*RECIFE
2017*

Catálogo na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

A948m Avelar, Edson Adriano Maravalho
MobCache: mobilidade de provedor e armazenamento eficiente em redes
sem fio orientadas a conteúdo / Edson Adriano Maravalho Avelar. – 2017.
119f.:il, fig., tab.

Orientador: Kelvin Lopes Dias.

Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da
Computação, Recife, 2017.

Inclui referências.

1. Redes de computadores. 2. Redes sem fio. I. Dias, Kelvin Lopes
(orientador). II. Título.

004.6 CDD (23. ed.) UFPE- MEI 2017-160

Edson Adriano Maravalho Avelar

**MobCache: MOBILIDADE DE PROVEDOR E
ARMAZENAMENTO EFICIENTE EM REDES SEM FIO
ORIENTADAS A CONTEÚDO**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 03/03/2017.

Orientador: Prof. Dr. Kelvin Lopes Dias

BANCA EXAMINADORA

Prof. Dr. Paulo Roberto Freire Cunha
Centro de Informática / UFPE

Prof. Dr. José Augusto Suruagy Monteiro
Centro de Informática / UFPE

Prof. Dr. Nelson Souto Rosa
Centro de Informática / UFPE

Prof. Dr. Edmundo Roberto Mauro Madeira
Instituto de Computação/ UNICAMP

Prof. Dr. Jó Ueyama
Instituto de Ciências Matemáticas e de Computação/ USP

Resumo

O paradigma ICN (*Information Centric Network*) propõe que as operações da rede tenham o foco na recuperação do conteúdo ou informação, não importando sua localização física. Esta arquitetura difere substancialmente do projeto original da Internet, em vigor, centrado no *host* e dependente do endereçamento IP. O armazenamento de conteúdo nos roteadores da rede e não apenas no provedor/fonte de conteúdo, além do suporte intrínseco à mobilidade do consumidor de conteúdo, são algumas das características que tornam o ICN uma das soluções proeminentes para lidar com a explosão tanto do tráfego de vídeo na Internet quanto do acesso via dispositivos móveis. Apesar do suporte à mobilidade de consumidor no ICN, há questões em aberto para que as operadoras invistam na implantação do ICN em suas redes, como o suporte transparente à mobilidade do provedor. Nesse contexto, esta tese investiga e propõe soluções para dois problemas tratados isoladamente pela literatura da área de redes centradas na informação: (a) decisão de armazenamento nos elementos da rede (*in-network caching*) e (b) mobilidade transparente de provedor de conteúdo. Dessa forma, propõe-se a sinergia entre mobilidade e *caching* com o objetivo de solucionar o problema de mobilidade de provedor em cenários fim-a-fim. Primeiramente, foi desenvolvida uma solução para cada problema. A solução para (a) consistiu em um modelo binomial e um algoritmo para ranqueamento de *cache* baseado no número de acertos; a avaliação, utilizando o simulador Omnet++, mostrou ganhos de 10% a 30% em comparação com algumas das estratégias mais citadas na literatura usando métricas de taxa de acerto, número de saltos e atraso médio de pacotes. A solução para (b) foi baseada em um esquema de envio de pacotes especiais para atualização das rotas de forma rápida e direta, após o *handover*. Um modelo analítico para analisar o custo do *handover* da proposta foi desenvolvido e validado via simulação. Diversas soluções da literatura foram comparadas em termos de métricas de QoS (*Quality of Service*). Em relação à vazão, os ganhos da proposta variam de 3%, com o provedor movendo-se a 1m/s, e 244%, com o provedor movendo-se a 30m/s. Além disso, foi demonstrado na tese que há relação entre mobilidade e *caching* em redes ICN e que é possível melhorar ainda mais os serviços se a solução para os problemas (a) e (b) for desenvolvida de forma conjunta. Com isso, foi proposto o MobCache, um arcabouço para tratar a mobilidade de provedor nas CCNs (*Content Centric Networks*) visando melhorar a QoS e QoE (*Quality of Experience*) para tráfego de vídeo, levando-se em consideração tanto a parte cabeada quanto a sem fio da rede. Os resultados de QoS e QoE mostraram a superioridade da proposta em relação ao estado da arte, com ganhos de até 58% em relação à vazão e de 16% a 80% em relação à avaliação de qualidade de vídeo, em particular o PSNR (*Peak Signal to Noise Ratio*).

Palavras-chave: ICN. CCN. Armazenamento nos elementos da Rede. Mobilidade de Provedor.

Abstract

The Information Centric Network (ICN) paradigm proposes that network operations focus on the content retrieval, regardless of its physical location. Such architecture differs from the original Internet project, i.e., host-centric and IP-address-dependent. Storing content on routers rather than on the content source, as well as the intrinsic support for the consumer's mobility, are some features that make ICN one of the prominent solutions to deal with the boost in both video and mobile traffic. New standards such as those for Vehicular Ad Hoc Networks (VANETs) created the need to address new requirements such as efficient content delivery while moving between wireless access points. For instance, smart vehicles may share data so as to report bad weather, accidents or even transmit multimedia content. Despite the mobility support given for the consumer, there are open issues for operators in the ICN deployment, such as transparent support for provider mobility. This thesis investigates and proposes separate solutions for two problems: (a) network equipment storing decision (in-network caching) and (b) transparent mobility for content providers. It is proposed the synergy between mobility and caching, aiming to solve the problem of provider mobility in end-to-end scenarios. The solution to (a) consisted of a model and cache ranking algorithm that is based on the number of hits. The adopted simulation tool was the Omnet++ simulator and our results showed gains of 10% to 30% compared to some of the most cited strategies, using hit rate, number of jumps and average packet delay as the main performance metrics. The solution to (b) aimed at fast route update (after the *handover* events) and was based on a scheme for special package forwarding. An analytical model to analyze the proposal *handover* cost was developed and validated through simulation. In terms of flow, the proposal gains range from 3%, with the provider moving at 1m/s, and 244%, with the provider moving at 30m /s. In addition, it has been demonstrated a relationship between mobility and caching in ICN networks and that it is possible to further improve services if the solution to problems (a) and (b) are tackled simultaneously. MobCache, a framework for addressing the provider mobility in CCNs (Content Centric Network) was proposed to improve QoS (Quality of Service) and QoE (Quality of Experience) for video traffic, taking into account both wired and wireless network. The results showed the proposal's superiority, with gains of up to 58% and considering QoS and QoE, MobCache achieved 58% in terms of flow rate and 16% to 80% in video quality evaluation, in particular PSNR (*Peak Signal to Noise Ratio*). In addition, the proposal was also evaluated through a mathematical model, presenting gains that ranged from 150% to 300% in this scenario. The model was validated by means of software simulation, resulting in a similar output for a single input data, in both methods.

Keywords: ICN. CCN. In-network Caching. Provider Mobility.

Lista de Figuras

Figura 1.1 Arquitatura de comunicação entre dispositivos CCN móveis.	16
Figura 2.1 Modelo Ampulheta do CCN.	21
Figura 2.2 Comparação redes convencionais e CCN.....	22
Figura 2.3 Sinalização OSPFN	24
Figura 2.4 Comunicação Consumidor-Provedor antes da mobilidade do consumidor.....	27
Figura 2.5 Comunicação Consumidor-Provedor após mobilidade do Consumidor	28
Figura 2.6 Comunicação Consumidor/Provedor antes da mobilidade do Provedor.	28
Figura 2.7 Falha ao encontrar o conteúdo após a mobilidade do Provedor	29
Figura 3.1 Diagrama das propostas de mobilidade de provedor e seus grupos.	35
Figura 3.2 Sinalização genérica para o grupo AD-FIB.....	36
Figura 3.3 Sinalização Genérica para o Grupo I-MIP.....	37
Figura 3.4 Sinalização Genérica para Entidades Centralizadas	40
Figura 3.5 Arquitetura Genérica de comparação	44
Figura 4.1 Topologia da Rede IPê implementada no simulador (RNP, 2017).....	55
Figura 4.2 Taxa de Acertos comparada com a capacidade dos <i>caches</i> variando de 100 MBytes a 900 MBytes.....	56
Figura 4.3 Número de saltos até um acerto em relação à capacidade dos <i>caches</i> variando de 100 MBytes a 900 MBytes.	56
Figura 4.4 Número de Acertos no servidor em relação à capacidade dos <i>caches</i> variando de 100 MBytes a 900 MBytes.	57
Figura 4.5 RTT médio em relação à capacidade dos <i>caches</i> variando de 100 MBytes a 900 MBytes.....	57
Figura 4.6 Número de acertos no servidor em relação ao expoente Zipf α , onde α varia de 5×10^{-1} a 10×10^{-1} . Tamanho do <i>cache</i> igual a 900MB	58
Figura 4.7 RTT médio em relação ao expoente Zipf α , onde α varia de 5×10^{-1} a 10×10^{-1} . Tamanho do <i>cache</i> igual a 900MB	58
Figura 4.8 Valor médio dos pacotes perdidos, em 15 simulações, com modelo de erro de bit igual a 1×10^{-6}	59
Figura 5.1 Cenário Base para explicação da proposta de mobilidade mostrando a Tabela de Sessão	61
Figura 5.2 Sinalização da Proposta de Mobilidade de Provedor.....	63
Figura 5.3 Interesse Virtual.....	63
Figura 5.4 Vazão com e sem a proposta com velocidade do usuário a 1m/s.....	65
Figura 5.5 Vazão com e sem a proposta com velocidade do usuário a 10m/s.....	65
Figura 5.6 Vazão com e sem a proposta com velocidade do usuário a 30m/s.....	66
Figura 5.7 Vazão média calculada das 20 repetições para cada estratégia e velocidade.	66
Figura 5.8 Gráfico de perda de pacotes variando a velocidade de provedor.	67

Figura 6.1 Cenário de Simulação Mobilidade e <i>Caching</i> extraído do simulador	69
Figura 6.2 Taxa média de <i>Hits</i>	70
Figura 6.3 Área de Aceitação para o teste Kolmogorov-Smirnov	71
Figura 6.4 Gráfico de Probabilidade do Teste KS para o RankCache	73
Figura 6.5 Gráfico de Probabilidade do Teste KS para o <i>ProbCache</i>	73
Figura 6.6 Gráfico de Probabilidade do Teste KS para o LCE	74
Figura 6.7 Gráfico de Probabilidade do Teste KS para o LCD	74
Figura 6.8 Gráfico de Probabilidade do Teste KS para o Fix-0.3.....	75
Figura 6.9 Gráfico de Probabilidade do Teste KS para o Fix-0.7.....	75
Figura 6.10 Gráfico de Probabilidade do Teste KS para o Betweenness Centrality.....	76
Figura 6.11 Teste T-Emparelhado entre o RankCache e as outras propostas	77
Figura 6.12 Resultado do teste T-Emparelhado das estratégias com e sem mobilidade.....	79
Figura 6.13 Boxplot da vazão das estratégias de <i>caching</i> com mobilidade.....	80
Figura 6.14 Boxplot da vazão das estratégias de <i>caching</i> sem mobilidade	80
Figura 7.1 Sinalização do MobCache	83
Figura 7.2 Nova Tabela de Sessão (estendida da Figura 5.1) com informações de UP e 	85
Figura 7.3 Gráfico de vazão em cada instância de simulação.....	86
Figura 7.4 <i>Boxplot</i> comparativo da vazão - Cenário 1	86
Figura 7.5 Soma de pacotes de Interesse retransmitidos pelo Provedor	87
Figura 7.6 Número de Pacotes de Controle durante as simulações	88
Figura 7.7 Tamanho do buffer vs perda de quadros (frames)	90
Figura 7.8 PSNR do MobCache com e sem ordenação	90
Figura 7.9 PSNR de todas as propostas	91
Figura 7.10 Pior quadro de cada proposta recuperado de uma das simulações	92
Figura 7.11 Modelo Random Waypoint.....	93
Figura 7.12 Mensagens utilizadas para o cálculo do custo de <i>handover</i>	96
Figura 7.13 Custo de <i>handover</i> em relação à probabilidade de erro do MobCache	100
Figura 7.14 Custo total de <i>Handover</i> em Relação à velocidade do Provedor.....	101
Figura 7.15 Custo total de <i>handover</i> em relação à taxa de transmissão.....	102
Figura 7.16 Custo de <i>handover</i> em relação ao custo de transmissão entre Provedor e Consumidor.....	103
Figura 7.17 Comparação entre custo de <i>handover</i> na Modelagem e na Simulação do MobCache	104
Figura 7.18 Intervalo das médias com 95% de confiança	105
Figura 7.19 Boxplot das diferenças entre as médias	105

Lista de Tabelas

Tabela 2.1 Prefixos de Nomes	24
Tabela 3.1 Tabela comparativa dos grupos de soluções	42
Tabela 3.2 Lista de Parâmetros Específicos.....	44
Tabela 3.3 Tabelas comparativas das soluções de Mobilidade de Provedor	45
Tabela 4.1 Configuração dos parâmetros dos experimentos.....	54
Tabela 5.1 Parâmetros usados nas simulações.....	64
Tabela 5.2 Estatística descritiva dos resultados da avaliação	66
Tabela 6.1 Parâmetros de Simulação da Mobilidade e <i>Caching</i>	69
Tabela 6.2 D Crítico do teste Komolgorov-Smirnov	72
Tabela 6.3 Resumo do Teste Kolmogorov-Smirnov.....	78
Tabela 7.1 Estatística descritiva do número de pacotes retransmitidos	87
Tabela 7.2 Qualidade estimada de vídeo a partir do PSNR.	89
Tabela 7.3 Estatística descritiva da simulação do PSNR usando as estratégias de mobilidade.....	92
Tabela 7.4 Resumo dos parâmetros	99
Tabela 7.5 Tabela de parâmetro para o Modelo e para a Simulação	103

Abreviações e Acrônimos

AD-FIB	Atualização Direta da FIB
BTW	Betweenness Centrality
CapEx	Capital Expenditures
CCN	Content Centric Network
CCND	Content Centric Network Daemon
CDN	Content Distribution Network
CR	Content Router
CS	Content Source
DP	Domain Proxy
E-C	Entidade Centralizada
EIGRP	Enhanced Interior Gateway Routing Protocol
FIB	Forwarding Information Base
ICN	Information Centric Network
I-MIP	Inspirado no MIP
IoT	Internet of Things
IP	Internet Protocol
ISP	Internet Service Provider
LCD	Leavy Copy Down
LCE	Leavy Copy Everywhere
LFU	Least frequently Used
LMA	Localized Mobility Anchor
LRU	Least Recently Used
MIP	Mobile IP
NDN	Named-Data Networking
NSF	National Science Foundation
OLSA	Opaque Link State Advertisements
OSPF	Open Shortest Path First
OSPFd	OSPF Daemon
OSPFN	OSPF for Named data
P2P	Peer-to-Peer
PBMA	Proxy Based Mobility Approach
PD	Ponto de Destino
PIT	Pending Information Table
PMC	Publisher Mobility Support Protocol
PMIP	Proxy MIP
PMIP	Proxy MIP
PNPCCN	Proactive Neighbor Pushing CCN
PO	Ponto de Origem
PoA	Point of Attachments
QoE	Quality of Experience
QoS	Quality of Service
RCPI	Received Channel Power Indicator
RH	Resource Handler

RIP	Routing Information Protocol
RP	Rendezvous Point
RSSI	Received Signal Strength Indicator
RTT	Round Trip Time
SDN	Software Defined Network
SD-PAM	Domain Proxies based Provider Mobility Management
SLA	Service Level Agreement
TCP	Transmission Control Protocol
TSB	Time Since Birth
TSI	Time Since Inception
TTL	Time To Live
UMTS	Universal Mobile Telecommunication System
WSI	Wilson Score Interval

Sumário

1	INTRODUÇÃO	13
1.1.	Motivação e Contextualização	13
1.2.	Problema	16
1.3.	Objetivos.....	18
1.3.1.	Objetivo Geral	18
1.3.2.	Objetivos Específicos.....	18
1.4.	Estrutura da Tese	19
2	REFERENCIAL CONCEITUAL	20
2.1.	Introdução às Redes Orientadas a Conteúdo.....	20
2.2.	Armazenamento Interno de Conteúdo nos elementos da rede CCN	22
2.3.	Disseminação de prefixos	23
2.4.	Mobilidade.....	25
2.4.1.	Mobilidade TCP/IP	25
2.4.2.	Mobilidade em Redes CCN	26
2.4.3.	Mobilidade do Consumidor	26
2.4.3.1.	<i>Mobilidade do Provedor</i>	<i>28</i>
2.5.	Considerações Finais	29
3	ESTADO DA ARTE	31
3.1.	Armazenamento no CCN	31
3.2.	Mobilidade.....	33
3.2.1.	Soluções para a mobilidade de provedor em CCN.....	34
3.2.1.1.	<i>Atualização Direta da FIB (AD-FIB)</i>	<i>35</i>
3.2.1.2.	<i>Abordagens Baseadas no protocolo MIP (I-MIP)</i>	<i>37</i>
3.2.1.3.	<i>Entidades Centralizadas (E-C).....</i>	<i>39</i>
3.2.2.	Comparação das Estratégias de Mobilidade	43
3.3.	Considerações Finais.....	47
4	ALGORITMO DE RANQUEAMENTO DE <i>CACHE</i>	48
4.1.	Introdução.....	48
4.2.	Algoritmo de Armazenamento em redes CCN baseado no <i>Ranking</i> do <i>Cache</i>.	49

4.2.1.	Intervalo de Confiança para a proporção binomial	49
4.2.2.	RankCache.....	51
4.3.	Avaliação do RankCache	54
4.4.	Considerações Finais	59
5	MOBILIDADE DE PROVEDOR EM REDES CCN.....	60
5.1.	Proposta de Mobilidade de Provedor.....	60
5.2.	Avaliação.....	64
5.3.	Considerações Finais	67
6	MOBILIDADE E <i>CACHE</i>	68
6.1.	Introdução	68
6.2.	Resultados.....	70
6.3.	Considerações Finais	80
7	MOBCACHE	81
7.1.	Introdução	81
7.2.	Avaliação de Desempenho.....	85
7.2.1.	Avaliação por meio de simulação.....	85
7.2.1.1.	<i>Avaliação de QoS (Quality of Service).</i>	85
7.2.1.2.	<i>Avaliação de QoE (Quality of Experience).</i>	88
7.2.2.	Avaliação analítica	93
7.2.2.1.	<i>Resultados do Modelo</i>	99
7.2.3.	Validação do Modelo Analítico.....	103
7.3.	Considerações Finais.....	106
8	CONSIDERAÇÕES FINAIS.	107
8.1.	Contribuições.....	108
8.2.	Trabalhos Futuros.....	108
	REFERÊNCIAS	112

1

INTRODUÇÃO

Este capítulo apresenta os direcionamentos que motivaram e guiaram o estudo presente nesta tese. Inicialmente, o capítulo aborda a contextualização, mostrando que os protocolos tradicionais estão tendo dificuldade em manter as demandas das redes atuais. Posteriormente, o capítulo mostra que existem soluções mais eficientes na literatura, mas que podem ser melhoradas. Em seguida, são apresentadas as soluções propostas, depois os objetivos e finaliza com a estrutura da tese.

1.1. Motivação e Contextualização

A Internet foi idealizada para resolver um problema fundamental: conexão de dados entre computadores e/ou entre computadores e terminais. Desde sua concepção, a Internet segue o modelo básico onde a comunicação é centrada na troca de informações entre *hosts* (Cerf, 2005). E, para tal, algoritmos de roteamento baseiam suas operações na localização física das máquinas comunicantes. Nesse modelo de comunicação, a localização do *host*, onde está o conteúdo, era o foco da maioria dos protocolos. Ao longo dos anos, o objetivo principal da Internet evoluiu para a busca pelo conteúdo (Cheriton, 2000), não mais importando sua localização física. Esses conteúdos incluem vídeos sob demanda, *streaming* de áudio/vídeo, páginas web, redes sociais, jogos online entre outros. É previsto que só o tráfego de vídeo represente de 80% a 90% de todo tráfego IP em 2018 (Cisco, 2015). Em 2016, a Netflix, uma das maiores fornecedoras de serviços de vídeo sob demanda do Brasil, possuía mais de 60 milhões de assinantes (Chambers, 2016). A preocupação dessa grande quantidade de usuários está no consumo do conteúdo com velocidade e qualidade, não importando onde a informação esteja localizada. Portanto, na Internet atual o foco principal está no consumo do conteúdo e não mais em sua localização física.

O modo como os usuários buscam conteúdo também sofreu uma grande mudança. O crescimento exponencial do número de equipamentos móveis nos últimos anos mostra que o dispositivo que acessa o conteúdo deixou de ser grande e fixo (*desktops* e *notebooks*) e passou a ser pequeno e móvel (*smartphones* e *tablets*). Estudos calculam que, em 2018, os dispositivos móveis serão responsáveis por cerca de 60% de todo tráfego (Cisco, 2015). Como a Internet não foi projetada para esse cenário móvel, torna-se cada vez mais evidente que os protocolos convencionais, desenhados para suportar comunicação *host-to-host*, falham em prover comunicação confiável em redes móveis, pois os usuários ainda sofrem com constantes interrupções de serviço, falhas de enlaces, comunicação insegura e consumo exacerbado de energia dos dispositivos.

Uma das possíveis soluções para a questão apresentada acima é o paradigma ICN (*Information Centric Network*) (Ahlgren, 2012). O ICN é uma arquitetura onde o foco das operações da rede está na recuperação do conteúdo, não importando sua localização física. É uma tecnologia relativamente nova e possui vários trabalhos com o foco em diversos campos de pesquisa (V. Jacobson, 2009) (P. Jokela, 2009) (Trossen, 2011) (Mosko, 2015) (Yaqub, 2016). Dentre as diversas propostas de arquitetura ICN, a CCN (*Content Centric Network*) (Jacobson V. e., 2009) se destacou das demais por seu alto grau de maturidade e do tamanho da comunidade de desenvolvimento em torno de sua implementação oficial (CCNx.org).

O CCN foi um dos projetos integrantes do NDN (*Named-Data Networking*) (Saxena, 2016). O NDN é um dos cinco grandes projetos de pesquisa, financiados pela NSF (*National Science Foundation*) (NSF, 2017), que tem como objetivo redesenhar a infraestrutura da Internet atual. Originalmente, o NDN usava a implementação oficial do CCN (ccnx.org), mas, em 2013, começou a criar seus próprios plug-ins e módulos para necessidades específicas. Portanto, eles tornam-se iguais em termos arquiteturais, ou seja, redes centradas no conteúdo, e serão referenciados de modo equivalente nesta tese.

Na arquitetura CCN, os nomes dos conteúdos não estão associados ao endereço físico do provedor. O conteúdo é consumido diretamente por meio de uma requisição de seu nome e os pacotes possuem campos que permitem a checagem de integridade e autenticação interna. Deste modo, as aplicações podem buscar e consumir conteúdos de forma mais simples e direta, apenas usando seus nomes, sem precisar de protocolos complexos de mapeamento de endereço e/ou roteamento. Essas características são atrativas para a Internet do futuro, pois reduzem o tráfego na rede e simplificam a entrega e obtenção de conteúdos.

A comunicação no CCN possui dois elementos: o consumidor e o provedor. O consumidor é a denominação do usuário que busca o conteúdo. Ao necessitar de uma informação, o consumidor envia uma mensagem do tipo

Interesse¹ (do inglês, *Interest Message*) com o nome do conteúdo desejado. Por outro lado, o provedor é o responsável por guardar o conteúdo desejado pelo consumidor. No CCN, qualquer nó da rede pode armazenar pedaços de conteúdo (chamado *chunks*). Essa característica é chamada *in-network caching*² ou armazenamento nos elementos internos da rede. Portanto, o provedor pode ser o nó de origem da informação ou qualquer nó intermediário que possua o conteúdo armazenado. O provedor só envia um pacote do tipo Dados³ (do inglês, *Data Packet*) como resposta ao pacote de Interesse. Desta forma, o consumidor consegue controlar o fluxo de informações que recebe. Cada mensagem CCN pode ser autenticada e possui informações de integridade, tornando a comunicação mais confiável e segura sem a necessidade de protocolos extras. Os nomes dos conteúdos são usados também para o roteamento do pacote de Interesse e de Dados. Ao ser requisitado, um pacote de Dados percorre o mesmo caminho do respectivo pacote de Interesse requisitante, porém no sentido inverso (Jacobson V. e., 2009).

Um dos grandes potenciais do CCN é seu suporte à mobilidade do consumidor. Se um consumidor deseja um conteúdo, ele envia um pacote de Interesse e quem possuir essa informação retorna o pacote de Dados correspondente. Em cenários móveis, onde um consumidor migra de uma rede de acesso sem fio para outra, eventualmente, não recebe pacotes de dados devido à desconexão. O CCN contorna esse problema apenas reenviando os pacotes de Interesse que não obtiveram os respectivos dados.

Por outro lado, quando o provedor é quem migra para outra rede, a recuperação do conteúdo, após a mobilidade, não é trivial. O pacote de Dados é encaminhado de volta ao consumidor usando o caminho contrário do seu respectivo pacote de Interesse. Como o pacote de Interesse não consegue chegar ao provedor, devido o processo de mobilidade, a comunicação é perdida. Portanto, o caminho até a nova localização do provedor precisa ser completamente atualizado, caso contrário, o encaminhamento do pacote de Interesse sempre vai levar à localização antiga do provedor. Apesar de o CCN acrescentar melhorias no gerenciamento de mobilidade, o problema da mobilidade de provedor ainda é relevante e carente de soluções eficientes (Kurose, 2014).

Com o advento de novas tecnologias sem fio, como Internet das Coisas (*IoT - Internet of Things*) e Redes Veiculares (VANET), os desafios da mobilidade de provedor em redes CCN ficam ainda mais evidentes. Como é ilustrado na **Error! Reference source not found.**, o provedor de conteúdo CCN pode ser qualquer dispositivo móvel fornecedor de Informações. A **Error! Reference source not found.** mostra três camadas de comunicação. Na camada

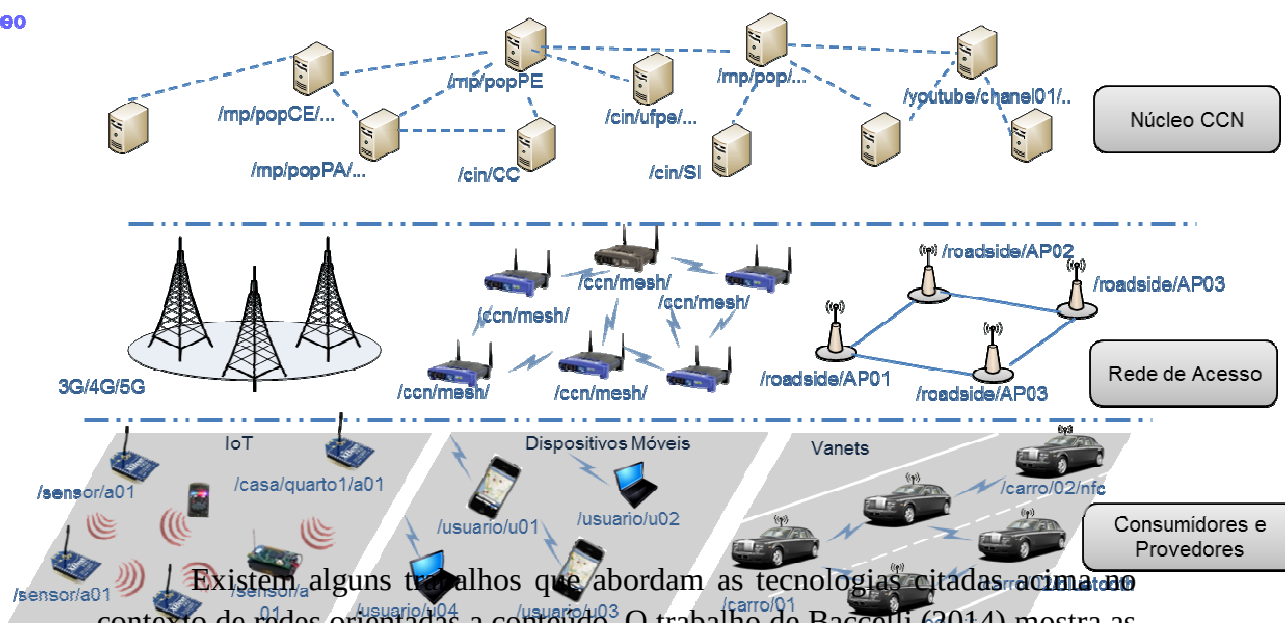
¹ Nesta tese, a palavra “Interesse” com i maiúsculo refere-se a mensagens de requisição de conteúdo do CCN.

² Nesta tese, o termo “caching” fará referência ao armazenamento nos elementos internos da rede CCN.

³ A mensagem “Dados” refere-se ao pacote de resposta ao pacote Interesse.

mais baixa, usuários consomem e produzem conteúdo. Um sensor, pertencente a uma rede de IoT, pode prover informações de temperatura, umidade, estado de objetos; um veículo, em VANETs, pode fornecer dados sobre a via em que está trafegando, acidentes, buracos; um *smartphone*, pode fornecer informações sobre sua localização, *streaming* de vídeo ou mensagens em redes sociais. São todos provedores que, se estiverem migrando entre redes, estão suscetíveis aos problemas inerentes à mobilidade de provedor.

Núcleo



Borda

Existem alguns trabalhos que abordam as tecnologias citadas acima no contexto de redes orientadas a conteúdo. O trabalho de Baccelli (2014) mostra as vantagens e desafios de implementar IoT em ICN. O foco de Rayes (2012) é no

Figura 1.1 Arquitetura de comunicação entre dispositivos CCN móveis.

trabalhos avançam a implementação de redes veiculares com ICN (Taleb et al., 2015) (Amadeo M. C., 2012). Em Amadeo (2016) é apresentado um *survey* sobre perspectivas futuras nessa área. Portanto, é notório que redes orientadas a conteúdo estão ganhando atenção de modo crescente em cenários de mobilidade. Com isso, a mobilidade de provedor em CCN torna-se um desafio justificável de investigação.

1.2. Problema

Os autores em Yao (2016) afirmam que as redes 5G⁴ foram concebidas para integrar todas as ideias relacionadas às redes do futuro, como SDN, ICN, IoT e Big Data (Chen M. S., 2014) (Sri, 2016). O uso do CCN em arquitetura da Internet do futuro tem o objetivo de aproveitar a importante característica do armazenamento nos nós internos da rede. Por outro lado, pesquisas mostram que soluções padrão de armazenamento *in-network* (*caching*), como o LCE (*Leave Copy Everywhere*), onde cópias dos conteúdos são armazenadas em todos os nós, são ineficazes e desperdiçam recursos da rede (Zhang G. L., 2013). Portanto, o uso efetivo de *caching* em ambientes com mobilidade precisa ser

⁴ Quinta geração das comunicações móveis.

investigado. Neste contexto, esse trabalho propõe soluções para três problemas principais:

A. *Caching* em redes CCN.

O armazenamento nos elementos da rede é uma das características mais importantes das redes CCN. O foco de estratégias de *caching* é decidir em que local armazenar os *chunks* na rede de modo mais eficiente possível. Apesar de existirem vários trabalhos, nenhum método proposto na literatura é implementado em redes de produção. Com isso, esse problema ainda é pertinente e é alvo ativo de pesquisas (Saxena, 2016).

B. Mobilidade de Provedor

Quando um consumidor migra para outra rede, a mobilidade é naturalmente suportada pelo CCN. Porém, quando o provedor migra de uma rede para outra, o caminho entre o consumidor e o provedor precisa ser atualizado e isso não é feito pelo protocolo CCN. Com isso, esse problema também foi analisado e uma solução foi elaborada.

C. *Caching* e Mobilidade

Tendo em vista que soluções fim-a-fim para a distribuição de conteúdo precisam ser consideradas, o terceiro problema de pesquisa abordado é a união entre os dois problemas anteriores em um único cenário. Não é do conhecimento do autor desta tese que haja trabalhos na literatura CCN sobre a integração entre *caching* e mobilidade. Geralmente, os trabalhos abordam *caching* como um problema de rede cabeada. Não há trabalhos consolidados que mostrem os dois cenários de modo sinérgicos, como é apresentada nesta tese.

Uma solução foi proposta para cada um dos problemas abordados. Elas serão submetidas à avaliação usando modelagem e simulação. Para a simulação, um simulador foi desenvolvido exclusivamente para a tese, a partir da extensão do OMNET (Varga, 2017).

Este trabalho toma por base a seguinte hipótese: estratégias de armazenamento de conteúdo podem melhorar o desempenho de serviços tanto em redes cabeadas como em redes sem fio, especialmente na presença de mobilidade de provedor.

A partir dessa hipótese, podemos destacar os seguintes questionamentos iniciais.

Q01 – Qual o benefício de esquemas de armazenamento de conteúdo?

Q02 – Como funciona o armazenamento em redes cabeadas?

Q03 – Qual o comportamento dos algoritmos de armazenamento em redes móveis?

Q04 – Qual o problema da mobilidade de provedor?

Q05 – Quais são as soluções da literatura para esse problema? Quais as mais eficientes?

Q06 – Técnicas de *caching* são influenciadas por ambientes com mobilidade de provedor? Uma técnica pode ser superior à outra, mesmo em ambientes com alta taxa de mobilidade?

Q07 – A mesma técnica de *caching* possui diferentes desempenhos quando submetidas a cenários com e sem mobilidade?

Q08 – Como as diversas propostas apresentadas podem ser avaliadas? Qual a melhor forma de avaliá-las?

1.3. Objetivos

A seguir, serão apresentados os objetivos gerais e específicos da pesquisa desenvolvida nesta tese.

1.3.1. Objetivo Geral

Desenvolver um arcabouço com algoritmos e mecanismos para melhorar a qualidade de serviços em redes móveis CCN, focando, principalmente, na mobilidade de provedor e usando armazenamento eficiente de conteúdos nos elementos da rede. Para isso, a proposta precisa fornecer técnicas para o armazenamento de conteúdos na parte cabeada da rede, bem como resolver o problema de mobilidade de provedor.

1.3.2. Objetivos Específicos

Os objetivos específicos desta tese serão listados abaixo.

- 1) Desenvolver e implementar um algoritmo para armazenamento de conteúdo nos elementos da rede e avaliar a proposta comparando-a com os algoritmos da literatura em ambientes cabeados e móveis (Q01, Q02, Q03).
- 2) Detalhar o problema de mobilidade de provedor. Desenvolver e implementar uma técnica que resolva esse problema (Q04, Q05).
- 3) Demonstrar que *caching* e mobilidade são problemas correlacionados (Q06) (Q07). Propor novas estratégias de *caching* que visam ambientes móveis e podem gerar ganhos na qualidade dos serviços nas redes CCN.
- 4) Avaliar as propostas apresentadas e compará-las com os da literatura (Q8).

1.4. Estrutura da Tese

Esta tese está organizada da seguinte forma: o Capítulo 2 introduz a tese com os conceitos necessários para o seu completo entendimento. O Capítulo 3 detalha trabalhos relacionados ao tema de pesquisa abordado, alguns dos trabalhos apresentados são usados em experimentos para comparar com soluções propostas. O Capítulo 4 apresenta o problema de armazenamento de conteúdos na rede e o funcionamento do RankCache, que é um algoritmo desenvolvido para resolver esse problema. O Capítulo 5 aborda os problemas da mobilidade de provedor nas redes NDNs e a solução desenvolvida. Esta tese apresenta soluções para três desafios das redes CCN. São eles, mobilidade, *caching* e perdas de pacotes durante o *handover*. Ao invés de apresentar uma solução para os três problemas ao mesmo tempo, eles foram divididos em subproblemas para atacar cada um individualmente (Capítulos 4 e 5). O Capítulo 6 mostra o estudo que apresenta a relação entre mobilidade e *caching*, demonstrando estatisticamente sua correlação. O Capítulo 7 descreve o MobCache, que é a proposta completa que une solução para mobilidade de provedor, armazenamento nos elementos da rede e perdas de pacotes. O Capítulo 8 finaliza esta tese e mostra possíveis caminhos para trabalhos futuros.

2

REFERENCIAL CONCEITUAL

Neste capítulo são abordados os conceitos fundamentais para o bom entendimento desta tese. Serão apresentados os conceitos chaves de redes CCN como: armazenamento nos nós internos da rede, disseminação de prefixos e mobilidade. O capítulo esclarece os benefícios das redes CCN em comparação com as redes tradicionais e mostra a limitação do CCN quando há mobilidade de provedor.

2.1. Introdução às Redes Orientadas a Conteúdo

Quando foi projetada, o objetivo da Internet era a comunicação eficiente entre estações. A busca constante por conteúdo e a popularização da Internet estimularam a criação de novas aplicações, como serviços de publicação de vídeo, redes sociais e redes par-a-par (*peer-to-peer* ou P2P) (Passarella, 2012). Essa revolução mudou bastante a forma como os usuários finais se comunicam e trocam dados entre si. Com isso, o objetivo atual da Internet é o consumo de conteúdo independente da localização física do mesmo. Esse novo objetivo implica em uma infraestrutura que possibilite a troca de conteúdos de forma segura, eficiente e com grande disponibilidade.

Soluções foram propostas para atender os requisitos apresentados acima. No entanto, essas propostas foram implementadas sobre a arquitetura limitada da Internet atual. Algumas dessas soluções obtiveram algum sucesso, como, por exemplo, as redes de distribuição de conteúdo (*CDN - Content Distribution Network*) (Passarella, 2012), como Akamai ou redes P2P como BitTorrent. Porém, a arquitetura atual da Internet ainda sofre com problemas de persistência, segurança dos conteúdos e disponibilidade de serviços. É comum ocorrerem ataques e invasões que prejudicam as redes CDN e/ou P2P. Outro problema é o reposicionamento dinâmico de conteúdos que, em CDN, precisa ser feito por meio de consultas a serviços centralizados, gerando grande lentidão na entrega dos dados. Com os protocolos da Internet cheios de remendos e engessados para

novas ideias, muitos autores defendem a remodelagem da arquitetura da Internet atual, as chamadas propostas *clean-slate* (Rexford, 2010).

O paradigma CCN faz parte de uma das propostas *clean-slate*. No CCN, cada conteúdo possui um nome único, persistente e hierárquico. A segurança é implementada no nível do pacote, ou seja, existem campos dentro das mensagens que são usados para autenticação e integridade. Como ilustrado na Figura 2.1, a arquitetura CCN herda o modelo ampuheta da pilha de protocolos TCP/IP, em que uma ampla variedade de aplicações e protocolos é suportada. No entanto, a parte mais estreita do modelo utiliza nomes de conteúdo ao invés de endereços IPs.

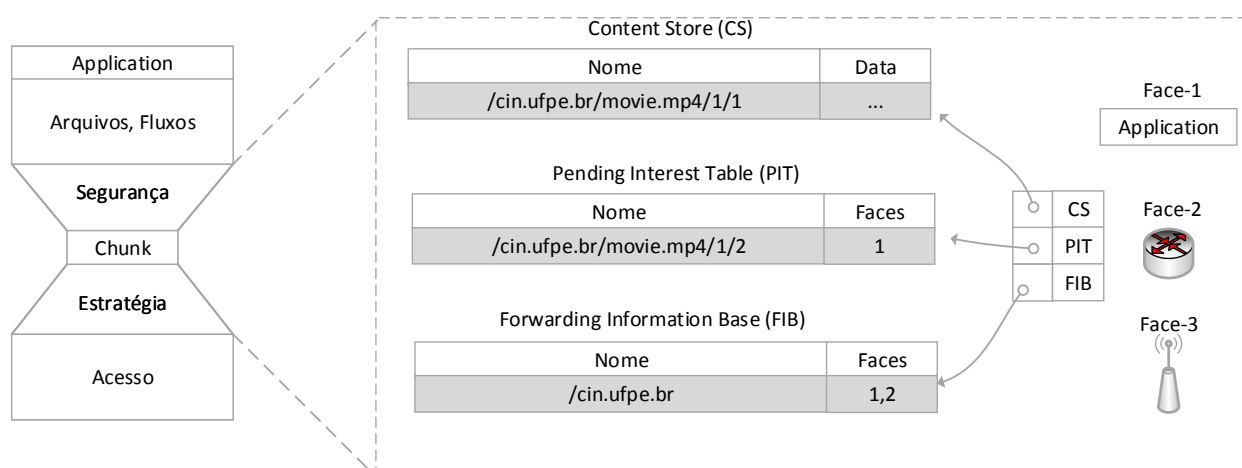


Figura 2.1 Modelo Ampulheta do CCN.

As interfaces, ou apenas “*faces*”, são usadas para representar abstrações de canais de fluxos de pacotes. As *faces* podem ser aplicações CCN ou até mesmo tecnologias físicas como canais de comunicação cabeados e sem fio. Esse modo de classificar os canais de comunicação flexibiliza a entrega de conteúdos. Como é mostrado na Figura 2.1, o pacote “`/cin.ufpe.br`” pode ser transmitido por duas *faces* diferentes (face 1 e 2) ao mesmo tempo, sendo a *face* 1 uma aplicação no espaço do usuário e a *face* 2 uma saída para a rede cabeada. Essa característica de envio *multicaminho* é um dos benefícios da flexibilização do uso das *faces*.

Cada nó CCN possui três estruturas principais: o CS (*Content Store*), a PIT (*Pending Interest Table*) e a FIB (*Forwarding Information Base*). No CS ficam armazenados os pedaços (*chunks*) de conteúdo. Toda vez que um pacote de dados passa por um CS, um algoritmo avalia se o dado será armazenado para posterior requisição. Se sim e o CS já estiver cheio, são usados algoritmos de substituição como LFU (*Least Frequently used*) ou LRU (*Least Recently Used*) (Podlipnig, 2003). Esse conceito torna o CCN uma rede de *caches*, porém, diferente das CDNs, que implementam a ideia de redes de *cache web*, o *caching* em CCN é limitado em termos de capacidade de armazenamento. Além disso, o *caching* é aplicado aos conteúdos de todos os usuários da rede e não apenas ao conteúdo de provedores que tenham contrato com o serviço, como é feito no

CDN (Ghodsí, 2011). Isso torna o gerenciamento do *caching* no CCN bem mais desafiador (Zhang G. L., 2013).

A PIT armazena informações de pacotes de Interesse já recebidos e encaminhados pelo nó, ou seja, executa um serviço de monitoramento de pacotes de saída. Cada entrada da tabela possui um tempo de vida e quando esse tempo expira a entrada é removida. Quando um pacote de dados chega ao nó, ele é encaminhado até o nó requisitante através da PIT. Por outro lado, a FIB armazena informações para o encaminhamento correto dos pacotes, realizando o mapeamento do nome dos conteúdos e uma ou mais interfaces de saída. A estrutura hierárquica dos nomes no CCN ajuda a manter a FIB simples e pequena. A decisão de encaminhamento baseada na PIT e na FIB mantém o roteamento simples e eficiente.

2.2. Armazenamento Interno de Conteúdo nos elementos da rede CCN

Uma das características mais importantes do CCN é o armazenamento de *chunks* nos nós da rede. Enquanto nas redes tradicionais o conteúdo precisa ser obtido nos níveis de servidores acima dos clientes, no CCN o conteúdo pode ser fornecido por qualquer nó da rede. A Figura 2.2 mostra como funciona o serviço de vídeo sob demanda (*YouTube*®) na rede tradicional e como funciona nas redes CCN, ambos passando por vários provedores de conteúdo (ISP - Internet Service Provider). Enquanto na Internet tradicional a comunicação é fim-a-fim, nas redes CCN o conteúdo pode ser entregue de cliente para cliente, reduzindo o caminho percorrido pelo conteúdo, diminuindo o atraso de entrega de pacotes e o tráfego total na rede.

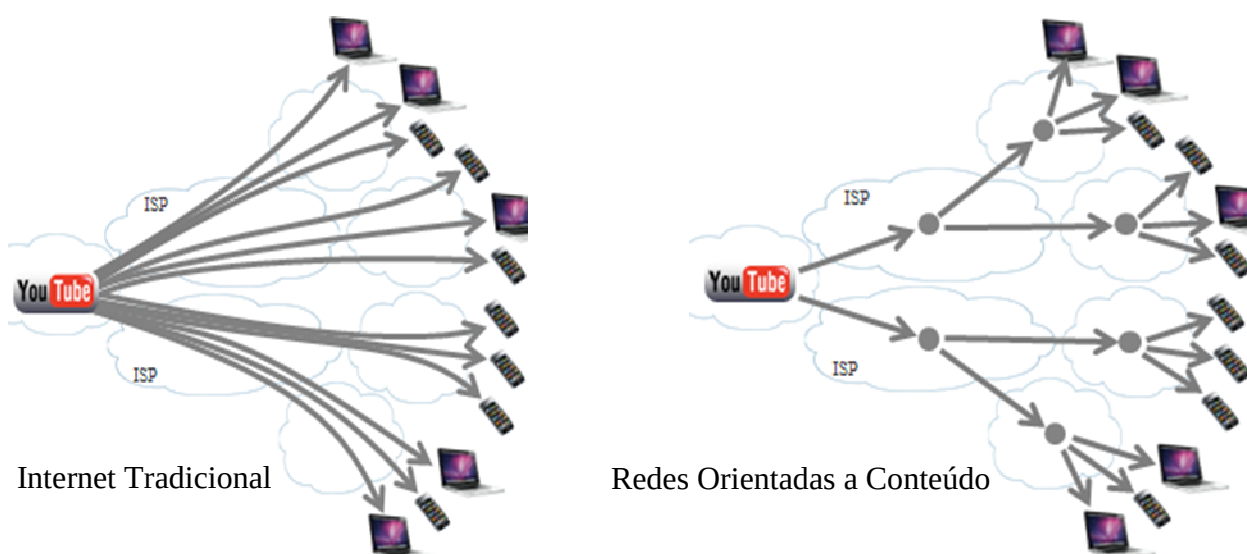


Figura 2.2 Comparação redes convencionais e CCN

O CCN permite armazenamento dentro dos elementos da rede (denominado em inglês de *in-network* ou *in-path caching*). Essa característica melhora significativamente a entrega de conteúdos populares (constantemente requisitados), pois os usuários podem receber o conteúdo diretamente de elementos do núcleo da rede, sem que a requisição precise chegar ao provedor. Se um *chunk* do conteúdo requisitado for encontrado no *cache* de um nó da rede, que não seja o provedor, diz-se que ocorreu um “acerto” (do inglês, *hit*). Se um *chunk* não for encontrado em um *cache*, diz-se que ocorreu um “erro” (do inglês, *miss*). Nesse último caso, o pacote Interesse é encaminhado ao próximo *cache* no caminho até o provedor.

As estratégias de armazenamento na rede podem ser divididas em duas: políticas de substituição (do inglês, *replacement*) e de decisão (do inglês, *decision*). A primeira define qual *chunk* armazenado será removido quando o *cache* estiver cheio. A segunda está relacionada à decisão de guardar ou não um *chunk* que atravessa o nó. Existem várias políticas de substituição de *chunks*, porém, a mais utilizada em trabalho de *cache* é a LRU (*Least Recently Used*) (Chai W. K., 2013) (Psaras, 2011), a qual remove o item menos usado do *cache*. Essa política também foi recomendada pelos autores do CCN (V. Jacobson, 2009).

Neste trabalho, o foco será em políticas de decisão. O desafio é decidir em qual *cache* um *chunk* deve ser armazenado, de modo a maximizar a probabilidade de ocorrer um acerto, diminuir a redundância de *chunks* nos *caches* e reduzir o atraso na entrega dos conteúdos. A Seção 3.1 explicará mais sobre essas políticas, detalhando algumas, da literatura, que servirão como base para comparação com o algoritmo de *caching* proposto na tese.

2.3. Disseminação de prefixos

O CCN precisa de um mecanismo para disseminação de nomes (ou prefixos) de conteúdo, e desta forma, preencher as FIBs com informações para o encaminhamento correto dos conteúdos. Nas redes TCP/IP, existem vários protocolos de roteamento e disseminação de prefixos IP como RIP, OSPF, EIGRP (Vetriselvan, 2014). Visando a implementação de protocolos de disseminação de prefixos no CCN, o (Wang L. H., 2012) usou o protocolo *Open Shortest Path First* (OSPF) como ponto de partida, pois ele é amplamente empregado na Internet e possui implementação *open-source*. Em Wang (2012), os autores apresentam o OSPFN (*OSPF for Named-data*) que usa mensagens OLSA (*Opaque Link State Advertisements*) (L. Berger, 2008) para anunciar os nomes de prefixos na rede. O OSPFN foi implementado no simulador que foi desenvolvido por esta tese.

O roteamento em redes CCN é diferente do tradicional IP em dois sentidos: (a) roteamento é feito com prefixos de nomes ao invés de prefixos IP; (b) o roteamento é *multipath*, ou seja, um nome pode ser alcançado através de

múltiplos caminhos. O OSPFN cria os prefixos de nomes e os instala no CCND (*Content Centric Network Daemon*) (CCNx.org, 2016), o qual gerencia o encaminhamento de mensagens *Interest* e *Data*. Na implementação do CCNx, cada roteador CCN executa uma instância do CCND, do OSPFN e OSPFD (*OSPF daemon*). A Figura 2.3 ilustra a troca de mensagens entre os componentes. Quando o nó CCN é inicializado, ele lê o arquivo de configuração e cria mensagens OLSA para cada nome de prefixo que queira anunciar na rede.

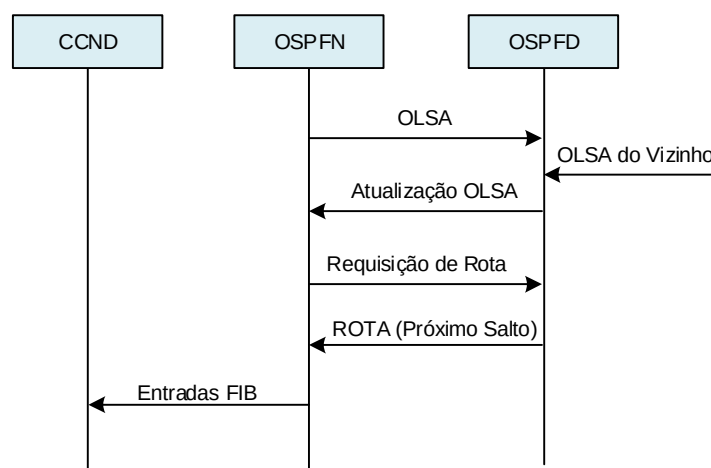


Figura 2.3 Sinalização OSPFN

As mensagens OLSA são entregues ao OSPFD para serem encaminhadas para todos os nós da rede. O OSPFD pode avisar o OSPFN de qualquer tipo de atualização vinda de outras mensagens OLSA enviadas por vizinhos. Quando o OSPFD recebe uma mensagem de consulta do OSPF, ele procura em sua tabela de roteamento a lista de próximos saltos e os custos do caminho associado. O protocolo inclui essas informações em uma única mensagem e, em seguida, envia a resposta ao OSPFN. Com as informações obtidas do OSPFD, o OSPFN atualiza a FIB do CCN. Nesse momento, a tabela de encaminhamento é montada e o prefixo pode ser roteado.

As informações dos prefixos são armazenadas na *Name Prefix Table* ou tabela de prefixos de nomes, ilustrada na Tabela 2.1. Essa tabela armazena os prefixos e o roteador de origem. Toda vez que um pacote OLSA é recebido e validado, uma nova entrada é criada na tabela de prefixos. Essa tabela fica armazenada no OSPFN. Quando há atualização de rotas, o OSPFD pode enviar uma requisição ao OSPFN para remover e/ou adicionar alguma entrada da tabela.

Tabela 2.1 Prefixos de Nomes

CCN-ROUTER-03 Prefixos de Nomes	
<i>Name Prefix</i>	<i>Advertising Router</i>
/cin/ufpe/computacao	150.161.8.1 (CCN-ROTEADOR-01)
/cin/ufpe/	150.161.9.1 (CCN-ROTEADOR-02)

/cin/ufpe/br/	150.1.8.1 (CCN- ROTEADOR-05)
/cnpq/br/lattes	150.161.10.1 (CCN-ROTEADOR-10)

Esta tese implementou o OSPFN no simulador desenvolvido. Para tal, tomou-se como base o código fonte disponibilizado no link <https://github.com/NDN-Routing/OSPFN2.0>, na RFC do OSPF (RFC 2328) e no trabalho de Wang (2012). Esse protocolo é necessário principalmente para experimentos com redes móveis, pois, ao disseminar os prefixos, o OSPFN garante que haverá rota para os pacotes na rede de destino. A Seção 2.4 irá detalhar o comportamento do CCN em ambientes móveis.

2.4. Mobilidade

Esta seção apresentará

2.4.1. Mobilidade TCP/IP

Com a evolução tecnológica, os dispositivos móveis começaram a diminuir de tamanho e a necessidade de comunicação do usuário mudou de somente através de voz, em aparelho fixo, para dados, com aparelhos móveis em qualquer lugar e a qualquer hora. Mais próximos aos usuários, esses dispositivos se popularizaram, gerando crescimento de conexões simultâneas em uma mesma rede. Com isso, a migração (processo chamado de *Handover* ou *Handoff*) de um dispositivo móvel entre PoAs⁵ (*Point of Attachments*) distintos começou a se tornar um problema.

Em relação ao domínio administrativo, existem dois tipos de mobilidade: intra-domínio e inter-domínio. O primeiro refere-se à mobilidade entre redes pertencentes ao mesmo domínio administrativo, e no segundo a mobilidade ocorre entre domínios diferentes (Lee K. W., 2010). O segundo é mais complexo, pois é necessário algum acordo de nível de serviço, SLA (*Service Level Agreement*), negociado previamente entre os domínios (Ro, 2015).

Este cenário de mobilidade, com diversos dispositivos móveis e mobilidade intra e inter-domínios, não foi considerado na arquitetura da Internet na concepção do protocolo IP. Para tentar resolver os problemas causados pela evolução das redes, muitos protocolos foram propostos como “remendos” à pilha TCP/IP, como o Mobile IP (Perkins, 1997), HAWAII (Ramjee, 1999), Proxy MIP (Gundavelli, 2008) e outros (Akyildiz, 2004) (Saha, 2004) (Gladisch, 2014). Atualmente, pode-se dizer que nenhum deles foi largamente aceito como solução definitiva para o problema, pois a maioria desses protocolos sofre de problemas como roteamento triangular, sobrecarga de sinalização e complexidade. A falha desses protocolos é atacar os sintomas e não o problema principal, ou seja, a forte ligação entre a identificação e a localização do

⁵ Pontos de Acesso nas redes 802.11, Estação Base nas redes 802.16 ou eNodeB nas redes LTE.

conteúdo (Jokela, 2004). Alguns trabalhos tentaram resolver esse problema, como P. A. Nikander (2010), propondo maneiras de separar a localização da identificação. No entanto, a solução é mais um “remendo” em uma arquitetura que não foi planejada para suportar uma grande quantidade de informações, nem a mobilidade de números cada vez maiores de dispositivos móveis.

2.4.2. Mobilidade em Redes CCN

O CCN suporta nativamente a separação entre localização e identificação do conteúdo. A comunicação é controlada pelo consumidor e o nome do conteúdo pode ser hierarquicamente organizado para facilitar o roteamento e simplificar as tabelas de encaminhamento. Essas características satisfazem os requisitos das comunicações sem fio do futuro, onde a preocupação é o conteúdo e não sua localização física. Deste modo, todas as entidades da rede (objetos de informação, *hosts* e outras entidades) são unicamente identificadas por meio de seu nome e encontradas dinamicamente através do roteamento hierárquico desses nomes. Com isso, a mobilidade do consumidor é naturalmente suportada pela arquitetura.

A única diferença entre **consumidor** e **provedor** nas redes CCN é que o primeiro é o elemento que deve enviar o pacote de Interesse (ou seja, é o nó que está interessado no conteúdo); o segundo é o elemento que envia o pacote de Dados (ou seja, é o nó que fornece o conteúdo). Tanto o provedor quanto o consumidor pode ser um dispositivo móvel. Quando um usuário consumidor move-se para outra rede, o dispositivo precisa apenas retransmitir qualquer pacote de Interesse que não tenha sido satisfeito. Por outro lado, a mobilidade de provedor é um pouco mais complexa.

2.4.3. Mobilidade do Consumidor

As Figura 2.4 e Figura 2.5 ilustram a mobilidade de consumidor em redes CCN. A Figura 2.4 mostra que, inicialmente, o consumidor, conectado ao PoA1, busca por um conteúdo de vídeo enviando um pacote de Interesse com o nome `"/cin/ufpe/movie.mp4/1/1"`. Este pacote do tipo Interesse percorre o *caminho-1*, composto pelos nós PoA1, Roteador01, Roteador04 e PoA3. Este caminho é formado através das regras de encaminhamento instaladas dentro das FIBs de cada nó. A FIB dos nós envolvidos no *caminho-1* está ilustrada no lado esquerdo na Figura 2.4. O caminho do pacote de Dados percorre o caminho inverso ao *caminho-1*, usando as informações da PIT.

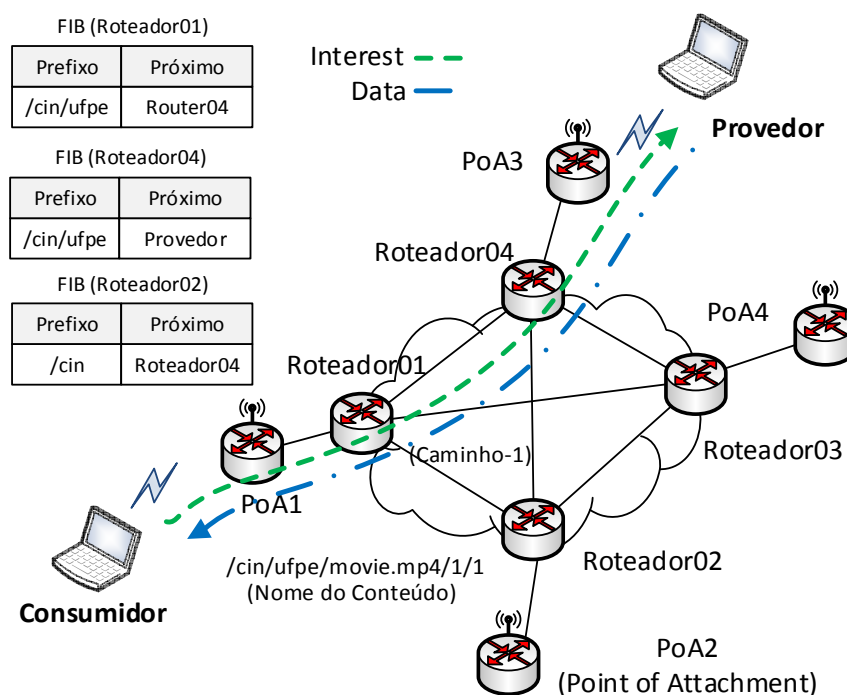


Figura 2.4 Comunicação Consumidor-Provedor antes da mobilidade do consumidor

Quando o consumidor migra do PoA1 para o PoA2 (Figura 2.5), ele precisa apenas retransmitir qualquer pacote de Interesse enviado anteriormente que não tenha sido satisfeito. Esse Interesse percorrerá o *caminho-2*, formado pelos nós PoA2, Roteador02, Roteador04 e PoA3. Assim como no *caminho-1*, o encaminhamento é feito pelas regras presentes nas FIBs de cada nó. Desta forma, o consumidor obtém seus dados da mesma forma que estava obtendo anteriormente no PoA2, ou seja, o conteúdo vem pelo caminho inverso ao *caminho-2*. Percebe-se que nada foi mudado nas tabelas ou em outras entidades para que a mobilidade do consumidor fosse concluída com sucesso. A mobilidade é automática porque o Roteador02 já possui a entrada “/cin” que satisfaz a requisição do conteúdo “/cin/ufpe/movie.mp4/1/1”. Como o CCN faz encaminhamento através de agregação de prefixo, apenas “/cin” é o suficiente para saber para onde encaminhar a requisição. Vale ressaltar, que a entrada “/cin” foi instalada previamente no Roteador02, através de algoritmos de disseminação de prefixos como o OSPFN, ver Seção 2.3.

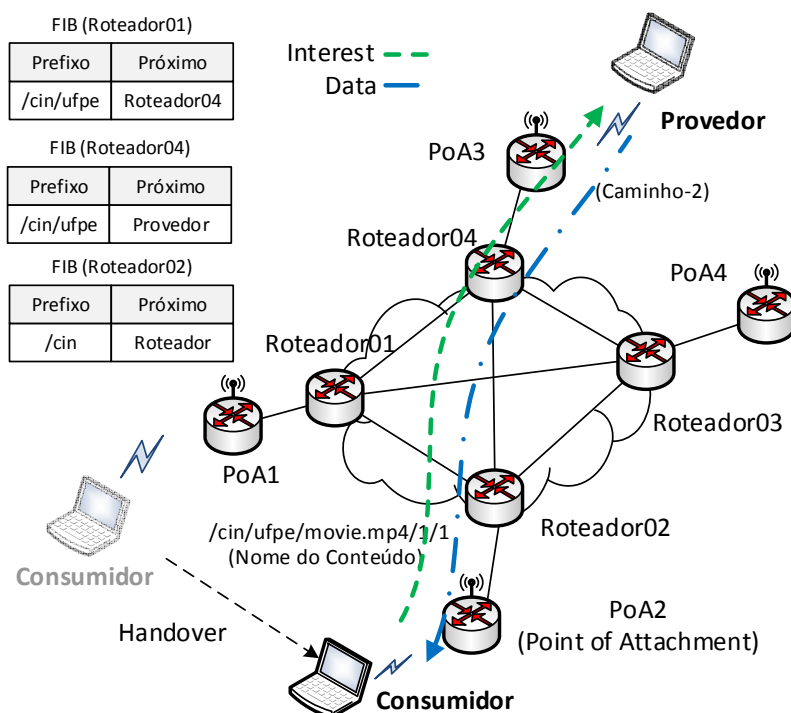


Figura 2.5 Comunicação Consumidor-Provedor após mobilidade do Consumidor

2.4.3.1. Mobilidade do Provedor

O CCN suporta a mobilidade de consumidor nativamente. No entanto, a mobilidade de provedor é mais complexa. A Figura 2.6 mostra a comunicação entre o consumidor e o provedor de conteúdos antes da mobilidade. A comunicação segue o comportamento mostrado na Figura 2.6, com o consumidor fazendo requisições no *caminho-1* (PoA1, Roteador01, Roteador04 e PoA3 antes do provedor).

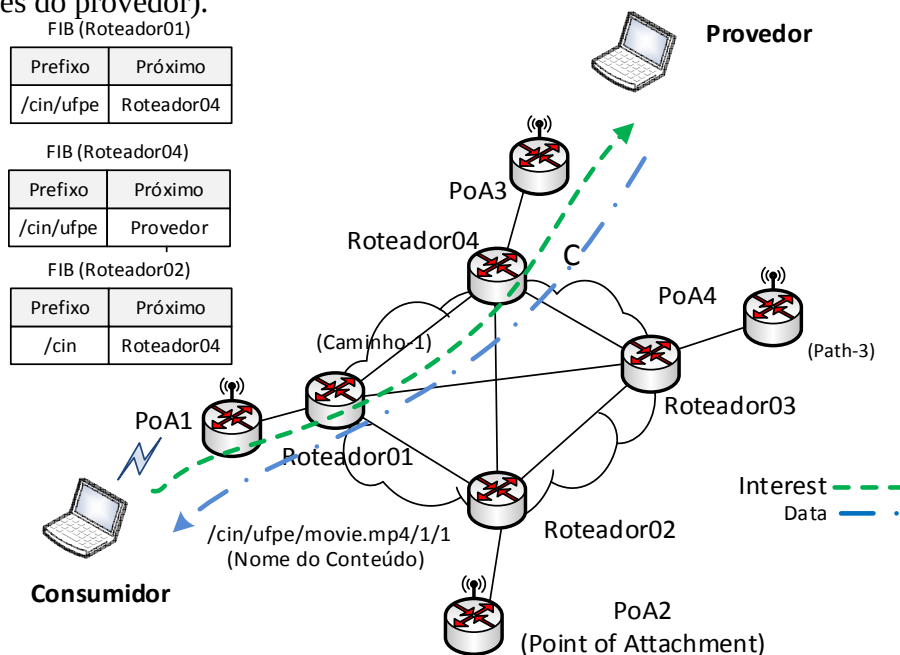


Figura 2.6 Comunicação Consumidor/Provedor antes da mobilidade do Provedor.

Neste cenário, quem se move é o provedor de conteúdo. E, como mostra a Figura 2.7, o encaminhamento do pacote de Interesse falha ao tentar encontrar o provedor após a mobilidade. Isso ocorre porque, mesmo após a mobilidade do provedor, o pacote “/cin/ufpe/movie.mp4/1/1” continua sendo encaminhado pelo *caminho-1*. A solução para o problema é atualizar as tabelas de encaminhamento dos nós da rede. Ou seja, a FIB do Roteador01, ao invés de apontar para o Roteador04, como está na Figura 2.7, apontaria para o Roteador03 formando o *caminho-3* (PoA1, Roteador01, Roteador03 e PoA4). Após essa atualização, a comunicação entre consumidor e provedor seria reestabelecida e os conteúdos seriam recuperados com sucesso.

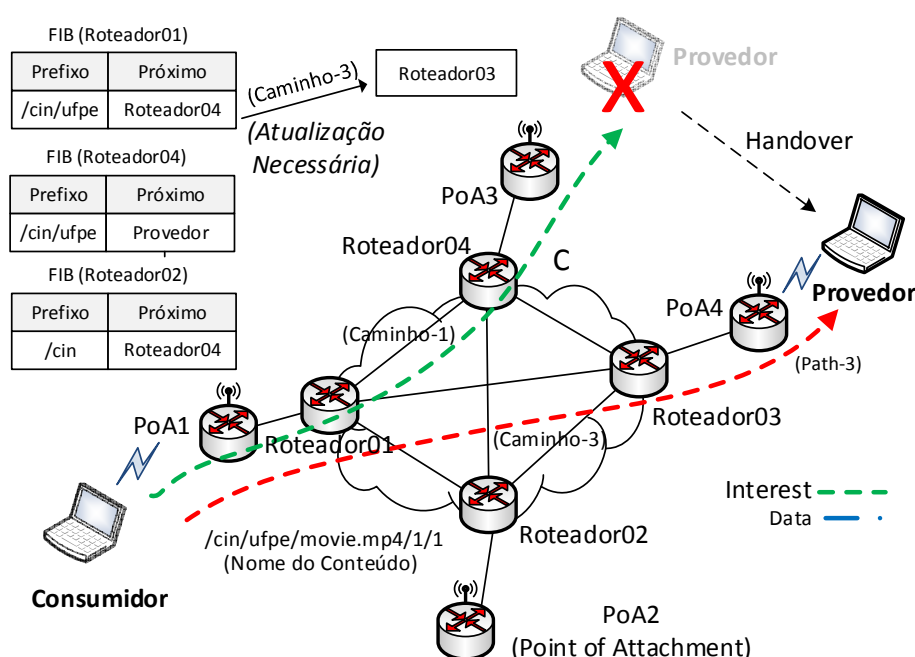


Figura 2.7 Falha ao encontrar o conteúdo após a mobilidade do Provedor

A atualização da FIB, necessária para restabelecimento da comunicação após a mobilidade do provedor, não é naturalmente suportada pelo protocolo CCN. Portanto, a mobilidade de provedor ainda é um ponto em aberto nessa arquitetura. Na Seção 3.2, serão analisados diversos trabalhos que abordam esse problema. Um dos parâmetros que se usa para comparar as soluções de mobilidade de provedor é o tempo de duração entre a saída do usuário da rede de origem e o recebimento do primeiro conteúdo na rede de destino. Esse tempo é conhecido como latência de *handover*.

2.5. Considerações Finais

Este capítulo abordou os conceitos fundamentais para o entendimento desse trabalho. Foram detalhadas as redes CCN, armazenamento nos nós internos da rede, disseminação de prefixos com o OSPFN e mobilidade de provedor. O próximo capítulo irá investigar os trabalhos relacionados ao tema de pesquisa desta tese. Além de mostrar os algoritmos de decisão de *caching* usados como comparação com uma das propostas da tese, o capítulo apresenta um

trabalho detalhado sobre a mobilidade de provedor, incluindo classificação e comparação entre diversas propostas da literatura.

3

ESTADO DA ARTE

Este capítulo detalha os principais trabalhos relacionados ao tema de pesquisa da tese. A Seção 3.1 descreve os algoritmos mais importantes na área de políticas para a decisão de cache em ICN e que foram utilizados na comparação com a proposta concebida na tese para o problema de armazenamento eficiente. A Seção 3.2 apresenta uma revisão detalhada da literatura sobre soluções para a mobilidade de provedor em redes CCN, incluindo classificação e comparação entre essas diversas propostas.

3.1. Armazenamento no CCN

Existem alguns trabalhos na literatura que propõem políticas de decisão de *cache*. O objetivo de tais políticas é decidir onde armazenar os *chunks* de um determinado conteúdo. Normalmente, são políticas que decidem se um *chunk* que está atravessando um nó CCN será ou não armazenado nele. Essas políticas são distinguidas pelas informações utilizadas na tomada de decisão de armazenamento do *chunk* e como essas informações são obtidas. A seguir, serão detalhadas as propostas mais citadas e que farão parte da avaliação comparativa do Capítulo 4.

a) LCE (*Leave Copy Everywhere*)

Esta é a proposta mais usada em *caches* multiníveis. Após detectar o *chunk* requisitado, uma cópia é guardada em todos os *caches* intermediários no caminho de volta para o usuário requisitante. Apesar de simples, esse esquema apresenta grandes desvantagens, como o alto nível de redundância de *chunks* na rede. Porém, é bastante usada como base para comparação entre propostas em redes CCN (Laoutaris, 2004) (Chai W. K., 2013). Por esse motivo, também será incluída como uma das propostas de comparação no Capítulo 4.

b) FixProb (ou apenas Fix)

É uma versão randômica do LCE. Cada nó intermediário ao longo do caminho entre o consumidor e o provedor é um candidato possível para guardar uma cópia do objeto requisitado. Os *caches* intermediários podem guardar os objetos com uma probabilidade p . Nota-se que o FixProb com p igual a 1 (ou 100%) é idêntico ao LCE. Para fins de comparação usou-se p igual a 0.3 e 0.7, mesmos valores usados em alguns trabalhos (Psaras b, 2012).

c) LCD (*Leave Copy Down*)

É um algoritmo, proposto por Laoutaris (2004), onde uma cópia do conteúdo requisitado é armazenada no próximo *cache* após um acerto. Sempre que um acerto no *cache* acontece, o *chunk* é armazenado em apenas um *cache* após o acerto. O algoritmo verifica o número de saltos H que o *chunk* percorreu; o *cache* armazena o conteúdo somente se o valor de H for “1”, ou seja, apenas após o primeiro salto é que o *chunk* é armazenado. O LCD é mais “conservador” que o LCE, pois precisa de várias requisições a fim de levar o conteúdo para mais perto do consumidor. Por outro lado, esse algoritmo reduz o número de cópias redundantes de *chunks* na rede.

d) BTW (*Betweenness Centrality*)

Uma importante classe de políticas de decisão são as baseadas no conceito de centralidade dos nós (Izquierdo, 2006). O princípio fundamental dessa estratégia é: se um nó está posicionado ao longo de um grande número de caminhos de entrega de conteúdo, é mais provável que esse nó consiga mais acertos que outros nós. Existem várias estratégias de algoritmos de *caching* baseados em centralidade. Nesse trabalho, utilizou-se o algoritmo *Betweenness Centrality* (Chai W. K., 2011) por ser o que apresenta melhor desempenho e maior facilidade de implementação.

e) ProbCache

O ProbCache, proposto em Psaras (2012), é um algoritmo que explora o conceito de capacidade de armazenamento do caminho. Este valor é calculado por meio da medição da capacidade de armazenamento médio dos *caches* no trajeto do pacote. A proposta introduz dois novos cabeçalhos nas mensagens CCN, o TSI (*Time Since Inception*) e o TSB (*Time Since Birth*). Em cada salto, o valor TSI é incrementado unitariamente. O provedor do conteúdo atribui o valor TSI que vê na mensagem de requisição para a mensagem de resposta. Cada roteador incrementa o valor TSB da mensagem de resposta também em uma unidade. Assim, quando o conteúdo viaja de volta para o cliente, o TSI tem um valor fixo e indica o comprimento do caminho desse fluxo específico, enquanto o valor de TSB indica o número de saltos que o conteúdo da mensagem de resposta percorreu desde o acerto. O ProbCache faz um cálculo baseado nos

valores de TSI e TSB, que resulta na probabilidade de armazenar ou não um determinado conteúdo no *cache*.

Através de análise da literatura sobre algoritmos de armazenamento, chegou-se à conclusão de que as melhores propostas de políticas de decisão de *caching* levam em conta a popularidade dos *chunks*. Um acerto no *cache* envolve nada mais do que achar conteúdos requisitados anteriormente por outros usuários e que estejam armazenados no respectivo *cache*. A probabilidade de um acerto é diretamente proporcional à popularidade de um *chunk*. Uma política de decisão de *cache* eficiente deve ser capaz de prever quando essas requisições repetidas ocorrerão. Com isso, o presente trabalho propõe uma nova estratégia de armazenamento chamado RankCache, que explora os conceitos de popularidade de *chunks* e calcula o *rank* dos *caches*, a fim de maximizar a probabilidade de acertos.

3.2. Mobilidade

Alcançar a mobilidade de provedor já é um grande desafio. Se forem levadas em consideração aplicações de tempo real, o problema é ainda maior (Ravindran, 2012). Isso porque o tempo que o usuário fica sem o serviço precisa ser o menor possível, a ponto de o usuário nem perceber a degradação na qualidade do serviço, devido aos atrasos de pacotes ou quebra da conexão. A latência de *handover* é o tempo de duração entre a saída do usuário da rede de origem e o recebimento do primeiro conteúdo na rede de destino. Essa latência pode ser expressa pela Eq. 1:

$$\text{Handover Latency} = t_{\text{TIMEOUT}} + t_{\text{PoA_DETECT}} + t_{L2} + t_{\text{FIB_UPDATE}} \quad \text{Eq. 1}$$

A Eq. 1 é formada pela somatória dos tempos necessários para reestabelecer a conexão do usuário após a ocorrência de *handover*. Essa equação é uma adequação da latência de *handover* das redes móveis IP (Labiod, 2010) para a arquitetura CCN. As primeiras três parcelas são inerentes à camada física e de enlace da pilha TCP/IP. A última parcela, $t_{\text{FIB_UPDATE}}$, está associada ao CCN. Abaixo segue a explicação de cada parcela.

- t_{TIMEOUT} : Tempo necessário para que o dispositivo perceba que não está mais conectado à rede e, como consequência, precisa fazer o *handover* para outra rede. Esse parâmetro pode ser atenuado com técnicas de predição de *handover*.
- $t_{\text{PoA_DETECT}}$: Tempo necessário para o dispositivo detectar um novo PoA. Esse tempo depende da frequência de escaneamento utilizada. Quanto maior for a frequência de escaneamento, menor será o tempo para detectar um novo PoA, no entanto, maior também serão a sobrecarga e o consumo de energia no dispositivo.
- t_{L2} : Refere-se ao tempo de reconexão da camada L2. Este tempo pode ser reduzido se houver mais de uma interface ativa no

dispositivo. Enquanto uma interface estivesse se comunicando com a rede de origem, a outra faria o *handover* para a rede de destino, criando uma intercessão entre as duas redes, o que facilitaria a troca de rede.

- t_{FIB_UPDATE} : Tempo gasto para modificar todas as FIBs necessárias para atualizar o caminho entre o consumidor e o provedor.

O parâmetro t_{FIB_UPDATE} é o mais difícil de ser otimizado, pois, diferente dos outros, ele depende de estratégias e protocolos que envolvem vários elementos da rede. No caso do *handover* de consumidor esse parâmetro é zero, pois nesse tipo de *handover* a FIB não precisa ser atualizada. Porém, o *handover* de provedor depende de uma inteligência que não está implementada nativamente no CCN. Desse modo, diminuir o impacto desse parâmetro é o principal objetivo dos trabalhos apresentados na próxima seção.

3.2.1. Soluções para a mobilidade de provedor em CCN

Uma das contribuições desta tese foi criar uma classificação para as soluções, presentes na literatura, dadas ao problema de mobilidade de provedor. Essas soluções foram classificadas em três grupos: (a) Atualização Direta da FIB (AD-FIB), (b) Inspiradas no protocolo MIP (I-MIP) e (c) baseadas em entidades centralizadas (E-C). Para agrupá-las foram usados os critérios de similaridade nos mecanismos de trocas de mensagem e atualização das FIBs. Nas propostas AD-FIB, uma mensagem é enviada para o núcleo da rede com o objetivo de atualizar automaticamente as FIBs. Nas soluções I-MIP, é possível identificar elementos característicos do protocolo MIP, como Agente Doméstico, Agente Estrangeiro, Túneis. As propostas do grupo E-C são compostas por soluções que usam entidades centralizadas para tomar as decisões.

A Figura 3.1 apresenta as propostas analisadas e seus respectivos grupos. Nas seções seguintes, cada grupo será explicado em detalhes. No diagrama de Venn da Figura 3.1, algumas soluções estão presentes em mais de uma classificação. Isso acontece porque os trabalhos propõem mais de uma solução para o problema ou a solução proposta possui elementos que estão em mais de uma classificação. Como exemplo, o trabalho (Kim H., 2012) propõe três soluções distintas, cada uma se enquadra em um dos grupos, por esse motivo, o artigo está na interseção dos três grupos na Figura 3.1.

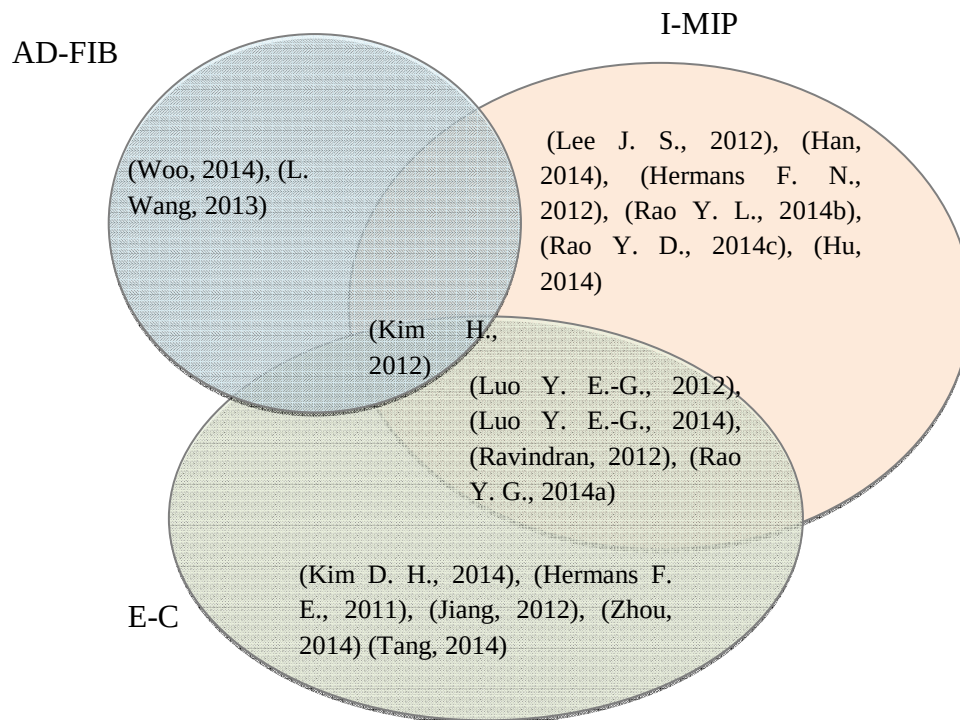


Figura 3.1 Diagrama das propostas de mobilidade de provedor e seus grupos.

3.2.1.1. Atualização Direta da FIB (AD-FIB)

Nesse grupo estão os trabalhos que propõem atualização da FIB usando mensagens diretas enviadas pelo provedor após sua mobilidade, geralmente uma mensagem é enviada ao núcleo da rede a fim de atualizar as FIBs dos nós do caminho. A mostra a sinalização base de todas as propostas das soluções desse grupo. Primeiramente, o consumidor MN2 está se comunicando com o provedor MN1 através do *caminho-1* (Eventos 1 e 2 da Figura 3.2), composto pelos elementos PoA-3, CN03, CN01 e PoA-2. Ao perceber a iminência da mobilidade, o provedor envia uma notificação ao PoA-2 (Evento 3). O Evento 4 marca o momento do *handover*. Após completar o processo de migração, o provedor informa ao PoA (PoA-1) do destino sobre sua entrada na rede (Evento 5). Esse aviso é feito por meio de uma mensagem (Evento 6) que também é usada para atualizar as tabelas de encaminhamento. Geralmente essa mensagem é criada apenas para esse fim e é direcionada ao PoA anterior, nesse caso, o PoA-2. Após o Evento 6, o CN03, que estava encaminhando mensagens de Interesse ao nó CN01, será atualizado para encaminhar para o CN02. A seguir, as propostas que implementam essa ideia serão explicadas com mais detalhes.

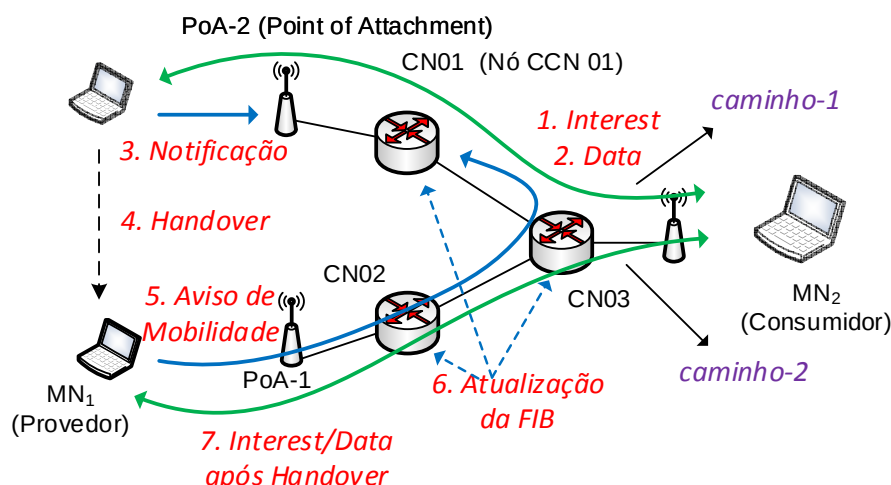


Figura 3.2 Sinalização genérica para o grupo AD-FIB.

Os autores do trabalho apresentado em Kim H. (2012) propõem o *Interest Forwarding*. Esse esquema usa uma mensagem de Interesse virtual para informar o PoA doméstico (ao qual o provedor está conectado) que o provedor está prestes a realizar o *handover*; após essa informação, o nó CCN doméstico começa a armazenar mensagens *Interest* que chegam do consumidor. Quando o provedor finaliza o processo de mobilidade e chega no PoA de destino, ele envia uma mensagem de Interesse especial para o PoA antigo. Essa mensagem tem a função de informar aos roteadores intermediários a atualização corrente do provedor. A atualização da posição do provedor é feita diretamente nas FIBs dos roteadores que estiverem no caminho do PoA antigo. Nessa proposta, a atualização é rápida e a mensagem de Interesse especial não afeta o desempenho da rede, por ser pequena. No entanto, o esquema é bastante dependente de tecnologia e o caminho atualizado pode não ser o menor possível.

A proposta de L. Wang (2013) implementa um esquema de roteamento no CCN para implementar mobilidade baseada em algoritmos gulosos (*Greedy Algorithm*). A proposta sugere um novo tipo de identificação do pacote CCN, começando com “greedy:” ao invés do padrão “ccnx:”. A proposta usa pacotes especiais para atualizar as FIBs diretamente. A avaliação apresentada pelo trabalho mostrou que o algoritmo é superior em relação a alguns protocolos existentes. No entanto, essa solução faz acréscimos ao padrão CCN que podem dificultar sua implementação e compatibilidade com o CCNx.

Os autores em Woo (2014) propõem um esquema de mobilidade chamado PNPCCN (*Proactive Neighbor Pushing CCN*). Quando o provedor está perto de iniciar o processo de *handover*, ele envia seu conteúdo para os nós vizinhos. O conteúdo é enviado apenas aos vizinhos que ainda não tenham tal conteúdo. Essa sincronização de dados é feita usando funções SCAN e PUSH que enviam mensagens especiais de Interesse e Dados com o objetivo de transferência de informações. A função de transferência usa um cálculo chamado “Probabilidade de Transferência” (do inglês, *Transfer Probability*) que usa

conceitos de popularidade e raridade de conteúdo. O algoritmo foi avaliado através de simulações e comparado com o CCN tradicional. Embora pareça promissor, o trabalho viola a filosofia do CCN quando decide não descartar um pacote de dados cujo nome não esteja na PIT. Isso pode causar problemas de negação de serviço e inundação de pacotes. A proposta é dependente de topologia, e o algoritmo apenas funciona se o nó de destino for um vizinho do nó de origem. O trabalho propõe um espalhamento do conteúdo, mas não descreve como atualizar as rotas para a nova localização do provedor.

3.2.1.2. Abordagens Baseadas no protocolo MIP (I-MIP)

As soluções desse grupo são baseadas no protocolo Mobile IP (Perkins, 1997) onde um túnel é criado entre o antigo PoA, chamado roteador doméstico (do inglês, *Home Router*), e o novo PoA, chamado roteador estrangeiro (do inglês, *Foreign Router*). A Figura 3.3 mostra a sinalização genérica para essa proposta. Após o *handover*, uma mensagem de alerta de *mobilidade* é enviada do novo PoA (PoA-1 na Figura 3.3) para o antigo PoA (PoA-2 na Figura 3.3), com o objetivo de criar um túnel entre os dois (Evento 5). Quando um pacote de Interesse for enviado ao antigo PoA (Evento 6), ele será redirecionado ao novo PoA através do túnel (Evento 7). Algumas propostas utilizam técnicas de otimização de caminho (Nikander P. A., 2003) para eliminar o túnel e evitar o roteamento triangular (Evento 8).

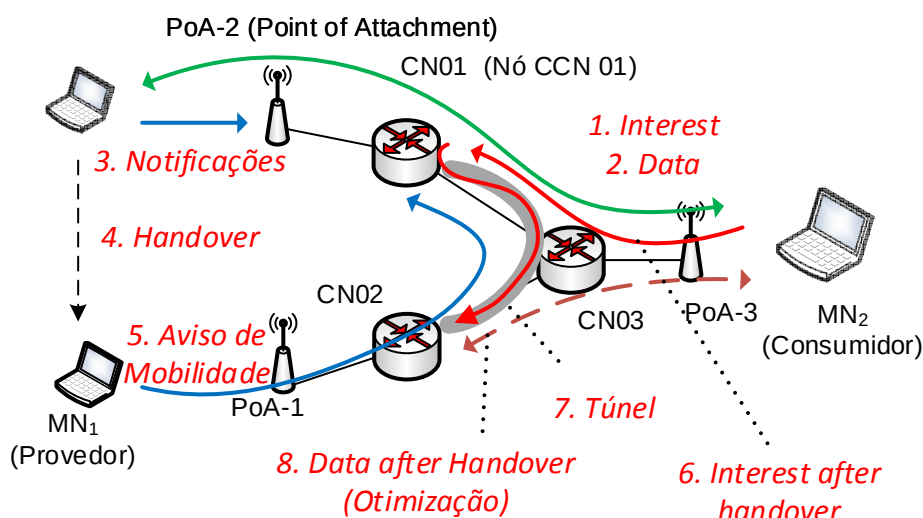


Figura 3.3 Sinalização Genérica para o Grupo I-MIP.

A proposta em Lee (2012) apresenta uma solução onde a ideia principal é redirecionar os pacotes de Interesse para um túnel entre o CR (*Content Router*, ou nó CCN) da rede doméstica do provedor e o CR da rede estrangeira. Os autores assumem que o CR distribui seus prefixos periodicamente para anunciar sua presença e os dispositivos CCN móveis podem detectar se o ponto de acesso foi trocado por meio da comparação entre os nomes dos CRs anunciados. As propostas desse grupo podem sofrer de problemas como o *overhead* de mensagens de anúncios de roteadores, alta latência de *handover*, o nó móvel

pode demorar a perceber a mobilidade, e roteamento triangular, uma vez que há criação de túneis.

O trabalho de Han (2014) introduz dois tipos de entrada de FIB: entrada original e entrada de mobilidade. A entrada original é a mesma da FIB do padrão CCN, enquanto a entrada de mobilidade mantém a informação do próximo salto de destino do dispositivo móvel. As entradas originais são mantidas pelo protocolo de encaminhamento, enquanto as entradas de mobilidade são mantidas pelo PMC (*Publisher Mobility Support Protocol in CCN*). O primeiro possui um valor de tempo de vida relativamente longo e o segundo, um tempo de vida mais curto. Para configurar o caminho do PoA antigo para o novo PoA para onde o provedor migrou, o provedor envia pacotes com os nomes reservados. A proposta utiliza uma grande quantidade de mensagens para garantir a mobilidade. Além disso, os nós CCN são modificados para armazenar os pacotes de Interesse durante a mobilidade, a fim de retransmitir para a rede de destino.

O trabalho de Hermans (2012) usa as ideias das entidades do protocolo MIP para propor soluções para a mobilidade de provedor. Como no MIP, a sua abordagem usa a âncora de mobilidade (do inglês, *Mobility Anchor*), mas não usa encapsulamento. Na proposta, o agente doméstico torna-se o "*Home Repository*". O provedor móvel pode obter um endereço temporário na rede estrangeira, como *Care-of-Address* do MIP. Na proposta, os pacotes de Interesse e de Dados são modificados, com acréscimo de alguns campos. No pacote de Interesse são acrescentados dois campos. No primeiro, o *Nome de Localização* (do inglês, *Location Name*), o qual armazena as informações de localização de origem do pacote, e no segundo, informação da ligação (do inglês, *Binding Info*) entre o nome do conteúdo e a localização do provedor. O esquema adiciona dois novos campos nas mensagens CCN, além de modificar o processo de roteamento. Um dos pontos negativos dessa proposta é o fato de o consumidor precisar saber da localização do provedor, indo de encontro à filosofia ICN.

Luo (2012) apresenta três propostas para a mobilidade de provedor. A primeira introduz um servidor adicional (servidor local) para fornecer gerenciamento de mobilidade e alcançabilidade. Esse servidor é semelhante ao *Home Agent* do *Mobile IP*. A proposta também usa nomes de conteúdos temporários, semelhantes ao *Care-of-Address* do MIP. A segunda proposta é semelhante à primeira, porém fornece mecanismos proativos para antecipar o *handover* e reduzir atrasos no processo de mobilidade. A terceira apresenta mensagens de sinalização especiais para atualizações das rotas. Os autores assumem alguns pré-requisitos, tais como o dispositivo móvel precisa ter mais de uma interface para se conectar em diferentes redes simultaneamente ou pode se conectar com várias redes com apenas uma interface. A topologia apresentada no trabalho possui apenas um roteador CCN que trata de todo o tráfego de rede. O trabalho toma como base várias suposições e não há avaliação da proposta. Assim, não é possível saber o desempenho da mesma. Além disso, a duplicação de pacotes de Interesse criada pelo mecanismo de entrega proativa poderia gerar

sobrecarga. O trabalho de Luo (2014) é uma versão estendida de Luo (2012), porém com avaliação de QoE (*Quality of Experience*) entre a proposta e o padrão CCN.

O trabalho apresentado em Rao (2014a) propõe o PBMA (*Proxy Based Mobility Approach*), entidade que gerencia a mobilidade do provedor de conteúdos. Quando o provedor detecta *handover* iminente, ele envia um Interesse modificado com a opção "*Handover Iminente*". Após o recebimento do Interesse especial, o chamado *home-proxy* muda um estado da entrada FIB para "em mobilidade" e, em seguida, o *Content Store* apenas armazena os pacotes de Interesse que chegam e foram encaminhadas para o provedor. Quando o provedor finaliza a migração, envia uma mensagem "*handover finalizado*" para o *home-proxy*. Ele então atualiza a entrada na FIB, envia os Interesses armazenados em *cache* e avisa o consumidor da nova localização do provedor. A proposta presente no trabalho estende-se para *handover* Inter-domínio. Nesse caso, uma entidade chamada *foreign-proxy* é responsável pelo gerenciamento da mobilidade do provedor na rede estrangeira. Dois outros trabalhos dos mesmos autores apresentam ideias semelhantes, Rao (2014b) e Rao (2014c).

O artigo de Hu (2014) propõe um esquema chamado SD-PAM (*Domain Proxies based Provider Mobility Management*), onde *proxies* agem como *gateway* de domínio e gerenciam todos os pacotes que passam por ele. Na arquitetura proposta, o DP (*Domain Proxy*) é usado para armazenar pacotes de Interesse, enquanto o provedor está migrando para outra rede. Eles podem enviar esses pacotes após a mobilidade, evitando perdas por conta de desconexão. O cenário é dividido em domínios e cada domínio é gerenciado por um DP. Os nós CCN foram modificados para suportar a operação de armazenamento de pacotes de Interesse não satisfeitos, uma *flag* ligada dentro da PIT dos DP indica que essa operação está ativa. A avaliação foi feita no simulador ndnSIM (ndnSIM, NS-3 based Named Data Networking (NDN) simulator, 2017) e comparado com o CCN tradicional utilizando duas métricas: tempo de recuperação de pacotes e taxa de entrega. O esquema proposto foi classificado como I-MIP porque o DP utiliza a ideia de LMA (*Local Mobility Anchor*) do protocolo PMIP (*Proxy-MIP*). Embora baseado em um protocolo de mobilidade mais robusto que o MIP, como PMIP, o artigo não apresenta uma avaliação abrangente.

3.2.1.3. Entidades Centralizadas (E-C)

As soluções desta categoria estão crescendo e muitas propostas em ICN como roteamento, armazenamento e mobilidade têm sido investigadas com soluções centralizadas. Esta categoria pode ser subdividida em: *Rendezvous*, Controlador e *Proxy*. A Figura 3.4 detalha a sinalização genérica para as soluções desta abordagem. A diferença para as demais categorias é a entidade centralizada que pode apenas armazenar informações ou controlar toda a rede.

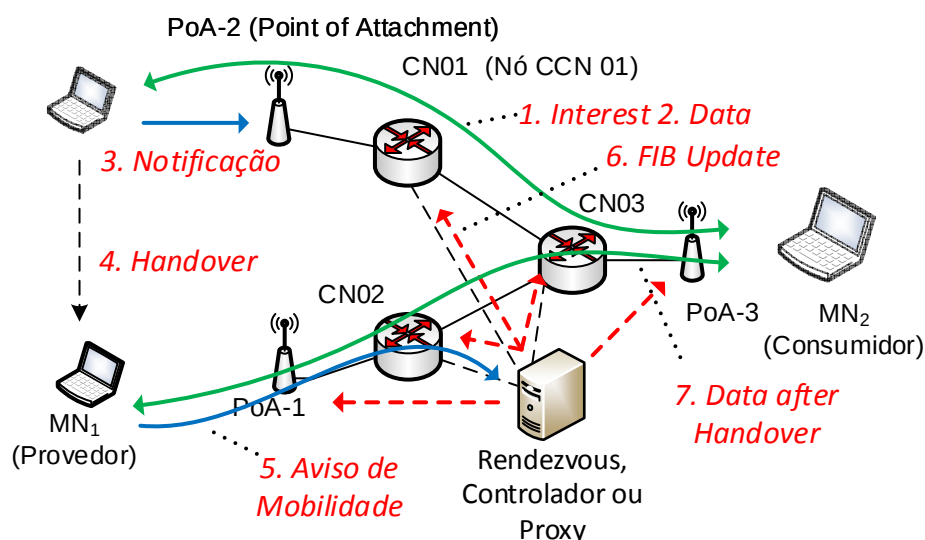


Figura 3.4 Sinalização Genérica para Entidades Centralizadas

a) Baseadas em *Rendezvous*

O *Rendezvous* é um servidor que mantém o controle da informação sobre a localização de todos os clientes da rede. O *Rendezvous* oferece um serviço de nomes responsável pela localização de informações das entidades comunicantes da rede. Nesta subcategoria, a mensagem de aviso de mobilidade, mostrada na Figura 3.4, é uma consulta feita a partir de nós CCN ao *Rendezvous* para adicionar algumas novas informações, neste caso, sobre a mobilidade. Mas, esta consulta poderia ser para obter informações sobre outro nó de rede.

A pesquisa de Jiang (2012) adiciona informações de localização nos pacotes de Interesse e depende de um sistema que mantém o mapeamento do nome do conteúdo e das informações de localização do mesmo. Em seu esquema, um provedor atualiza suas informações de localização para um sistema de mapeamento semelhante ao DNS. Em seguida, um pacote de Interesse primeiro visita o sistema de mapeamento para saber a localização do provedor, para posteriormente ser encaminhado para o destino correto. Com essa abordagem é possível encontrar a localização atual do provedor após sua mobilidade. No entanto, o controle de todos os pacotes de Interesse pela entidade centralizada pode gerar uma sobrecarga significativa. Além disso, os principais benefícios do CCN, como roteamento baseado em nomes, desaparecem uma vez que a localização e a identificação do conteúdo estão fortemente acopladas. Outros artigos que apresentam ideias similares são Luo (2012), Ravindran (2012), Kim (2012) e Kim (2014).

O artigo de Tang (2014) sugere uma separação entre o roteamento de Interesse e de identificação de conteúdo. A proposta toma emprestadas ideias do trabalho de Hermans (2012) e acrescenta um campo de localização no pacote

Interesse. Este campo é utilizado apenas para o encaminhamento do conteúdo. O esquema usa localizadores globais de conteúdo que estão ligados aos nomes dos prefixos. A localização dos conteúdos é gerida pela entidade RH (*Resource Handler*) e é armazenada no chamado RP (*Rendezvous Point*). Todos os pedidos passam por esta entidade central; na mobilidade, o RP não muda, portanto, a mobilidade é transparente.

b) Baseadas em Controladores

Algumas soluções são baseadas em controladores centralizados. Os nós e os conteúdos são registrados nesses controladores, que utilizam essas informações para criar tabelas de roteamento e para gerenciar tanto a mobilidade do provedor quanto do consumidor. Algumas destas propostas são baseadas nas ideias de Redes Definidas por Software (Arumaithurai, 2015). A principal diferença entre o *Rendezvous* e o controlador é que o primeiro tem apenas informações de rede, tais como mapeamento entre o nome e a localização do provedor, e o segundo tem o controle total de todos os elementos da rede.

A pesquisa de Zhou (2014) propõe uma arquitetura centralizada, chamada MobiNDN, para lidar com a mobilidade de provedor. A entidade gerenciadora é chamada de controlador e, na proposta, cada controlador pode lidar com alguns nós agrupados em um Sistema Autônomo. Algumas mensagens especiais são criadas para o registro, exclusão, atualização e descrição das regras que norteiam o comportamento de conteúdo na rede. O controlador gerencia a mobilidade através da CMRT (*Content Mobility Record Table*), uma tabela que mantém o nome, a versão, antiga localização e nova localização do provedor. Esta tabela é semelhante à FIB, mas é controlada pela entidade centralizada, semelhante à tabela de fluxo do OpenFlow (Lara, 2014). A avaliação, realizada no ndnSim, mostra que o MobiNDN pode reduzir o atraso do *handover* e entrega dos pacotes, se comparados com o padrão CCN.

c) Baseados em Proxies

Nesta categoria, o nó central serve como uma ponte entre os elementos da rede. Todos os pacotes de Interesse passam por essa entidade. Assim, o *proxy* tem o controle de mensagens e pode fazer a gestão mais eficiente da mobilidade do provedor. Na Figura 3.4, os pacotes de Interesse e de dados passam pelo *proxy* em toda a comunicação. Em umas das soluções presentes em Kim (2012) é proposto um tipo de servidor proxy chamado *Indirection Point*. Ele intercepta mensagem de Interesse e simula ser o consumidor para trocar tráfego com o provedor. Após a mobilidade do provedor, o *Indirection Point* é o único que precisa ser atualizado sobre a nova localização do provedor, sendo assim, todos os novos pacotes de Interesse são reencaminhados automaticamente para a nova localização do provedor. Algumas das soluções propostas nos trabalhos de Lee (2012), Ravindran (2012), Kim (2014), Hermans (2011), Rao (2014a), Rao (2014b) e Rao (2014c) apresentam ideias semelhantes.

A Tabela 3.1 ilustra o resumo das características encontradas nos três grupos de soluções discutidos.

Tabela 3.1 Tabela comparativa dos grupos de soluções

	AD-FIB	I-MIP	E-C
Tempo de <i>Handover</i>	Baixo	Moderado	Alto
Custo da entrega do pacote	Alto	Alto	Baixo
Roteamento	Próximo ao Ótimo	Distante do Ótimo	Ótimo
Escalabilidade das tabelas de encaminhamento	Baixa	Baixa	Alta
Controle dos caminhos	Baixo	Baixo	Alto
Vantagens	Simplicidade, Rápida atualização de rota	Controle da Mobilidade	Controle da Rede
Desvantagens	Dependência da topologia da rede, baixo controle de mobilidade.	Roteamento triangular, excesso de mensagem, tempo de <i>handover</i> , alto tempo de atualização das rotas.	Único ponto de falha, atraso adicional no tempo de <i>handover</i> .

Como pode ser visto na Tabela 3.1, as latências de *handover* das propostas AD-FIB são geralmente menores porque não há pontos de comunicação adicionais. Assim, o processo de atualização de caminho é feito enquanto o pacote passa pela rede. Aqueles baseados em MIP não possuem um ponto central de gerenciamento. No entanto, muitas vezes precisam informar à rede doméstica sobre a mobilidade do provedor, e isso acrescenta um tempo extra no processo de *handover*. Por outro lado, soluções I-MIP usam agente de mobilidade na borda da rede, o que proporciona um melhor gerenciamento do *handover* do provedor. Como o I-MIP sofre com alta latência de *handover*, técnicas de predição de mobilidade podem mitigar esse problema. A latência de transferência mais elevada pode ser vista nas propostas que utilizam um controle centralizado porque, após a entrega, o controlador precisa ser informado e, em seguida, tem de atualizar todo o percurso de distribuição para a nova localização do provedor. Esse problema também pode ser atenuado com técnicas de predição de mobilidade e prealocação de redes de mobilidade no controlador

A abordagem AD-FIB possui um maior custo de entrega de pacotes por causa do prefixo do nome adicional para rotear pacotes até o destino certo. A abordagem I-MIP também tem um alto custo de entrega de pacotes devido aos túneis criados ou aos nomes de conteúdo temporários. Como a entidade

centralizada tem a visão global da rede, ela pode modificar o caminho sem inserir qualquer custo adicional para entregar o pacote. Nas soluções AD-FIB, a mensagem de atualização modifica todas as FIBs dos roteadores intermediários. Deste modo, as mensagens de Interesse não precisam atravessar todos os roteadores antes de chegar até a origem do conteúdo. Se uma mensagem tentar atualizar uma FIB que já possui a mesma rota da mensagem, quer dizer que esse é um ponto de convergência do caminho e, portanto, a rota consumidor/provedor já está atualizada e o pacote é então descartado. Por causa das noções de tunelamento, o roteamento em soluções I-MIP é distante do ótimo. Em E-C, o controlador pode modificar o caminho a partir de uma de uma visão global da rede e, dessa forma, as rotas serão calculadas da forma mais otimizada possível.

Se não houver algum mecanismo de agregação de prefixo, o tamanho da tabela de encaminhamento nas abordagens AD-FIB irá aumentar a taxas elevadas e o problema pode ser ainda maior quando houver um grande número de *handovers* de provedores na rede. As soluções de I-MIP também sofrem de problemas de escalabilidade já que, na maioria das propostas desse grupo, há uma mudança no nome do prefixo ou há um prefixo temporário com a nova localização do provedor. Em E-C, as tabelas de roteamento se mantêm curtas porque todas as informações podem ser recuperadas a partir da entidade centralizada. Só as propostas centralizadas têm total controle do caminho de entrega. Com informações de todas as entidades de rede, eles podem calcular o melhor caminho para a nova localização do provedor. No entanto, propostas centralizadas geralmente têm a desvantagem de possuir um único ponto de falha. Se um controlador falhar, toda a rede fica inoperante.

3.2.2. Comparação das Estratégias de Mobilidade

Nesta seção, apresentamos uma comparação de todas as estratégias de mobilidade apresentadas neste estudo. Posteriormente, será apresentada uma tabela comparativa entre diversas propostas apresentadas nesse artigo. A Figura 3.5 ilustra a arquitetura utilizada, em particular, para analisar o tempo de *handover* das propostas na Tabela 3.3. Na Figura 3.5 também são destacados os nomes das entidades dadas em cada proposta. Esta arquitetura genérica é usada para fornecer um cenário único de avaliação, a fim de promover uma comparação justa entre as soluções. Quando uma entidade não é usada em uma arquitetura particular, ela ficará oculta, pois não terá interferência na avaliação. Por exemplo, uma entidade centralizada C, apresentada na Figura 3.5, pode ser ignorada no trabalho que não utiliza entidades centrais. A Figura 3.5 também descreve os símbolos referentes à latência entre duas entidades (por exemplo, L_{AC} significa latência entre as entidades A e C).

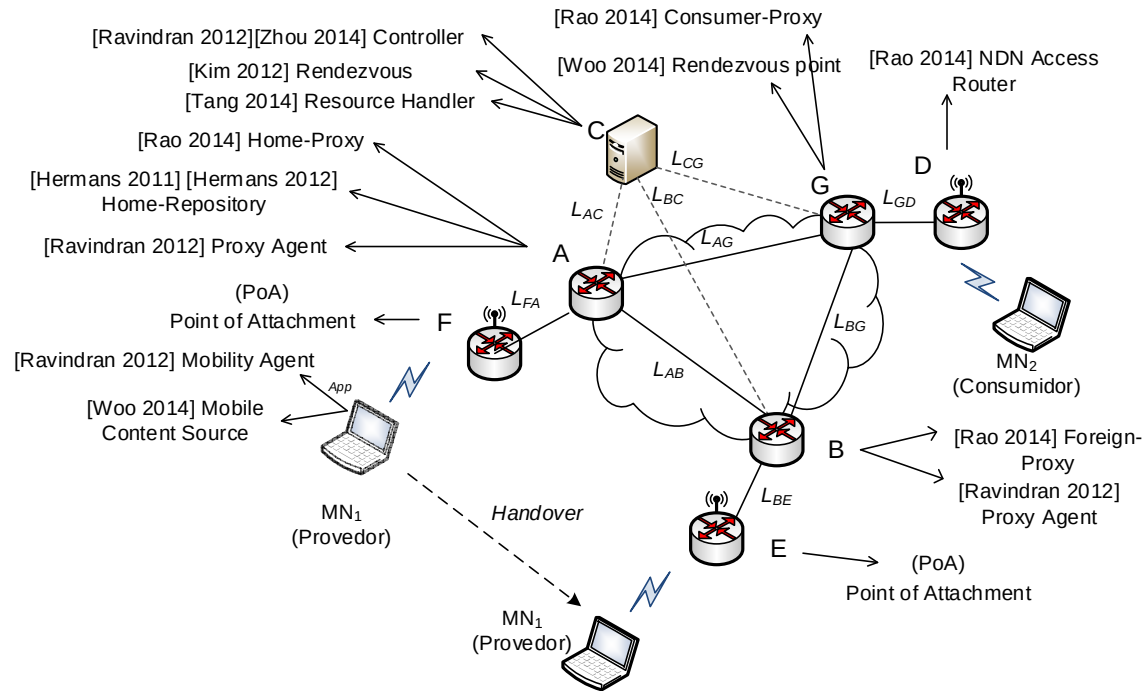


Figura 3.5 Arquitetura Genérica de comparação

Os parâmetros da coluna “Latência de *Handover*” da Tabela 3.3 foram obtidos através da Eq. 1, descrita no início da Seção 3.2. Nessa coluna, o parâmetro t_{FIB_UPDATE} é substituído pela soma dos tempos das mensagens de cada proposta. Alguns parâmetros da Eq. 1 podem ser ignorados, dependendo da solução avaliada. Por exemplo, quando a proposta usa técnicas de predição de mobilidade, o parâmetro $t_{TIMEOUT}$ tende a ser igual a zero. Outros parâmetros específicos apresentados na Tabela 3.3 são descritos na Tabela 3.2.

Tabela 3.2 Lista de Parâmetros Específicos

Parâmetro	Descrição
t_{PUSH}	Tempo necessário para que o conteúdo seja enviado para os nós vizinhos. Este tempo é usado no trabalho de Woo (2014). Foi criada uma única variável porque é parâmetro probabilístico e depende de muitos fatores, tais como topologia, a precisão do algoritmo e a popularidade do conteúdo.
t_{TN}	Na proposta de Luo (2014), o nó móvel na rede estrangeira usa um nome temporário, assim como <i>Care-of-Address</i> no protocolo MIP. O tempo para obtenção deste nome é representado por este parâmetro.
L_{XY}	Latência de transmissão entre o elemento X e o Elemento Y;

A comparação entre as soluções de mobilidade de provedor foi resumida na Tabela 3.3. Os termos usados na comparação seguem a ordem: ideia principal do trabalho, mudanças necessárias no CCN, tempo de duração do *handover*, número de mensagens necessárias, se o trabalho aborda mobilidade intra e/ou

inter-domínio, se há recuperação de pacotes perdidos devido à mobilidade, se há algum tipo de gatilho para o processo de *handover* e como a proposta foi avaliada. Para aqueles trabalhos em que os autores propõem mais de uma solução, será considerada apenas a que possuir melhor desempenho, segundo avaliação dos próprios autores.

Para construir a Tabela 3.3, foi considerado que os dispositivos MN_1 e MN_2 , da Figura 3.5 possuem apenas uma interface ativa. Alguns artigos, como Luo (2012) e Luo (2014), assumem o uso de dispositivos com múltiplas interfaces ativas ou com uma interface que pode se conectar a várias redes. Neste caso, a latência de transferência seria mais baixa do que as outras. As referências da Tabela 3.3 estão organizadas em ordem cronológica.

Tabela 3.3 Tabelas comparativas das soluções de Mobilidade de Provedor

Ref.	Ideia Principal	Mudanças no CCN	Latência de <i>handover</i>	Custo da Sinalização	Domínio	Recuperação de pacotes	Gatilho de <i>handover</i>	Avaliação
(Kim H., 2012)	Atualização da PIT via Encaminhamento de mensagens	Armazenamento de pacotes de Interesse no CS e mensagens especiais	$= t_{TIMEOUT} + t_{AP_DETECT}$	2 msg	Intra	Não	Mensagem de notificação de <i>handover</i> , mas esquema não foi detalhado.	Simulação (número de sequência de pacotes)
(Ravindran, 2012)	Plano de controle com agente de mobilidade, <i>proxy</i> e controlador.	Agentes de mobilidade implementados nos nós CCN.	$= t_{TIMEOUT} + t_{AP_DETECT}$	4 msg	Intra	Sim	Notificação da camada L2, mas não foram dadas maiores explicações.	Sem avaliação
(Hermans F. N., 2012)	Semelhante ao MIP	Adiciona campos nos pacotes de Interesse e de Dados, muda o mecanismo de processamento de pacotes.	$= t_{TIMEOUT} + t_{AP_DETECT}$	1 msg	Inter	Não	Não	Sem avaliação
(Kim D. H., 2014)	Inundação por zona, encaminhamento de múltiplas cópias de Interesse para os candidatos a novo PoA	Mudanças no esquema de roteamento e novas mensagens, Interesse Especial	$= t_{TIMEOUT} + t_{AP_DETECT}$	Inundação de Interesses, várias mensagens dependendo da topologia	Intra	Não	Não	Simulação (número de sequência de pacotes)
(Luo Y. E.-G., 2012) (Luo Y. E.-G., 2014)	Semelhante ao MIP, com um Local Server agindo como <i>Home Agent</i> do MIP	Mensagens especiais e novas entidades na rede	$= t_{AP_DETECT} + t_{TIMEOUT}$	5 msg	Intra	Não	Monitoramento da variação do atraso das mensagens. Se começar a aumentar sugere <i>handover</i> iminente	Sem avaliação no primeiro trabalho e <i>Testbed</i> (Métricas de QoE)

(Han, 2014)	Modificação na FIB, entrada original e de mobilidade.	Nomes reservados e entradas especiais na FIB	$= t_{TIMEOUT} + t_{AS}$	2 msgs	Intra	Não	Não	Simulação (número de sequência de pacotes e atrasos)
(Zhou, 2014)	Controlador Centralizado	Novos tipos de mensagens CCN	$= t_{TIMEOUT} + t_{AS}$	3 msgs (intra), 5msg(inter)	Intra/Inter	Não	Através do registro de conteúdo	Simulação: ndnSIM (número de sequência de pacotes),
(Rao Y. G., 2014a)	Entidades <i>Proxy</i> que controlam todos os pacotes da rede.	Novas entradas na FIB e novos pacotes CCN	$= t_{TIMEOUT} + t_{AS}$;	4 msgs (intra), 6msgs (inter).	Intra/Inter	sim	Sim, mas não houve maiores explicações de como o mecanismo funciona.	Modelagem analítica (Custo e Latência de <i>Handover</i>)
(Woo, 2014)	Disseminação de conteúdos de forma proativa	Adiciona campos nos pacotes de Interesse e de Dados	$= t_{PUSH}$	3 msgs, but no FIB update messages	Intra	Não	Assume que o dispositivo monitora a força do sinal de conexão com o PoA (RSSI ou RCPI ⁶)	Simulação, ndnSIM
(Tang, 2014)	Separa localização/ identificação e plano de dados de plano de controle usando <i>rendezvous</i>	Adiciona campos nas mensagens CCN e acrescenta um novo pacote: “Pacote de Atualização de Localização”	$= t_{TIMEOUT} + t_{AS}$	4 msgs (intra), 8 msgs (inter)	Intra/Inter	Não	Não	Modelagem Analítica
(Hu, 2014)	Entidade <i>Domain Proxy</i> que gerencia a mobilidade dos usuários	Armazenamento de Mensagens de Interesse, <i>flag</i> na PIT para indicar quando iniciar a bufferização	$= t_{TIMEOUT} + t_{AS}$	Sem mensagens de Controle	Intra/Inter	Sim	Não	Simulação, ndnSIM
(Zhang Y. H., 2014)	Usa informações da PIT para configurar o caminho reverso após a mobilidade	Novas tabelas, modificações na PIT e no processamento dos nós NDN.	$= t_{TIMEOUT} + t_{AS}$	4 msgs	Intra	Não	Não	Comparação conceitual e simulação (ndnSIM)

⁶ RSSI (Receive Signal Strength Indicator), RCPI (Received Channel Power Indicator)

(Kim D. T.-X., 2015)	Sinalização de mobilidade e estabelecimento de caminhos de modo antecipado.	Novos campos nas msgs NDN e novas tabelas.	$= t_{TIMEOUT} + t_{AI}$	4 msgs	Intra	Não	Não	Modelagem analítica (Custo e Latência de Handover)
(Augé, 2015)	Gerenciamento de mobilidade feito pelo provedor	Novas tabelas e mensagem de Interesse especial	$= t_{TIMEOUT} + t_{AI}$	4 msgs	Intra	Sim	Não	No
(Lehmann, 2016)	Mapeamento entre Nome e Localização de conteúdo.	Novas mensagens, modificação no processamento padrão dos nós NDN	$= t_{TIMEOUT} + t_{AI}$	5 msg	Intra	Não	Não	Modelagem Analítica

Algumas conclusões podem ser tiradas após a análise da Tabela 3.3: (i) todas as propostas fazem modificações no CCN. (ii) a maioria das propostas possui o foco na mobilidade intra-domínio. (iii) a maioria não possui mecanismos de recuperação de pacotes, as que possuem geralmente o fazem de forma custosa, enviando pacotes especiais para a rede anterior, à qual o provedor estava conectado, para requisitar Interesses não satisfeitos (iv) sempre há latência no processo de *handover*. (v) com exceção de Luo (2014), que avaliou em um *testbed* superficial sua proposta, nenhum outro trabalho fez avaliação através de *testbed*.

3.3. Considerações Finais

Nesse capítulo, conclui-se que, mesmo com a ineficiência do MIP e suas variações, muitas propostas de mobilidade de provedor são fortemente baseadas nesse protocolo. Não se sabe se mesmo com a falha do MIP, propostas com ideias baseadas no protocolo terão sucesso nas redes CCN. O estudo mostrou que apesar das redes centralizadas terem uma visão melhor da rede, a latência de *handover* nesse grupo é maior e, por esse motivo, ainda não são amplamente utilizadas na borda. Outra conclusão do estudo foi que propostas baseadas na atualização direta da FIB são mais simples, rápidas e menos custosas para a rede. Por esse motivo, o esquema proposto nesta tese será baseado no grupo AD-FIB.

4

ALGORITMO DE RANQUEAMENTO DE *CACHE*

Este capítulo aborda o RankCache, nome dado ao algoritmo de decisão de armazenamento de *chunks* proposto por esta tese. O objetivo do RankCache é prover entrega eficiente de conteúdo e armazenamento otimizado. O algoritmo distribui os *chunks* considerando o *ranking* de cada *cache*. O esquema foi avaliado através de simulação e o resultado mostra sua superioridade em relação a algumas propostas da literatura.

4.1. Introdução

O esquema proposto para otimizar o armazenamento de pedaços de conteúdo (chamados de *chunks*) nos elementos da rede foi nomeado RankCache. O objetivo do RankCache é prover entrega eficiente de conteúdo e armazenamento otimizado, com redução de sobrecarga de gerenciamento e cooperação implícita entre os nós da rede. O algoritmo RankCache distribui os *chunks* considerando o *ranking* de cada *cache*, que no CCN pode ser chamado de *Content Store* (CS), *Content Router* (CR) ou apenas *cache*. As principais características do RankCache são:

- a) *Ranking* baseado na Popularidade: O algoritmo ajusta o valor *dos rankings* de cada *cache* em um dado caminho. O *cache* com maior valor de *ranking* possuirá a maior probabilidade de armazenar um *chunk* do que um *cache* com menor *ranking*. Esse valor é calculado usando o WSI (*Wilson Score Interval*) (Wilson, 1927), que é uma fórmula para calcular o intervalo de confiança para proporção binomial que, nesse caso, é caracterizada como sucesso *versus* fracasso ou *hit versus miss*. Na proposta, a variável de sucesso é baseada no número de acertos de um *cache*. Quanto mais *chunks* populares passarem pelo *cache*, maior será o número de acertos dentro dele, e consequentemente, maior será sua probabilidade de armazenar outros *chunks*.

- b) *Cache* Descentralizado: No RankCache, a decisão de guardar ou não um *chunk* é feita de forma independente. Cada *cache* calcula sua probabilidade de armazenamento sem a interferência de outros *caches*. Assim, não há nenhum gerenciamento centralizado. Dessa forma, a proposta é escalável, pois não requer conhecimento a priori da topologia da rede.

4.2. Algoritmo de Armazenamento em redes CCN baseado no *Ranking* do *Cache*.

O RankCache é um algoritmo de ranqueamento de *cache* que distribui os *chunks* na rede de forma distribuída e baseada na popularidade dos *chunks*. Ele usa o número de acertos no *cache* para calcular um valor de *ranking* para o dado *cache*. No RankCache proposto neste trabalho, o armazenamento dos *chunks* é proporcional ao *ranking* de cada *cache*. O *ranking* de um dado *cache* é calculado com base no número de requisições de *chunks* repetidos que passam através do *cache*.

O problema de calcular o *ranking* de vários *caches* independentes, usando a proporção de número de acertos *versus* número de erros, não é simples como usar um sistema de classificação médio (número de acertos dividido pelo número total de pacotes). Por exemplo, com classificação média, um *cache* com 3×10^3 acertos e 6×10^3 pacotes no total (acertos + erros) terá o mesmo valor de *ranking* que um *cache* com 6 acertos e 12 pacotes, e isso não é uma representação justa. Neste trabalho, usou-se um cálculo mais sofisticado para criar o *ranking* de *caches*. O cálculo leva em consideração dois fatores: a proporção binomial e o valor de proximidade.

4.2.1. Intervalo de Confiança para a proporção binomial

No campo da estatística, o intervalo de confiança para a proporção binomial é uma estimativa amostral de uma variável aleatória X , onde X representa a presença ou não de uma determinada característica de uma população. Em geral, existem várias fórmulas para o intervalo de confiança binomial, mas a maioria assume que a amostra pode ser modelada como uma distribuição binomial.

A variável aleatória X pode ser representada como uma distribuição de Bernoulli com parâmetro p , onde p é a probabilidade de um elemento da amostra possuir a característica de interesse que, no problema abordado, representa o acerto no *cache*. Retirando-se uma amostra aleatória $X_1, \dots, X_n$ da população, cada $X_i, i = 1, \dots, n$ possui distribuição de Bernoulli com parâmetro p , isto é:

$$X_1, \dots, X_n \sim \text{Bernoulli}(p)$$

Onde:

- A média $\mu = p$
- A variância $\sigma^2 = p(1 - p)$

Nesse caso, o valor estimado $\hat{p}$ do parâmetro p é dado por:

$$\hat{p} = \frac{\sum_{i=1}^n x_i}{n} \quad \text{Eq. 2}$$

Para calcular o intervalo de confiança do parâmetro $\hat{p}$, utilizou-se uma aproximação da distribuição normal. Seguindo o Teorema Central do Limite, pode-se dizer que com um tamanho considerável de amostras é possível aproximar a proporção amostral de $\hat{p}$ usando uma distribuição normal com média p e variância $p(1 - p)/n$. Deste modo, pode-se representar $\hat{p}$ como:

$$\hat{p} \sim N\left(p, \frac{p(1 - p)}{n}\right) \quad \text{Eq. 3}$$

A variância de $\hat{p}$ depende do parâmetro p , porém esse último não é conhecido, pois está relacionado à população. Deste modo, assumindo-se que n é muito grande, p pode ser substituído por $\hat{p}$. Como n representa o número total de pacotes que passam por um *cache*, de fato, é factível supor que n seja muito grande. Com isso, adequando a Eq. 3 na forma $Z = (x - \mu) / \sigma$ pode ser escrita como:

$$\frac{\hat{p} - p}{\sqrt{\frac{\hat{p}(1 - \hat{p})}{n}}} \sim N(0,1) \quad \text{Eq. 4}$$

Sendo assim, o intervalo para a média da proporção p , com $100(1 - \alpha)\%$ de confiança é dada por:

$$IC(p, 1 - \alpha) = \left(\hat{p} - Z_{\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1 - \hat{p})}{n}}, \hat{p} + Z_{\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1 - \hat{p})}{n}} \right) \quad \text{Eq. 5}$$

O matemático americano Edwin Wilson provou que a equação Eq. 5 não possui uma boa aproximação com um número baixo de amostras e/ou com valores extremos de probabilidade (Edwardes, 1994). E se ela for usada como forma de ranqueamento, não irá modelar a relação sucesso/fracasso de forma justa. O problema com essa abordagem é que ela não descreve de forma correta

o número crescente de ocorrências da característica de interesse. Por exemplo, com classificação, um *cache* com 3×10^3 acertos e 6×10^3 pacotes no total (acertos + erros) terá o mesmo valor de *ranking* que um *cache* com 6 acertos e 12 *chunks*.

Wilson aprimorou a Eq. 5 para caracterizar a relação sucesso/fracasso de forma mais eficiente. E o resultado é dado pela Eq. 6. Percebe-se, nessa equação, que o valor n ganhou uma representatividade maior em relação à Eq. 5. Desse modo, a relação sucesso/fracasso está mais dependente do número total de observações.

$$IC(p, 1 - \alpha)_{wilson} = \left(\frac{\hat{p} + \frac{z_{\alpha/2}^2}{2} - z_{\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1 - \hat{p})}{n} + \frac{z_{\alpha/2}^2}{4n^2}}}{1 + \frac{z_{\alpha/2}^2}{n}}, \frac{\hat{p} + \frac{z_{\alpha/2}^2}{2} + z_{\frac{\alpha}{2}} \sqrt{\frac{\hat{p}(1 - \hat{p})}{n} + \frac{z_{\alpha/2}^2}{4n^2}}}{1 + \frac{z_{\alpha/2}^2}{n}} \right) \quad \text{Eq. 6}$$

4.2.2. RankCache

A equação Eq. 6 foi adaptada neste trabalho para calcular o *ranking* do *cache* usando o número de acertos no respectivo *cache*. Essa fórmula, denominada Proporção Binomial do i -ésimo *cache* (PBC_i), é dada por:

$$PBC_i = \frac{\rho_i + \frac{1}{2n_i} z_{1-\alpha/2}^2 \pm z_{1-\alpha/2} \sqrt{\frac{\rho_i(1 - \rho_i)}{n_i} + \frac{z_{1-\alpha/2}^2}{4n_i^2}}}{1 + \frac{1}{n_i} z_{1-\alpha/2}^2}$$

Onde:

- ρ_i é o número de acertos observados no *cache* i .
- n_i é o número total de *chunks* que passam pelo *cache* i .
- $z_{1-\alpha/2}^2$ é o valor padronizado da distribuição normal para um dado Intervalo de Confiança.

O cálculo do *ranking* considera a contagem de ocorrências de sucesso como uma amostra estatística a ser estimada. O algoritmo infere, em um intervalo de confiança, um valor de *cache* baseado na estimativa da ocorrência de *acertos* naquele *cache*. Usou-se como Intervalo de Confiança o valor 0,95 (95%), o que dá um valor padronizado ($z_{1-\alpha/2}^2$) igual a 1,96.

Além da inferência de proporção binomial, elaboramos nesta tese um cálculo do *ranking* usando o que denominamos de Fator de Proximidade (*FP*). Esse fator aumenta a probabilidade de um *chunk* ser armazenado perto do consumidor. Em (Psaras, 2011) é apresentado um estudo que mostra que os conteúdos tendem a ser armazenados por mais tempo nas bordas da rede, o *FP* ajuda a modelar esse comportamento. Quanto mais perto o conteúdo estiver na borda, maior será o *FP*. O Fator de Proximidade do *i*-ésimo *cache* (FPc_i) é calculado por:

$$FPc_i = \frac{\sum_{j=1}^{H_c} H_j}{\sum_{i=1}^{H_c} H_i}$$

Onde:

- H_c é o número atual de saltos após um acerto
- H_t é o número de saltos totais do pacote até a ocorrência do acerto.
- H_j e H_i são valores somados a cada salto dos pacotes. Alguns saltos podem ter pesos por alguma razão, porém, neste trabalho, foram usados saltos unitários.

Para exemplificar o funcionamento do FPc_i , suponha que um *chunk* se encontra no sétimo *cache* de um dado caminho. No primeiro salto do pacote de dados $FPc_7 = 1/7$, no segundo salto $FPc_6 = 2/7$, no terceiro $FPc_5 = 3/7$ e assim por diante, até chegar ao nó requisitante, que não entra no cálculo, ou seja, em nenhum momento o valor de FPc_i será igual a 1 (7/7). Percebe-se que o FPc_i é incrementado com a proximidade do nó requisitante, isso ajuda a aumentar a probabilidade de armazenamento na borda da rede.

O valor de H_c pode ser obtido através do campo TTL (*Time To Live*) da mensagem CCN. Para possibilitar todos os cálculos necessários, o RankCache requer a adição apenas do H_t no cabeçalho da mensagem CCN. Depois de um *acerto*, o valor de H_t pode ser calculado através do valor de H_c e será gravado na mensagem de resposta.

A fórmula final para o cálculo da probabilidade de guardar o *chunk* no *i*-ésimo *cache* é dada pela seguinte equação:

$$PRc_i = PBC_i * FPc_i$$

O Algoritmo 1 descreve como a função de decisão utiliza a fórmula PRc_i para o armazenamento de um *chunk*. Essa função recebe como parâmetro a mensagem CCN de chegada e retorna sua decisão como um valor booleano.

Algoritmo 1: RankCache

1. **Entrada: Mensagem CCN** //recebe como parâmetro a mensagem CCN :
 2. **Saída: booleano decisao** // Retorna um valor lógico indicando a decisão.
 3. *decisao* ← falso.
 - 4.
-

```

5. //Define acertos como o número total de acertos deste cache. Ele precisa
   ser um contador com escopo global.
6.  $acertos \leftarrow Acertos\ Totais$ 
7.
8.  $n \leftarrow n + 1$  //Incrementa o número total de chunks n.
9. obter o valor de Hc da mensagem CCN;
10. obter o valor de Ht da mensagem CCN;
11.
12.  $R_p \leftarrow H_c/H_t$ 
13. //PBC também é global para este cache, só é atualizado em um intervalo
   de tempo pré-definido para evitar overhead de atualização.(Linhas 27-30)
14.  $P_{Rc} \leftarrow P_{Bc} * R_p$ 
15.
16. //Gera um número randômico x
17.  $x \leftarrow$  Número Aleatório entre 0 e 1.
18. se  $x$  for menor que Prc então:
19.    $decisao \leftarrow verdadeiro$ ;
20. fim se
21.
22. se decisao for verdadeiro então:
23.   Incrementa o valor do número de acertos
24. fim se
25.
26. //atualiza o rank do cache.
27. se o intervalo de atualização do Rc foi alcançado então:
28.    $phat \leftarrow 1.0 * acertos / n$ ;
29.    $P_{Bc} \leftarrow (phat + z * z / (2 * n) - z * \sqrt{(phat * (1 - phat) + z * z / (4 * n)) / n}) / (1 + z * z / n)$ ;
30. fim se
31.
32. retorna decisao

```

Primeiramente, são obtidos os valores H_c e H_t através da mensagem CCN (linhas 9-10). Posteriormente, é calculada a probabilidade de armazenamento do *chunk*. Nota-se que, no algoritmo 1, “ P_{Bc} ” é uma variável global e é inicializada em “1”. Se a decisão de guardar o *chunk* for positiva, a variável “*acertos*”, que guarda o número de *acertos*, é incrementada (linha 23). O *ranking* do *cache* é atualizado, através do cálculo do “ P_{Bc} ”, em intervalos previamente definidos, para evitar o *overhead* de atualizações desnecessárias (linha 29). Ao final das operações, a função retorna a decisão de armazenamento.

Esse procedimento, detalhado pelo algoritmo 1, é invocado por uma entidade que gerencia as políticas de *cache*. Caso a decisão de armazenamento seja positiva e não haja mais espaço no *cache*, o algoritmo de reposição entra em ação para trocar um *chunk* armazenado pelo que acabou de ser escolhido. A Seção 4.3 apresenta a avaliação do algoritmo proposto.

4.3. Avaliação do RankCache

Para avaliar o RankCache, foi desenvolvido um simulador de rede CCN utilizando-se o *framework* OMNET++ (Varga, 2017) e com base em algumas ideias de dois outros simuladores: o ccnSim (Chiocchetti, 2013) e CCN-Lite (CCN-lite, 2016), ambos também implementados no OMNET++. O primeiro é um simulador em nível de *chunk* e possui um alto grau de abstração, por isso, torna-se difícil a avaliação de políticas de roteamento e de distribuição de prefixos; e métricas relacionadas à camada de rede como atrasos, perdas de pacotes e vazão.

Por outro lado, o CCN-Lite não possui a implementação da camada de núcleo CCN, onde residem a PIT (*Pendent Interest Table*), a FIB (*Forwarding Information Base*) e o CS (*Content Store*). No CCN-Lite, o núcleo CCN é implementado fora do OMNET++, de modo que possa ser aproveitado por outras plataformas como Linux e Raspberry Pi. Apesar de ser uma vantagem, em alguns casos isso dificulta bastante a implementação de novas propostas que envolvam o núcleo CCN, como roteamento, políticas de decisão e de armazenamento, por exemplo. No simulador desenvolvido, foi implementado todo o núcleo CCN, incluindo as tabelas PIT, FIB e CS. Além disso, o protocolo OSPFN para roteamento e distribuição de prefixos também foi uma contribuição desenvolvida para o simulador. O simulador também permite a extração de métricas de rede como atraso, perdas de pacote e vazão.

O experimento foi conduzido de acordo com os parâmetros e configurações descritos na Tabela 4.1. O LRU foi usado como política de substituição. A distribuição Zipf (Adamic, 2000) foi empregada como modelo de popularidade de conteúdo. O Zipf é uma das distribuições mais usadas para modelar esse comportamento e é, de acordo com alguns estudos, a que melhor explica as estatísticas de *download* de conteúdo de serviços de vídeo sob demanda, como o *Youtube* (Cha, 2009).

Tabela 4.1 Configuração dos parâmetros dos experimentos.

Rede	27 nós (Rede IPÊ)[RNP 2014]
Atrasos do enlace	10-100ms (uniformemente distribuídos)
Intervalo de Atualização (PBC)	15s
Capacidade de Processamento do nó	50 pacotes/s
Taxa de transmissão	Exponencial, média = 1/80
Tamanho do catálogo ⁷	10 ⁵ files
Tamanho médio dos conteúdos	100 <i>chunks</i>
Tempo de simulação	10 ³ s

⁷ Faz referência ao número de conteúdos únicos.

Foi utilizada, como cenário de simulação, a topologia da rede IPÊ (RNP, 2017). Ela possui 27 nós distribuídos, através de PoPs (*Points of Presence*/Pontos de Presença), por todos os estados brasileiros. A Figura 4.1 mostra a topologia construída no simulador e usada nas simulações.

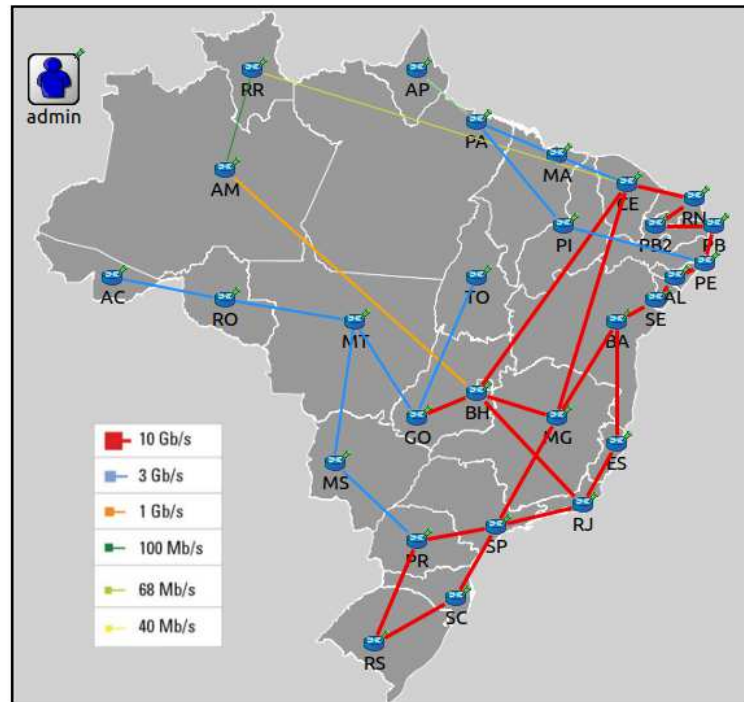


Figura 4.1 Topologia da Rede IPÊ implementada no simulador (RNP, 2017)

Para todos os experimentos foram realizadas 30 simulações. Os valores apresentados nas Figuras 15 a 18, foram obtidos com o expoente da distribuição Zipf iguais a 0.5, que é o valor médio usado em simulações desse tipo. Quanto menor o Zipf, menor o número de conteúdos populares no catálogo (Psaras b, 2012).

A Figura 4.2 apresenta a taxa média de acertos em relação à variação da capacidade dos *caches*. Como pode ser visto na figura, o RankCache possui uma taxa de acertos maior que todas as outras propostas. É superior em aproximadamente 15% em relação à média do Fix0.3 e mais de 17% em relação à média do *ProbCache*.

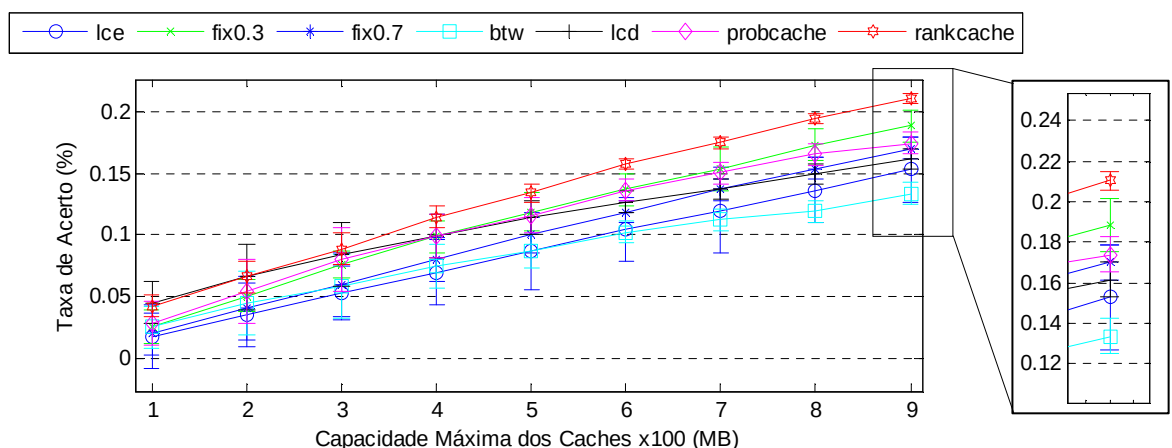


Figura 4.2 Taxa de Acertos comparada com a capacidade dos caches variando de 100 MBytes a 900 MBytes.

A Figura 4.3 mostra o número de saltos médios necessários para ocorrer um acerto. Essa métrica está relacionada à distância média entre o usuário que requisita o conteúdo e onde o conteúdo é encontrado. Quanto menor for o valor dessa métrica, menor será o caminho percorrido pela mensagem de requisição e de resposta. Além disso, mais rápido o usuário conseguirá o conteúdo requisitado. O número de saltos médio do RankCache, com 900Mb de *cache*, foi de aproximadamente 2,8 saltos, contra 3,2 saltos da segunda melhor proposta (*Probcache*). Nessa Figura, o LCE possui o pior desempenho, isso porque o LCE armazena todos os pacotes de forma indiscriminada e sem distinção de nível de popularidade, proximidade ou outro tipo de característica.

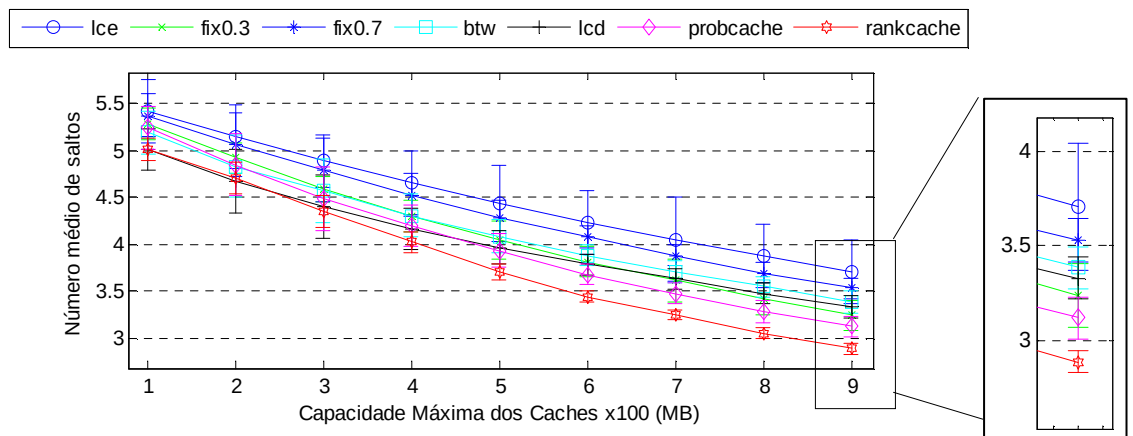


Figura 4.3 Número de saltos até um acerto em relação à capacidade dos caches variando de 100 MBytes a 900 MBytes.

Outra importante métrica para avaliar o desempenho de algoritmos de *cache* é a quantidade de acertos no servidor, ou seja, no provedor do conteúdo. Quanto maior o número de acertos no servidor, menor terá sido a eficiência do algoritmo em distribuir os *chunks* pela rede. A redução do tráfego total da rede é um dos benefícios do baixo número de acertos no servidor, uma vez que menos pacotes precisaram percorrer todo o caminho entre o consumidor e o provedor.

Figura 4.4, o RankCache também se mostrou superior em relação a essa métrica. Ele foi o algoritmo que menos atingiu o servidor nas simulações, menor até que o *ProbCache*, o qual foi desenhado para otimizar essa métrica (Psaras b, 2012). Com 900 Mbytes de *cache*, o RankCache foi 17% superior em relação ao *ProbCache*, segundo mais eficiente.

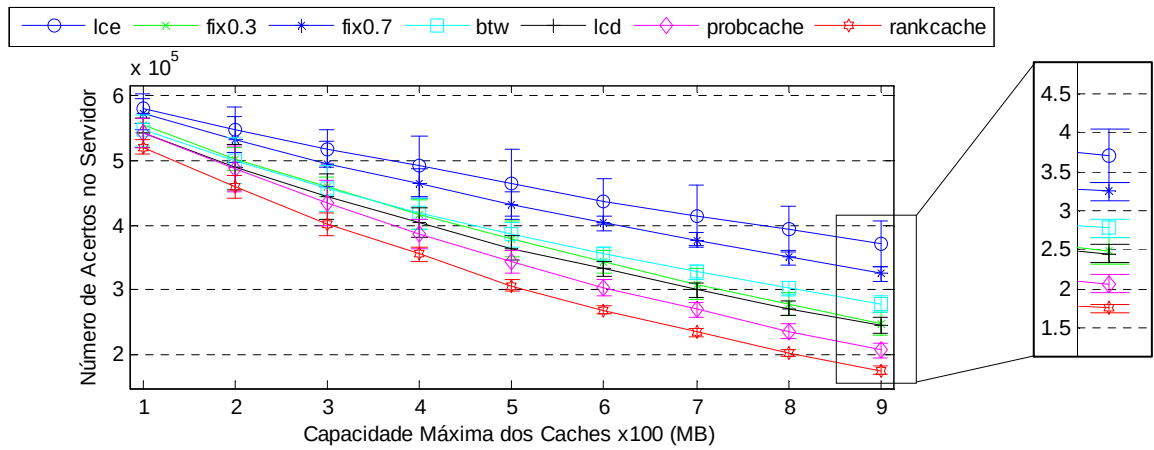


Figura 4.4 Número de Acertos no servidor em relação à capacidade dos caches variando de 100 MBytes a 900 MBytes.

O atraso dos pacotes é outra métrica importante que reflete diretamente no QoS (*Quality of Service*) e o QoE (*Quality of Experience*) das aplicações dos usuários finais. Um baixo atraso é requisito mínimo de serviços de tempo real como *streaming* de áudio e vídeo. A avaliação mostrou que o RankCache possui o menor atraso médio total. A Figura 4.5 mostra o tempo de ida e volta (RTT - *Round Trip Time*) das mensagens CCN na rede. Como o RankCache provê uma alta taxa de acerto, o conteúdo é entregue o mais rápido possível, o que reduz o atraso total dos pacotes. O ganho, neste caso, é de 10% em relação ao segundo melhor, *ProbCache*.

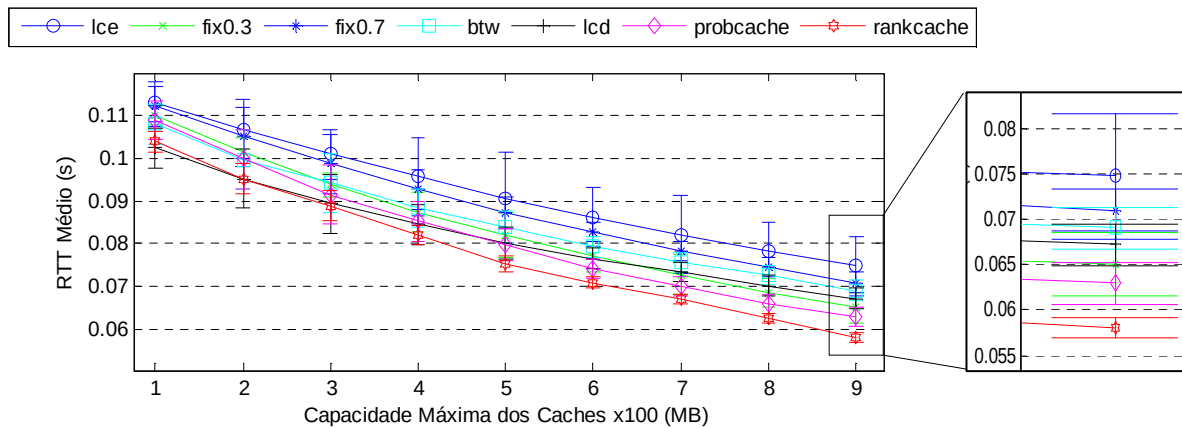


Figura 4.5 RTT médio em relação à capacidade dos caches variando de 100 MBytes a 900 MBytes

Na distribuição Zipf, um alto valor do expoente alfa (α) representa um tamanho pequeno de conteúdos com uma popularidade similar. Por outro lado, um valor baixo de α representa um grande número de conteúdos com a mesma popularidade. A Figura 4.6 mostra o número de acertos no servidor em relação à variação do expoente Zipf entre 0,5 e 1,0. O RankCache tem maior vantagem quando α é pequeno (maior concentração de conteúdos populares), uma vez que suas características levam em conta a popularidade dos *chunks*. O RankCache

tem o melhor desempenho, em torno de 17% com $\alpha = 0,5$ e 9% com $\alpha = 1,0$, em relação ao segundo melhor, Probcache.

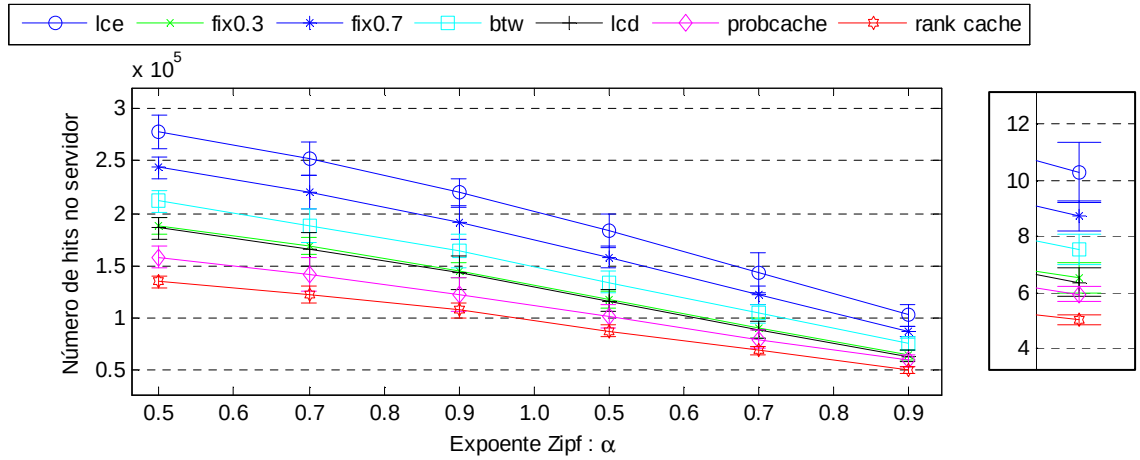


Figura 4.6 Número de acertos no servidor em relação ao expoente Zipf α , onde α varia de 5×10^{-1} a 10×10^{-1} . Tamanho do cache igual a 900MB

Os resultados que mostram o RTT médio de todos os pacotes em relação ao expoente do Zipf são apresentados na Figura 4.7. O ganho do RankCache foi de 10% comparado com o Fix03, segundo menor RTT médio.

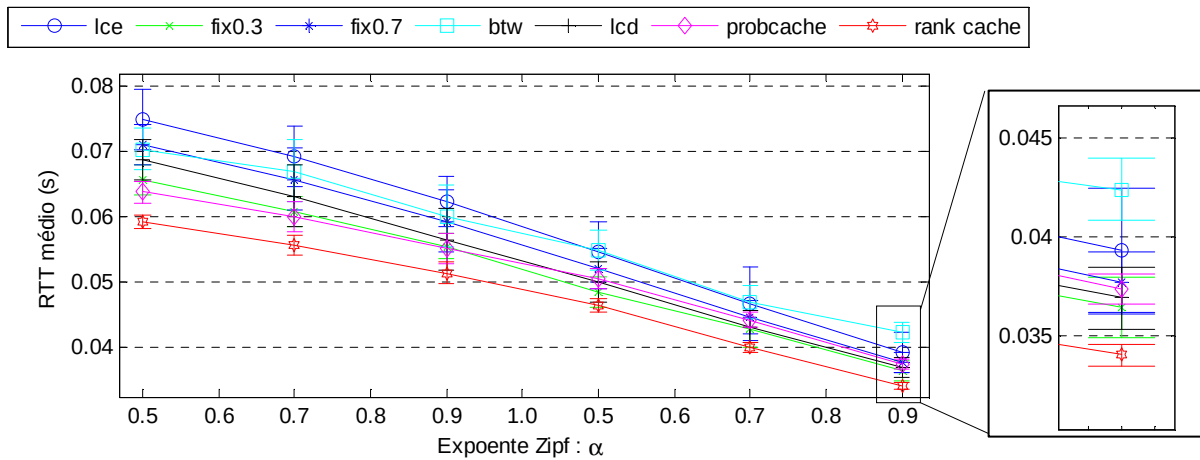


Figura 4.7 RTT médio em relação ao expoente Zipf α , onde α varia de 5×10^{-1} a 10×10^{-1} . Tamanho do cache igual a 900MB

Para verificar a natureza distribuída dos algoritmos testados, foi introduzido um modelo de erro de *bits* igual a 1×10^{-6} , uniformemente distribuído. Para essa simulação foram realizadas 15 repetições. A Figura 4.8 mostra os resultados do número médio de pacotes perdidos na rede. Com um modelo de erro aplicado na rede, perdas de pacotes são inevitáveis. Quanto maior o caminho percorrido, maior a probabilidade de o conteúdo ser perdido por erros na transmissão. O RankCache apresenta menor número de pacotes perdidos e isso se deve ao fato dele distribuir os *chunks* de forma mais eficiente na rede, de modo que as requisições não precisam percorrer longos caminhos

para obter o conteúdo. Neste caso, o ganho do RankCache foi de 33% se comparado com o LCD, segundo melhor.

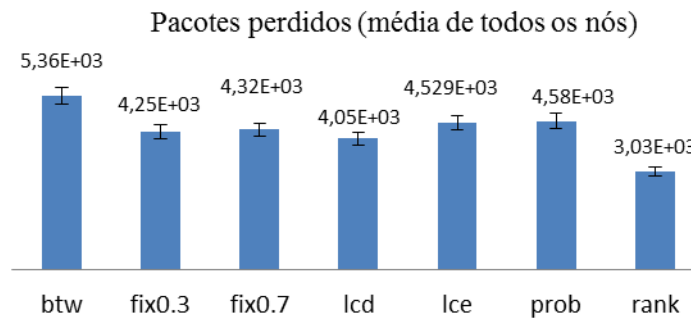


Figura 4.8 Valor médio dos pacotes perdidos, em 15 simulações, com modelo de erro de bit igual a 1×10^{-6}

4.4. Considerações Finais

Neste capítulo, foi apresentado um novo esquema de armazenamento em nós da rede chamado RankCache. A proposta cria um *ranking* de *caches* no caminho do fluxo de pacotes. Os *caches* com maior *ranking* possuem a maior probabilidade de armazenar um determinado pacote. Os *rankings* são calculados por meio do intervalo de confiança da proporção binomial, usando a equação de Wilson, onde o valor de interesse é o número de vezes que um conteúdo foi encontrado nos *caches* intermediários. A proposta foi avaliada e comparada com várias outras da literatura como *Leave Copy Everywhere*, *Fix*, *Betweenness Centrality*, *Leave Copy Down* e *ProbCache*. Em todos os casos, o RankCache se mostrou superior nas métricas avaliadas, com ganhos que variam de 10% a 30%, em relação ao segundo melhor de cada métrica.

5

MOBILIDADE DE PROVEDOR EM REDES CCN

Este capítulo apresenta a solução proposta para melhorar a qualidade de serviço em cenários com mobilidade de provedor. É proposto o envio de uma mensagem de Interesse especial que atualiza as rotas após a mobilidade do provedor. O esquema é avaliado usando simulação e os resultados mostram seus benefícios em relação ao CCN padrão. Esse capítulo tem o foco apenas na mobilidade. No capítulo seguinte, a correlação entre mobilidade e *caching* será apresentada e, no Capítulo 7, será descrita uma proposta completa com sinergia entre *caching* e mobilidade.

5.1. Proposta de Mobilidade de Provedor

O estudo dos trabalhos relacionados apresentados no Capítulo 3 permitiu chegar à conclusão de que a metodologia do grupo AD-FIB requer menor tempo para atualizar a rota entre o consumidor e o provedor, após a mobilidade. Conforme detalhado anteriormente, essa solução utiliza o envio de uma mensagem especial na rede para atualizar a FIB diretamente. Normalmente, essas mensagens saem da rede atual para a rede anterior. A simplicidade é outra vantagem das estratégias AD-FIB, pois a maioria das estratégias desse grupo não precisa de grandes modificações na rede para funcionar, algumas acrescentam poucos campos nos pacotes CCN, porém, a maioria das soluções dos outros grupos também modifica as mensagens CCN.

Embora apresentem algumas ideias interessantes, a maior parte das propostas AD-FIB não contempla todos os requisitos para a mobilidade nas redes de nova geração, como transparência, baixa latência de *handover*, independência de topologia, baixo *overhead* de mensagens e reconfiguração rápida dos caminhos. Por este motivo, nesta seção propõe-se um novo método para mobilidade de provedor que contempla todos os requisitos apresentados acima. A ideia principal da proposta é utilizar os mesmos benefícios obtidos pela mobilidade de consumidor: quando um consumidor migrar de uma rede para

outra, a rota até o provedor já deveria estar estabelecida e otimizada. Pretende-se usar essa mesma ideia na solução da mobilidade de provedor por meio da utilização de pacote especial chamado de *Interesse Virtual* (IV).

Existem soluções, como em Kim (2012) e Kim (2014), que propõem uma mensagem de Interesse Especial enviada do provedor após a mobilidade. Porém, naqueles trabalhos, essa mensagem é enviada em direção ao ponto de acesso antigo e tem o objetivo de criar um túnel entre o antigo e o novo ponto de acesso ao qual o provedor se conectou. A mensagem de Interesse Virtual proposta nesta tese é encaminhada em direção ao consumidor e tem o objetivo de atualizar a rota entre o provedor e o consumidor. O Interesse Virtual pode interceptar o caminho antigo e a atualização será muito mais rápida. No pior caso, para atualizar toda a rota, a mensagem chega até a rede próxima ao consumidor. O encaminhamento do Interesse Virtual até o consumidor só é possível devido à Tabela de Sessão apresentada a seguir.

A Figura 5.1 mostra o cenário básico da proposta de mobilidade de provedor. Na proposta, o provedor possui uma Tabela de Sessão que armazena os conteúdos que estão sendo requisitados naquele momento. A tabela roda como uma aplicação e armazena apenas dois valores, o índice de cada linha e o pacote de Interesse que representa a sessão ativa. Esse pacote difere da mensagem de Interesse convencional por armazenar informações do caminho entre provedor e consumidor. Esse pacote é transmitido apenas uma vez do consumidor ao provedor, no início da comunicação entre eles. As informações de índice e nome do pacote de Interesse são pequenas em termos de armazenamento e não geram *overhead* significativo no provedor.

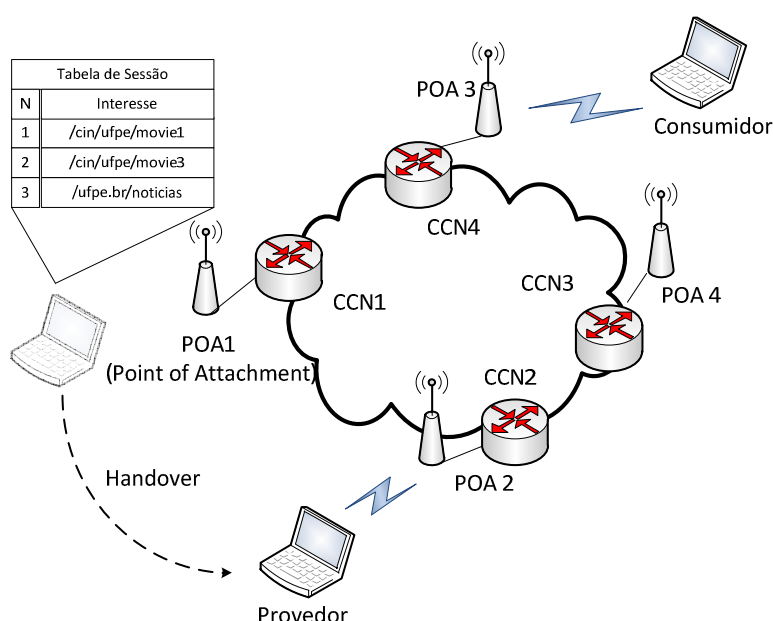


Figura 5.1 Cenário Base para explicação da proposta de mobilidade mostrando a Tabela de Sessão

A Figura 5.2 apresenta a sinalização completa do esquema de mobilidade de provedor. Essa figura toma como base o cenário destacado na Figura 5.1. Inicialmente, o consumidor está se comunicando com o Provedor pela rota PoA3/CCN4/CCN1/PoA1 (Figura 5.1).

- (A) O provedor monitora as sessões ativas, armazenando apenas um pacote de Interesse de cada sessão. Esse pacote possui todos os campos originais mais um campo com informações do caminho entre o provedor e o consumidor.
- [1] O consumidor envia um pacote de Interesse para requisitar um novo conteúdo do provedor. O pacote de Interesse segue o caminho até o provedor. Esse caminho já está configurado previamente por meio de alguns mecanismos de disseminação de prefixos. Nas simulações usaremos o OSPFN (Wang L. H., 2012) para esse propósito.
- [2] O Provedor recebe a requisição do conteúdo e envia o primeiro pacote de dados ao consumidor, informando tamanho do conteúdo, número do pacote e habilita uma *flag* de requisição de caminho. Essa *flag* possui 1 *bit* e informa à API dos roteadores que deverá marcar o próximo pacote de Interesse com o nome do roteador correspondente. Essa API do roteador é a mesma implementada no CCNx para monitorar pacotes de Interesse.
- [3] Quando o consumidor envia o segundo pacote de Interesse, os roteadores que estão com o monitoramento ativo marcam o nome do próprio roteador em cada nó. Assume-se que cada roteador possui um nome único na rede. Se o pacote passasse pelos roteadores CCN01, CCN03 e CCN04, nesta sequência, o campo de informações de rotas ficaria: “/CCN01” após o primeiro, “/CCN03/CCN01” após o segundo e “/CCN04/CCN03/CCN01” após o terceiro. Esse processo é realizado apenas uma vez durante cada sessão iniciada pelo consumidor. Todas essas informações são armazenadas na Tabela de Sessão dentro do Provedor. Cada entrada da tabela possui um tempo de vida curto que é atualizado com a chegada de novos pacotes de Interesse. Caso o tempo de vida expire, a entrada é removida. Essa operação ajuda a manter a tabela simples e pequena.
- [4] Os próximos pacotes de Interesse e de Dados trocados pelo consumidor e provedor seguem o padrão normal CCN.
- [5][6] Quando o provedor conecta-se à rede alvo, envia um pacote de Interesse Virtual (IV) para cada uma de suas sessões ativas. O nome do pacote de Interesse é formado pelas informações de rota obtidas na Tabela de Sessão (B).
- [7] À medida que atravessa os nós CCN, o pacote virtual atualiza as rotas nas FIBs com a nova localização dos conteúdos do provedor. O pacote é dito virtual porque não faz requisição a nenhum conteúdo e serve apenas para atualizar as rotas entre o provedor e consumidor.

- [8][9][10] Após a atualização das rotas, a comunicação entre provedor e consumidor segue normalmente.

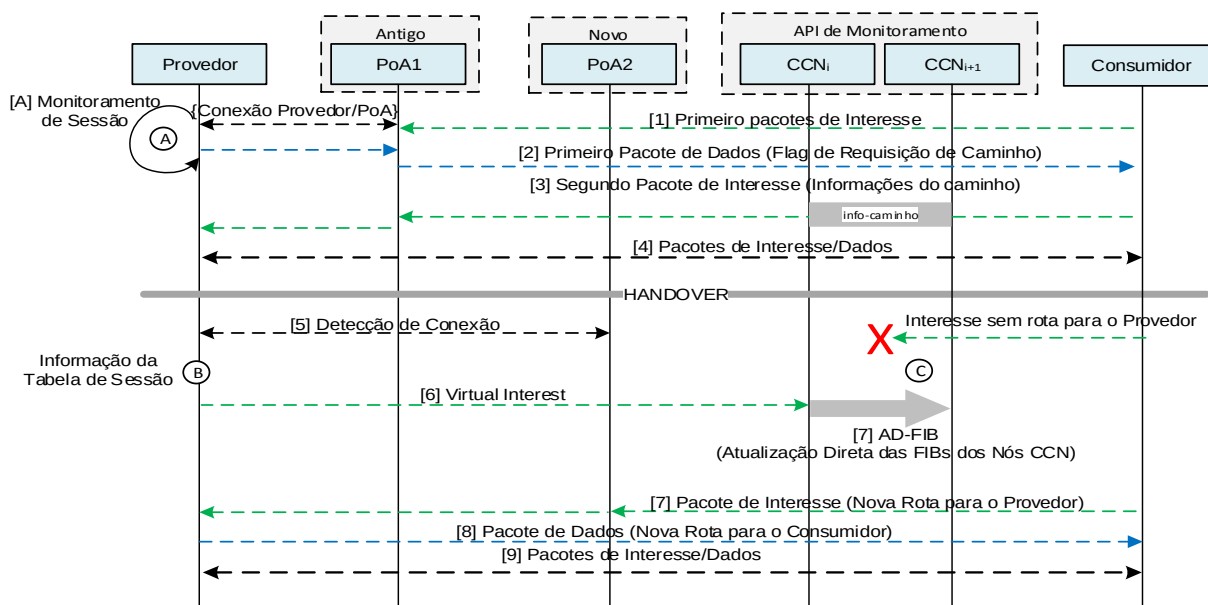


Figura 5.2 Sinalização da Proposta de Mobilidade de Provedor

O pacote de Interesse Virtual (IV) difere do Interesse convencional por armazenar informações do caminho até o consumidor. A Figura 5.3 ilustra como esse pacote é formado. A identificação de cada roteador atravessado pelo pacote de Interesse é acrescentada ao pacote como ilustrado nos passos 1.1, 1.2 e 1.3 da Figura 5.3. Para evitar *overhead*, esse acréscimo acontece apenas no primeiro pacote de cada sessão que se inicia. O *cache* do Provedor armazena essa informação na Tabela de Sessão. Após a migração para outra rede, o provedor envia um pacote IV (passo 3) usando as informações da tabela de sessão. O IV atualiza as informações de todas as FIBs pelas quais ele passa, para refletir a localização atual do provedor (passo 4). O novo caminho pode ter interseção com o caminho anterior, Rota-01, ou encontrar outra rota melhor, Rota-02. Dessa forma, a atualização das FIB é feita dinamicamente, sem túneis e sem depender da topologia da rede.

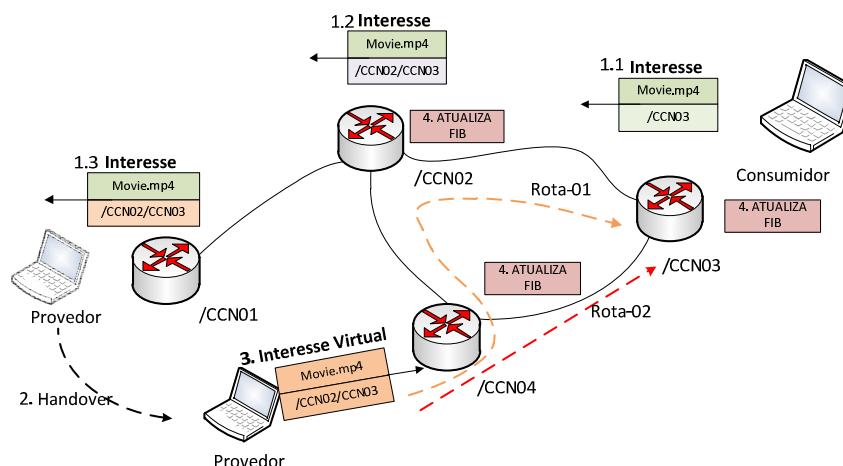


Figura 5.3 Interesse Virtual

5.2. Avaliação

A proposta foi avaliada por meio de simulações. A rede usada nas simulações foi semelhante a da Figura 5.1. O simulador foi o mesmo apresentado na Seção 4.3. No entanto, foi necessário acrescentar ao simulador as funcionalidades relacionadas à solução proposta para mobilidade de provedor. A Tabela 5.1 resume os parâmetros utilizados nas simulações.

Tabela 5.1 Parâmetros usados nas simulações

Parâmetros	Valores
Área de Cobertura	800m x 800m
Atrasos na rede.	10-100ms (Uniformemente Distribuído)
Quantidade Máxima de Pacotes no <i>Cache</i>	200MB
Tamanho dos pacotes	2048 KB
Taxa de Transmissão	Exponencial: média = 1/10, 1/20, 1/50, 1/80
Tamanho do catálogo ⁸	10 ⁵ arquivos
Número de Repetições	20 (cada cenário)
Variações de Velocidade	1m/s, 10m/s, 20m/s, 50m/s
Tempo de simulação	600s
Movimento do provedor	RandomWPMobility
Posição inicial do provedor	Aleatório Uniformemente Distribuído.

A Figura 5.4 mostra a vazão média (em Mbps) em cada uma das 20 simulações. Nesse caso, a velocidade do provedor é de 1 m/s. Nessa velocidade, com movimento aleatório do provedor, a probabilidade de ocorrer *handover* é menor. Por isso, pode-se observar dois momentos de queda na vazão da Figura 5.4, na simulação 4 e 5 e nas simulações 14,15 e 16. Para comparação justa, cada par de simulações (com e sem a Proposta) foi executada usando a mesma semente de aleatoriedade, isso permite que seja observado o mesmo comportamento em ambas as propostas.

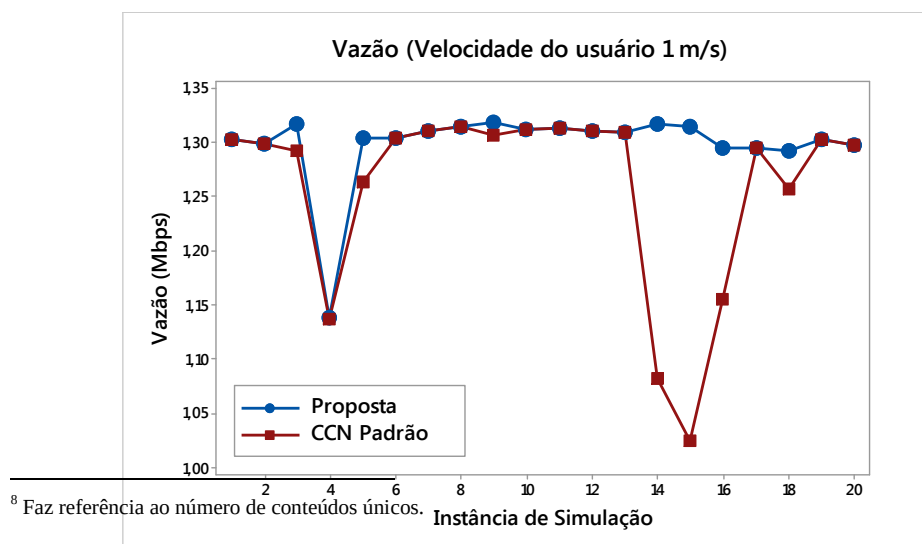


Figura 5.4 Vazão com e sem a proposta com velocidade do usuário a 1m/s

Com o usuário se movimentando a 10 metros por segundo, a probabilidade de *handover* é maior que a anterior. Com isso, ocorrem muitos *handovers*. Sem a proposta, há uma queda acentuada na vazão média das simulações, conforme é mostrado na Figura 5.5. Com a proposta, o caminho é atualizado rapidamente e não há muitas perdas, o que mantém a vazão no nível alto.

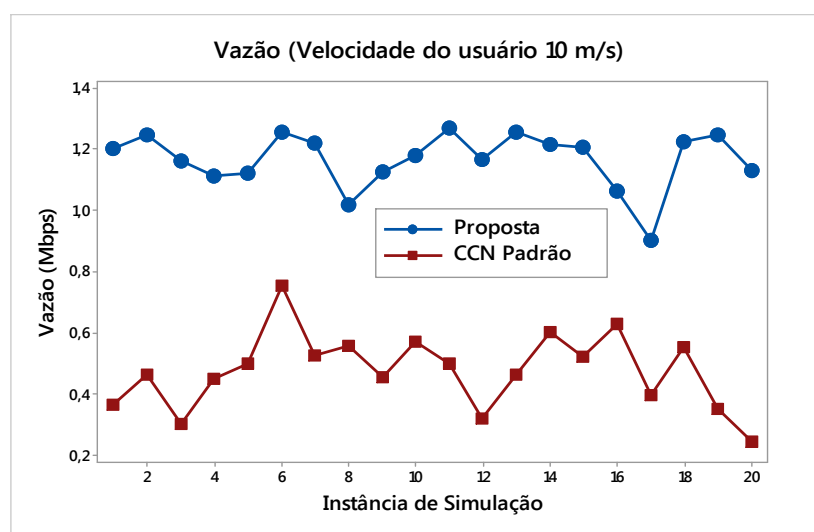


Figura 5.5 Vazão com e sem a proposta com velocidade do usuário a 10m/s

Com velocidade de 30 metros por segundo, a proposta também se apresenta superior ao CCN padrão. A Figura 5.6 é semelhante à Figura 5.5, no entanto, há uma queda evidente na média de ambas as técnicas. Isso ocorre porque com 30 m/s (ou 108km/h) o usuário desconecta-se com muita frequência e isso aumenta a probabilidade de perda de pacote.

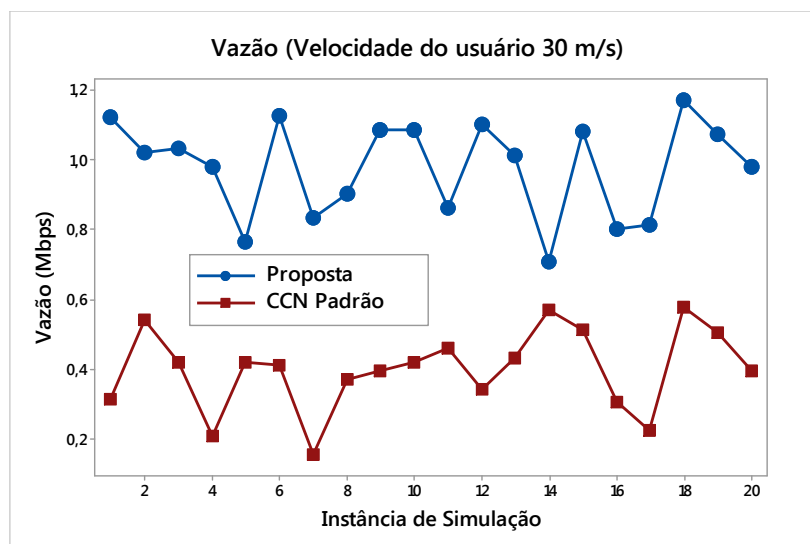


Figura 5.6 Vazão com e sem a proposta com velocidade do usuário a 30m/s

A Figura 5.7 apresenta a média, com 95% de intervalo de confiança, das 20 simulações feitas para cada velocidade. Percebe-se que com 1m/s não é possível dizer estatisticamente que a proposta é superior ao CCN, pois existem elementos em ambas as estratégias com o mesmo valor. Porém, a partir de 10m/s a estratégia proposta é superior ao CCN padrão. A partir de 10m/s, a média da vazão é de 0,97Mbps e sem a proposta fica em torno de 0,4 Mbps.

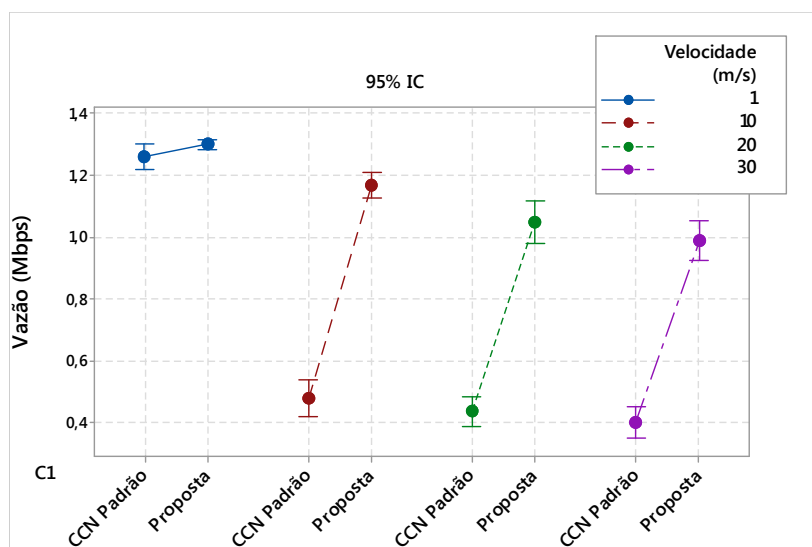


Figura 5.7 Vazão média calculada das 20 repetições para cada estratégia e velocidade.

A Tabela 5.2 apresenta a estatística descritiva de todos os cenários simulados com 95% de intervalo de confiança. A Tabela mostra que, com a proposta, há uma tendência de queda na média da vazão com o aumento da velocidade do usuário. Mostra também a superioridade da solução proposta, quando comparada ao CCN Padrão. O ganho da proposta com o provedor se deslocando a 30m/s é de 142%.

Tabela 5.2 Estatística descritiva dos resultados da avaliação

Proposta (1m/s)	1,298	0,00862	0,0385	1,1379	1,2972	1,3064	1,3134	1,3183
Proposta (10m/s)	1,1663	0,0205	0,0917	0,9045	1,1238	1,1901	1,2392	1,2706
Proposta (20m/s)	1,0533	0,0319	0,1427	0,544	1,041	1,0834	1,1374	1,2092
Proposta (30m/s)	0,9787	0,0307	0,1372	0,7093	0,8413	1,016	1,0863	1,1708
CCN (1m/s)	1,259	0,0194	0,0866	1,0236	1,2579	1,3005	1,3102	1,3137
CCN (10m/s)	0,4784	0,0276	0,1235	0,2452	0,3741	0,4841	0,5582	0,7575
CCN (20m/s)	0,4328	0,022	0,0986	0,2061	0,3881	0,4242	0,5105	0,6124

CCN (30m/s)	0,4004	0,026	0,1164	0,1577	0,324	0,4152	0,4942	0,5807
-------------	--------	-------	--------	--------	-------	--------	--------	--------

A Figura 5.8 mostra as médias de perda de pacotes conforme variação da velocidade do provedor. É possível notar que a partir de 10 m/s são perdidos muitos pacotes com o CCN padrão. Isso acontece porque quanto maior a velocidade do provedor, maior é o número de desconexões e, portanto, maiores são as perdas durante a sessão ativa. Percebe-se que mesmo com a proposta há perda de pacotes, pois muitos dos pacotes de Interesse enviados pelo consumidor são perdidos durante a mobilidade do provedor. Uma proposta para esse problema será apresentada no Capítulo 7.

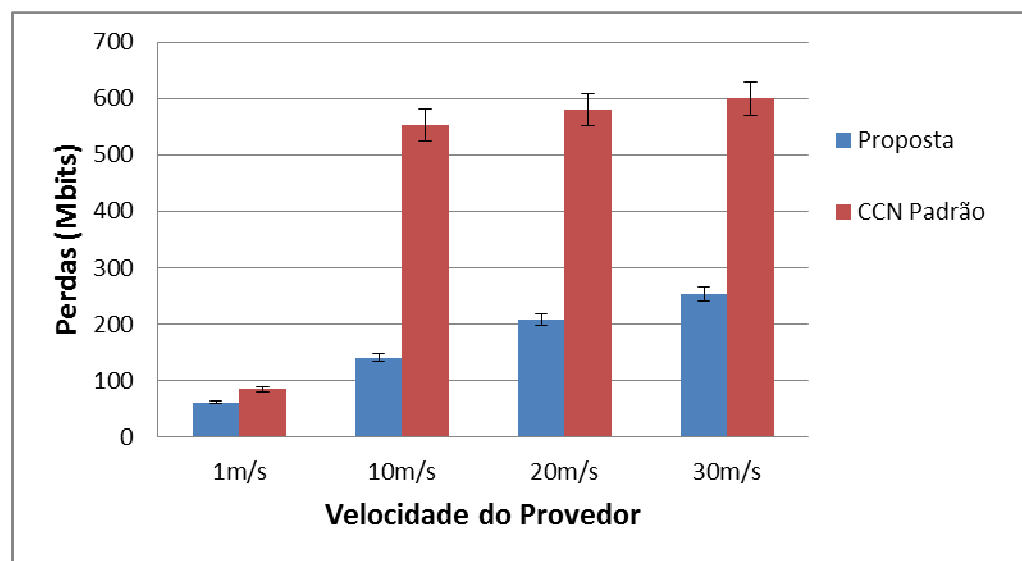


Figura 5.8 Gráfico de perda de pacotes variando a velocidade de provedor.

5.3. Considerações Finais

Este capítulo apresentou uma proposta para o problema de mobilidade de provedor. O esquema utiliza o envio de uma mensagem de Interesse Virtual que atualiza as rotas após a mobilidade do provedor. A proposta é avaliada usando simulação e os resultados mostram ganhos que podem chegar a 142% com o provedor em alta velocidade (30m/s). Os resultados mostraram também que quanto maior for a velocidade do provedor, maior será a ocorrência de *handovers*, podendo causar a diminuição da vazão, uma vez que gera o aumento da perda de pacotes devido às desconexões. Por este motivo, é importante que haja um mecanismo de prevenção ou recuperação dessas perdas. Esse mecanismo foi elaborado e integrado à proposta principal desta tese e será abordado no Capítulo 7.

6

MOBILIDADE E *CACHE*

O objetivo deste capítulo é apresentar a combinação das soluções discutidas nos capítulos 4 e 5 e demonstrar, estatisticamente, que o *caching* pode influenciar no desempenho da mobilidade e vice-versa. Para tal, foi aplicado o teste T-Emparelhado usando-se as mesmas estratégias de *caching* apresentadas no Capítulo 4. Os resultados mostram que a mobilidade pode ser influenciada usando-se estratégias de *cache* diferentes e, portanto, faz sentido propor soluções conjuntas.

6.1. Introdução

O armazenamento de *chunks* nos nós CCN é um desafio do núcleo da rede, enquanto que a mobilidade de provedor é uma preocupação da borda da rede. Geralmente, esses dois desafios são atacados de forma separada como se não estivessem correlacionados. Mas os experimentos apresentados nesta seção mostram que eles estão intimamente relacionados e que, para fornecer serviço de qualidade a usuários móveis nas redes CCN, ambos os problemas precisam ser resolvidos conjuntamente.

No experimento desta seção foram utilizadas as mesmas propostas explicadas na seção 3.1, são elas: *ProbCache*, LCE, LCD, Probabilidade Fixa (30% e 70%) e *Betweenness Centrality*. Além desses esquemas, foi utilizado o RankCache, que é a solução proposta por este trabalho para armazenamento eficiente de *chunks*.

A Figura 6.1 mostra o cenário de simulação utilizado para o estudo da mobilidade de provedor e armazenamento de *chunk*. Foram utilizados 20 usuários móveis, 27 nós CCN e 23 Pontos de Acesso. A área total de simulação é de 1000m x 1000m. Os pontos de acesso foram colocados de modo que todo o espaço fosse preenchido e não houvesse quebra de conexão do usuário devido à existência de buracos na rede.

6.2. Resultados

Como não é do nosso conhecimento que existam trabalhos na literatura que analisem a integração de cenários de mobilidade de provedor e armazenamento de *chunks* na rede, antes de propor novas metodologias para esse problema é preciso entender se as estratégias de *caching* melhoram o desempenho de serviços mesmo em ambientes com mobilidade e se a mobilidade influencia estratégias de *caching*. Demonstrar essa correlação entre mobilidade e *caching* é de vital importância para a consolidação desta tese.

Para fazer essa análise, simulou-se o cenário descrito acima e extraiu-se a métrica Taxa de Acerto (do inglês, *HitRate*). Como explicado no Capítulo 4, quando um pacote de Interesse passa por um *cache* e o pacote de Dados correspondente encontra-se previamente armazenado, o conteúdo é encontrado e o pacote de Interesse não precisa ser encaminhado até o provedor. Nesse caso aconteceu um acerto ou *hit*. Quando um pacote de Interesse passa por um *cache* e o pacote de Dados não está armazenado, o primeiro é encaminhado para o próximo *cache* no caminho até o provedor. Nesse caso ocorreu um erro ou *miss*. A taxa de acerto é calculada por meio da relação entre número de acertos por número de erros. Quanto maior for a taxa, mais efetivo será o algoritmo de armazenamento.

A Figura 6.2 mostra a média das taxas de *hits* em todos os nós da rede. Percebe-se na Figura que a média do RankCache é maior que as demais. No entanto, é possível observar que existem pontos coincidentes entre o RankCache e as demais propostas, com exceção do LCE.

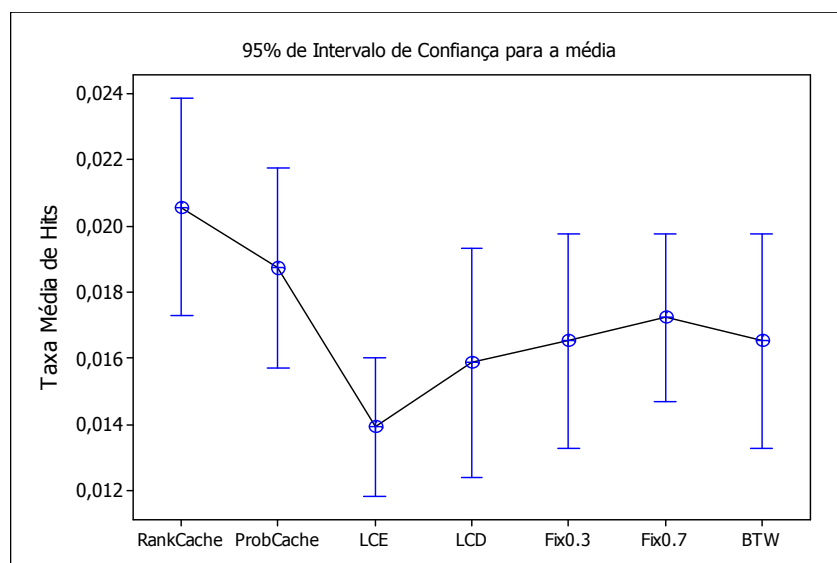


Figura 6.2 Taxa média de Hits

Para verificarmos se há ou não diferença estatística, optou-se por fazer o teste T-Emparelhado (Hsu, 2008), o qual é bastante usado em comparações de dados pareados. Para dados com menos de 30 repetições, o teste T exige que as amostras tenham uma distribuição normal. Para atestar a normalidade das amostras, usou-se o teste Kolmogorov-Smirnov (Lilliefors, 1967). Esse teste é bastante usado na literatura para verificar se duas distribuições diferem uma da outra ou se uma amostra finita pode ser explicada usando-se uma determinada distribuição. Para verificar se as amostras seguem uma distribuição normal, as seguintes hipóteses são levantadas:

- H_0 : A distribuição normal pode ser utilizada para prever o comportamento dos dados observados.
- H_A : A distribuição normal não pode ser utilizada para prever o comportamento dos dados observados.

No teste Kolmogorov-Smirnov (ou apenas KS), existem duas estatísticas a serem calculadas, o D Crítico e o D Teste. Se o D Teste for inferior ao D Crítico, a Hipótese Nula é aceita, caso contrário, a Hipótese Nula é refutada. Em outras palavras, adotando-se um nível de significância de 5 %, ou Intervalo de Confiança de 95%, se o valor P (*P-Value*) for inferior a 5% refuta H_0 .

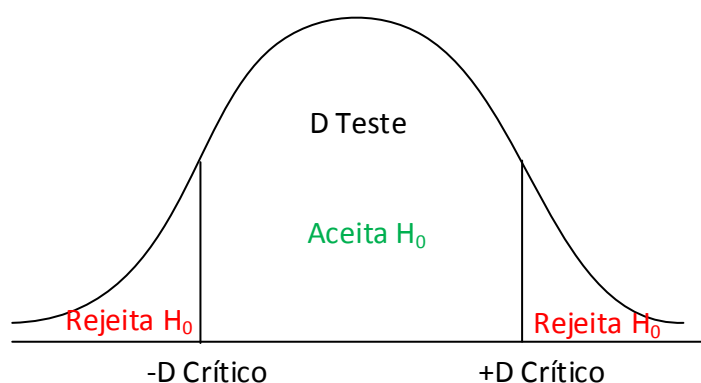


Figura 6.3 Área de Aceitação para o teste Kolmogorov-Smirnov

A estatística D crítico positivo e D crítico negativo são dadas pelas equações Eq. 7 e Eq. 8.

$$D+ = \max |F(x_{(i)}) - Fn(x_{(i)})| \quad \text{Eq. 7}$$

$$D- = \max |F(x_{(i)}) - Fn(x_{(i-1)})| \quad \text{Eq. 8}$$

Onde $F(x)$ é dada por:

$$F(x) = P\left(z_i \leq \frac{x_{(i)} - x''}{s}\right) \quad \text{Eq. 9}$$

As estatísticas D^+ e D^- medem a distância vertical entre os gráficos das duas funções, teórica e empírica, nos pontos $x_{(i-1)}$ e $x_{(i)}$. Com isso, a estatística de teste Kolmogorov-Smirnov pode ser obtida através da equação Eq. 10.

$$D_n = \max(D^+, D^-) \quad \text{Eq. 10}$$

Se D_n for maior que o D crítico, então a hipótese nula pode ser rejeitada. Caso contrário, não podemos rejeitar essa hipótese. A Tabela 6.2 mostra os valores críticos para a estatística de Kolmogorov-Smirnov para um número N repetições e um valor α de Significância. Nas simulações, usou-se 20 repetições e com 95% de confiança (ou 0,05 de significância) o D crítico, segundo a tabela, é de 0,29. Para H_0 ser aceito, e a amostra ser considerada como uma distribuição normal, o D_n precisa ser melhor que 0,29.

Tabela 6.2 D Crítico do teste Komolgorov-Smirnov

	Nível de Significância α			
N	0,2	0,1	0,05	0,01
5	0,45	0,51	0,56	0,67
10	0,32	0,37	0,41	0,49
15	0,27	0,30	0,34	0,40
20	0,23	0,26	0,29	0,36
25	0,21	0,24	0,27	0,32
30	0,19	0,22	0,24	0,29
35	0,18	0,20	0,23	0,27
40	0,17	0,19	0,21	0,25
45	0,16	0,18	0,20	0,24
50	0,15	0,17	0,19	0,23
Valores maiores	$\frac{1,07}{\sqrt{n}}$	$\frac{1,22}{\sqrt{n}}$	$\frac{1,36}{\sqrt{n}}$	$\frac{1,63}{\sqrt{n}}$

O gráfico da Figura 6.4 mostra a distribuição dos valores da amostra em torno da reta que representa a distribuição normal. Quanto mais próxima à reta, maior a probabilidade da amostra seguir uma distribuição normal.

Figura 6.4, é possível ver o valor KS. Esse valor é o mesmo D_n explicado anteriormente, portanto, como $KS = 0,20$ é menor que $D_n = 0,29$, então a hipótese nula é aceita e os dados observados para o RankCache seguem uma distribuição normal.

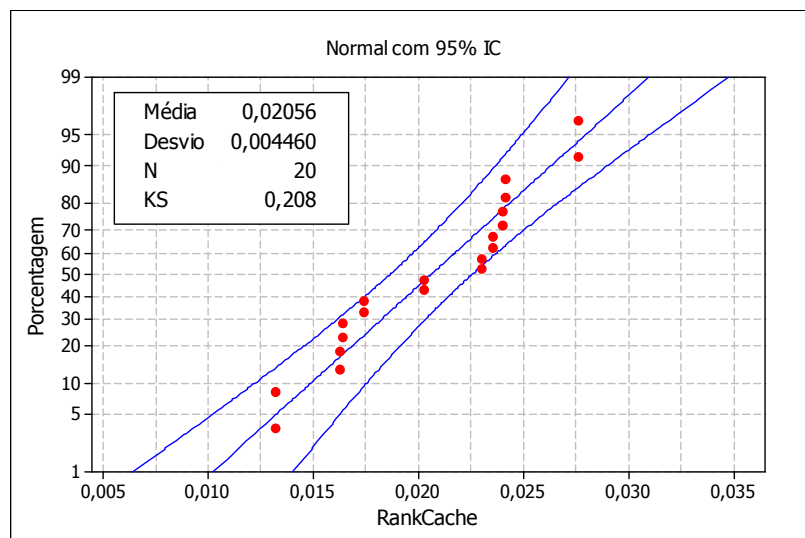


Figura 6.4 Gráfico de Probabilidade do Teste KS para o RankCache

Figura 6.5 exibe os resultados do teste de aderência à distribuição normal para a proposta *ProbCache*. Como o valor de KS (0,14) é menor que 0,29, podemos dizer com 95% de confiança que os dados seguem uma distribuição normal. As linhas laterais delimitam o intervalo de confiança.

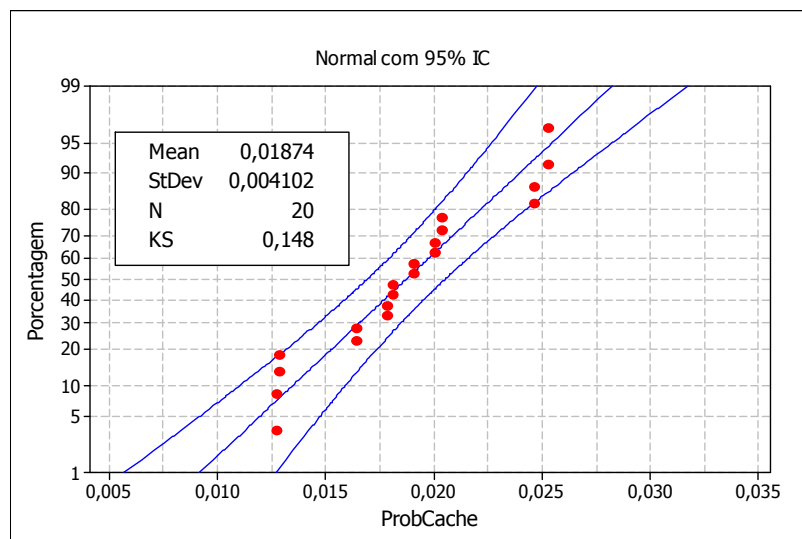


Figura 6.5 Gráfico de Probabilidade do Teste KS para o *ProbCache*

A Figura 6.6 apresenta os resultados do teste para a proposta LCE. Nesse caso, o valor de KS é 0,02 e também é menor que 0,29, portanto, pode-se dizer com 95% de confiança que os dados também seguem uma distribuição normal.

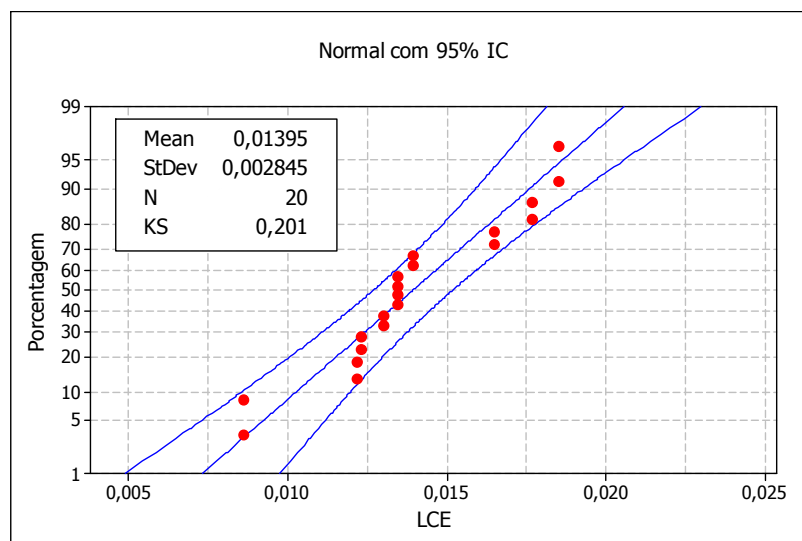


Figura 6.6 Gráfico de Probabilidade do Teste KS para o LCE

Para a proposta LCD, os resultados mostrados pela Figura 6.7 também confirmam que os dados seguem uma distribuição normal, com 95% de confiança. KS é igual a 0,22, portanto menor que 0,29.

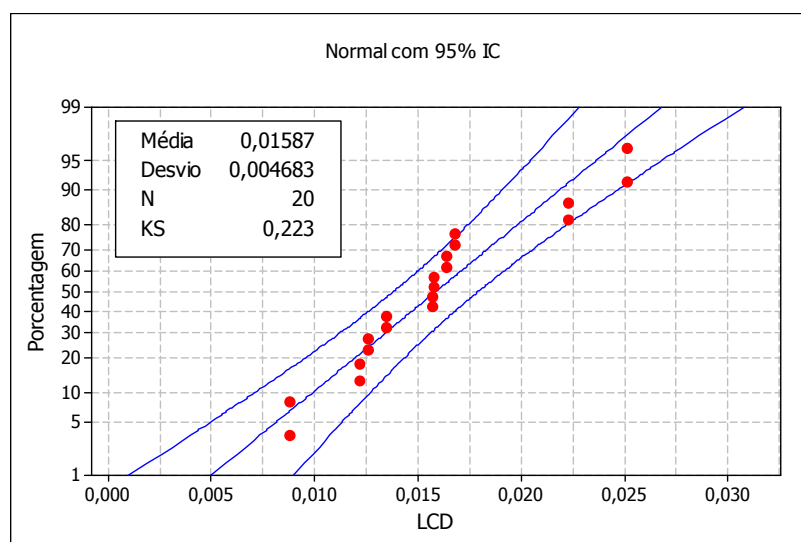


Figura 6.7 Gráfico de Probabilidade do Teste KS para o LCD

O mesmo pode ser observado pela Figura 6.8. KS é igual a 0,16 e, portanto, a hipótese nula é aceita e os dados podem ser representados por uma distribuição normal.

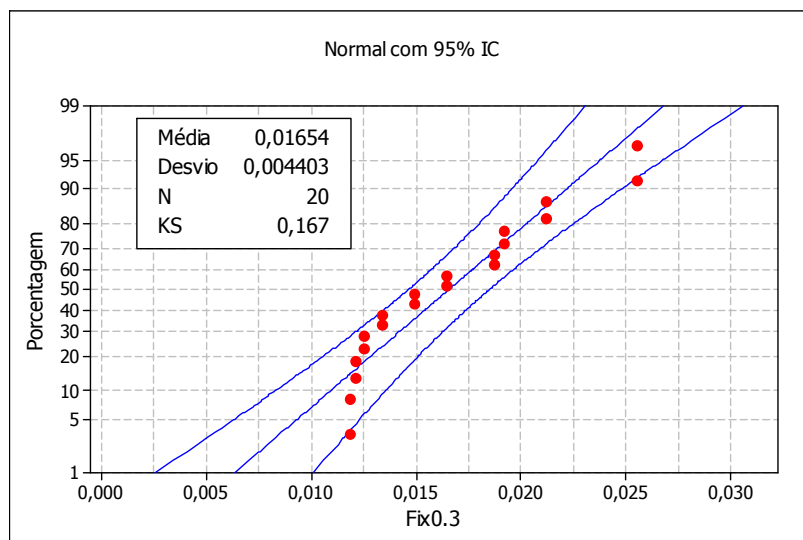


Figura 6.8 Gráfico de Probabilidade do Teste KS para o Fix-0.3

Na Figura 6.9, o KS é 0,157, e a distribuição normal também pode explicar os dados para a proposta de probabilidade fixada em 70%.

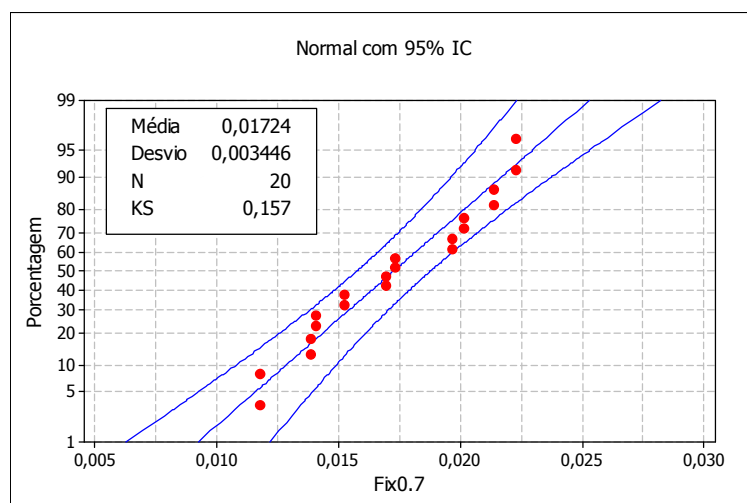


Figura 6.9 Gráfico de Probabilidade do Teste KS para o Fix-0.7

Por último, os resultados do teste de aderência da proposta Betweenness Centrality são apresentados pela Figura 39 e também confirma a hipótese nula.

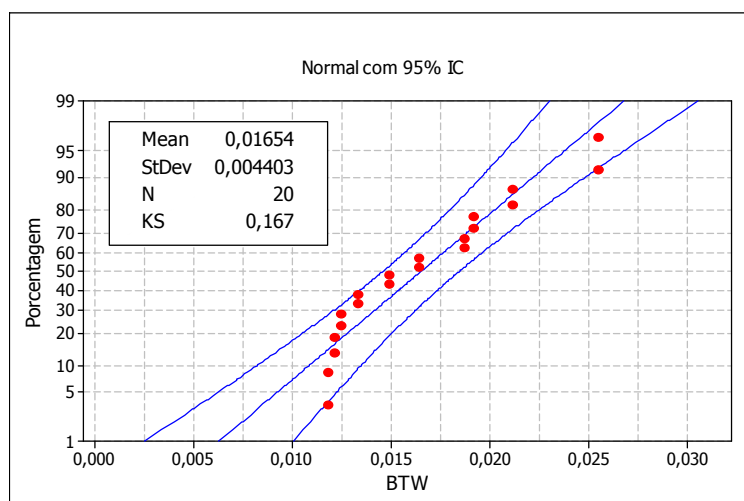
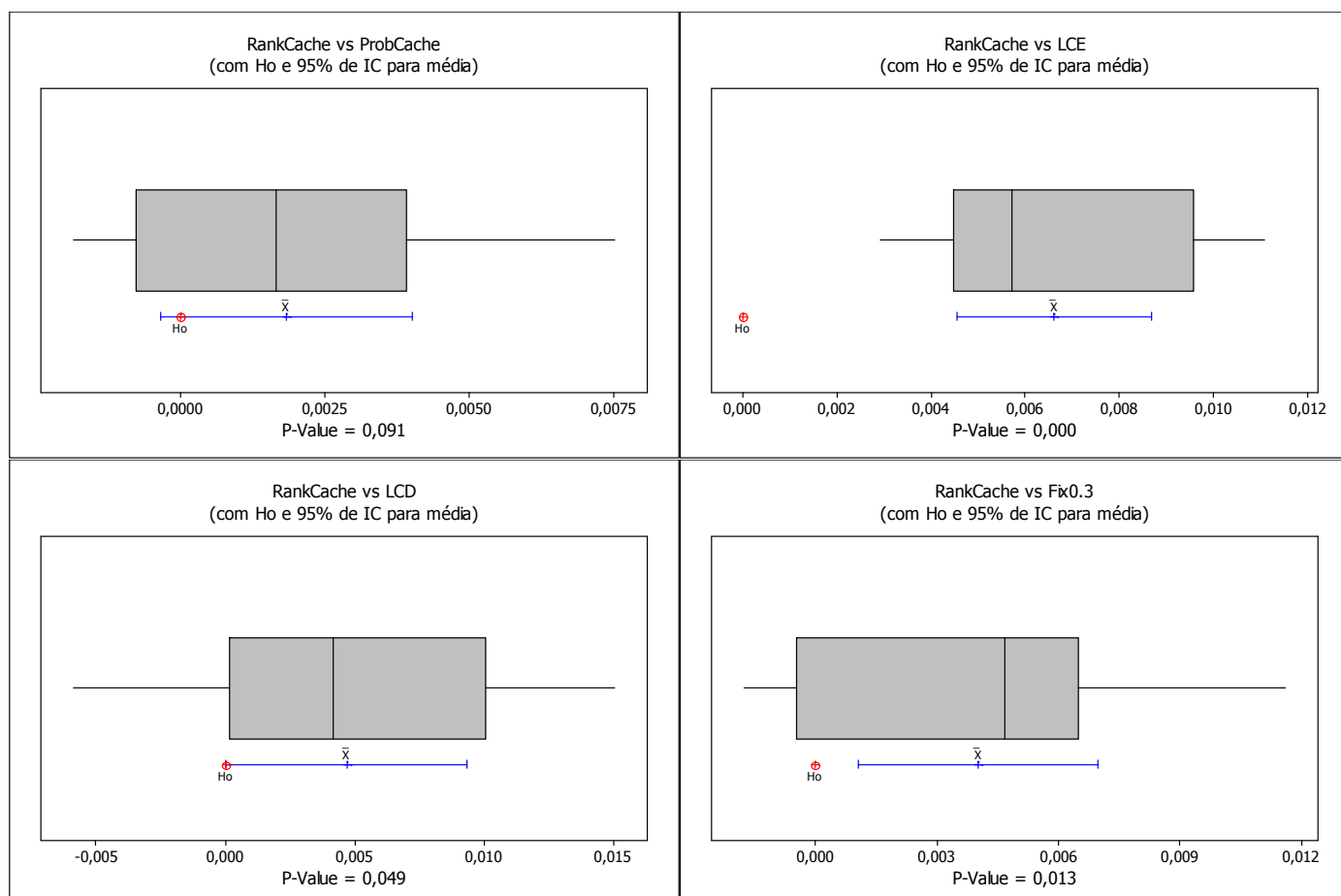


Figura 6.10 Gráfico de Probabilidade do Teste KS para o Betweenness Centrality

Observando os resultados de todos os testes Kolmogorov-Smirnov de aderência à distribuição normal, pode-se dizer que os dados seguem uma distribuição normal e, portanto, é possível usar o teste T-Emparelhado para comparar as amostras. Os experimentos de comparação foram conduzidos levando-se em consideração as seguintes hipóteses:

- H_0 : Diferentes estratégias de *caching* não influenciam no desempenho dos serviços em cenários com mobilidade.
- H_a : Diferentes estratégias de *caching* influenciam no desempenho dos serviços em cenários com mobilidade.

A hipótese nula é o comportamento natural do sistema, ou seja, que as novas estratégias de *caching* não influenciam em um ambiente móvel. Em termos estatísticos, isso é o mesmo que dizer que as médias entre parâmetros de desempenho das propostas analisadas são iguais. A Figura 40 apresenta os resultados do teste T-Emparelhado. O resultado está sendo mostrado com gráficos de *boxplot* e estatística *p-value* (ou valor-p). Em suma, se o *p-value* for menor que o nível de significância adotado, a hipótese nula será rejeitada. Em todos os experimentos foram adotados o nível de significância de 5% (95% de Intervalo de Confiança). Isso quer dizer que se o *p-value* for menor que 0.05, a hipótese nula será refutada, caso contrário, ela não pode ser rejeitada.



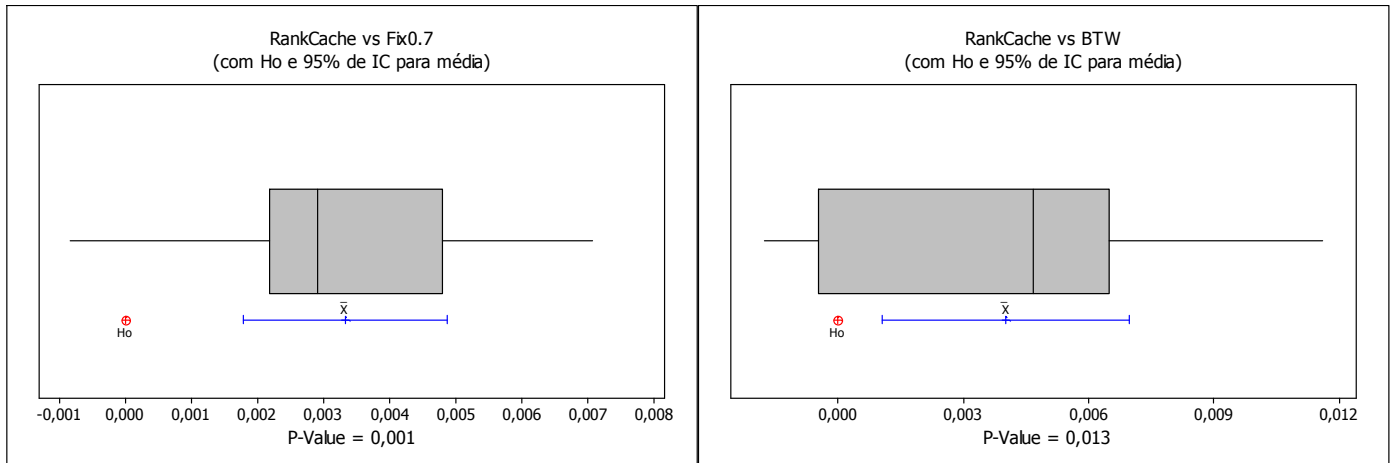


Figura 6.11 Teste T-Emparelhado entre o RankCache e as outras propostas

Para analisar a hipótese nula, o teste T foi conduzido entre o RankCache e as demais propostas. Se há algum ganho entre as propostas, as médias serão estatisticamente diferentes. A comparação entre o RankCache e *ProbCache*, o p-value foi de 0,091, sendo assim, não é possível refutar a hipótese nula nesse caso. No entanto, em todas as outras comparações, o p-value é melhor que 0,05. E nesse caso, como é possível ver nos *BoxPlots* da Figura 40, o ponto em H0 está relativamente distante da média estimada $\bar{X}$. Portanto, H0 foi refutada em 5 de 6 experimentos, dando uma probabilidade de 0,83 de negação da hipótese nula. Deste modo, temos fortes indícios que a hipótese alternativa é mais correta, ou seja, diferentes estratégias de *caching* influenciam no desempenho dos serviços para a métrica taxa de *hits*.

O próximo passo é saber se há diferenças de desempenho entre as mesmas métricas em cenários com mobilidade e sem mobilidade. Para isso, as seguintes hipóteses são levantadas:

- H_0 : O desempenho de estratégias de *caching* é igual em cenários com e sem mobilidade.
- H_a : O desempenho de estratégias de *caching* é diferente em cenários com e sem mobilidade.

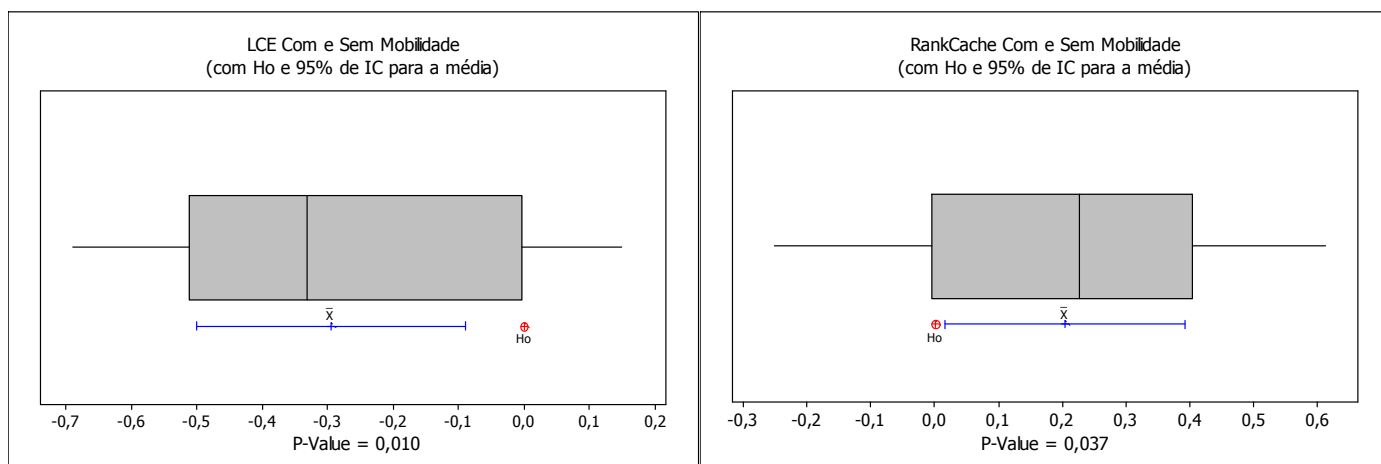
Para determinar se a hipótese nula é aceita, foram realizadas simulações pareadas com e sem mobilidade para cada métrica, usando-se o mesmo cenário e parâmetros apresentados na Figura 6.1 e Tabela 6.1. A métrica que será usada para avaliar a hipótese nula será a vazão média total de todos os clientes da rede. Tomando essa métrica como referência aplicou-se o teste T-Emparelhado para uma mesma métrica em ambos os cenários, com e sem mobilidade. Como foi dito anteriormente, para aplicar o teste T, as amostras precisam seguir a distribuição normal. Desta forma, aplicou-se o teste de aderência Kolmogorov-Smirnov.

A Tabela 6.3 resume o resultado do teste de aderência para as propostas. Para 5% de nível de significância, se aceita a hipótese de que as amostras seguem uma distribuição normal caso a estatística de D Teste ou KS seja menor que 0,29. Percebe-se na tabela que quase todas são menores que 0,29, com exceção da estratégia “Fix0.7 sem”, que é de 0,453. Por esse motivo, ela foi retirada do teste de hipóteses.

Tabela 6.3 Resumo do Teste Kolmogorov-Smirnov

Proposta	KS	< 0,29
LCE Sem	0,264	Sim
LCE Com	0,180	Sim
RankCache Com	0,125	Sim
RankCache Sem	0,183	Sim
ProbCache Sem	0,205	Sim
ProbCache Com	0,186	Sim
BTW Sem	0,247	Sim
BTW Com	0,099	Sim
LCD Sem	0,28	Sim
LCD Com	0,206	Sim
Fix0.3 Sem	0,106	Sim
Fix0.3 Com	0,150	Sim
Fix0.7 Sem	0,453	Não
Fix0.7 Com	0,143	Sim

A Figura 41 mostra o resultado do teste T-Emparelhado das estratégias de armazenamento. Em todos os casos, o *p-value* é menor que 0.05, o que nos faz refutar a hipótese nula. Portanto, pode-se dizer com 95% de certeza que o desempenho de estratégias de *caching* é diferente em cenários com e sem mobilidade, ou seja, a mobilidade afeta o desempenho das soluções de *caching*.



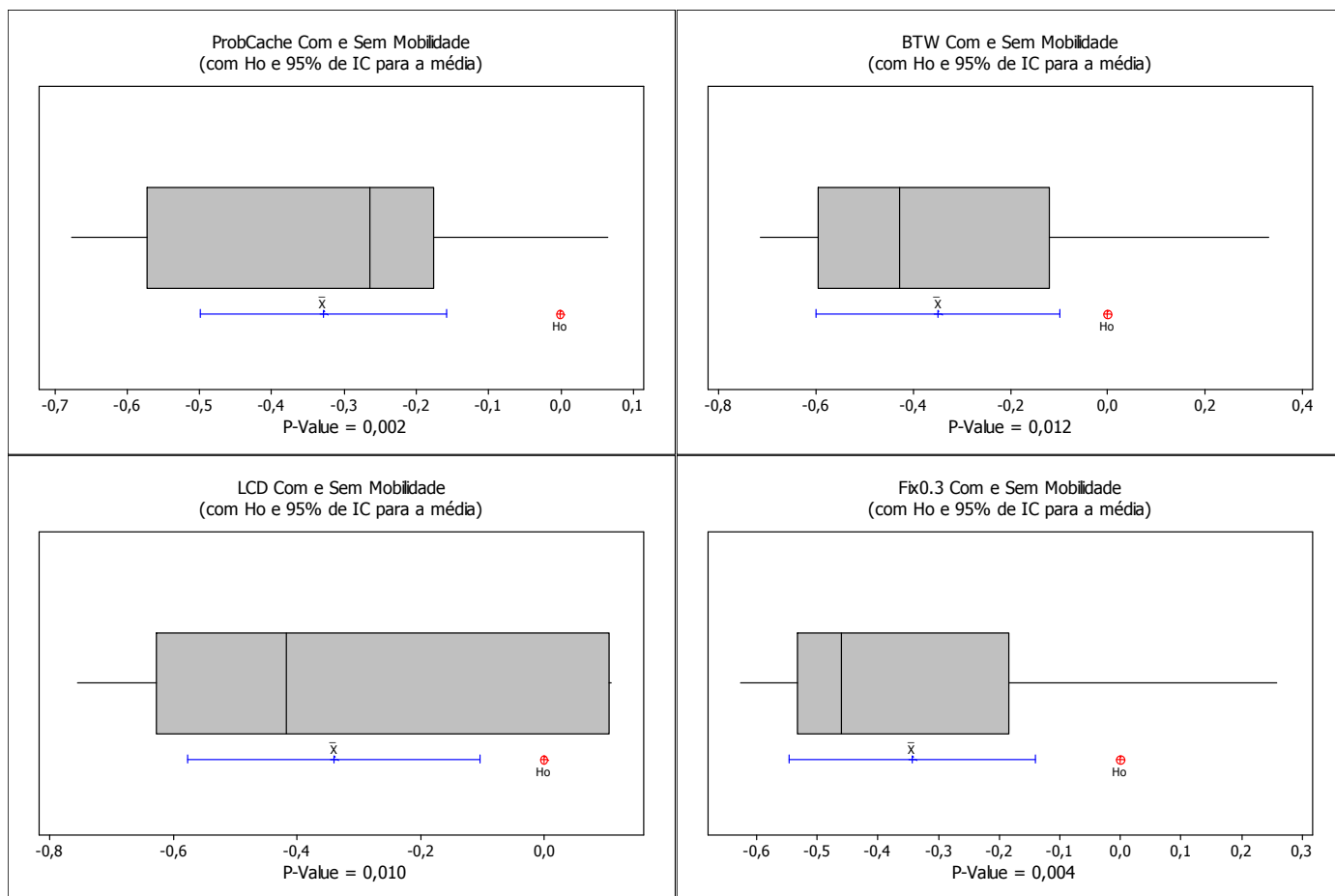


Figura 6.12 Resultado do teste T-Emparelhado das estratégias com e sem mobilidade

Após a análise dos resultados, pode-se dizer estatisticamente que há fortes evidências de que a mobilidade está correlacionada com as estratégias de *cache* e vice-versa, isso porque mostrou-se que a mobilidade de provedor influencia nas estratégias de *caching* e as mesmas podem alterar valor de desempenho dos serviços em ambientes móveis. A

Figura 6.13 apresenta o desempenho das estratégias em relação à vazão no cenário com mobilidade. A estratégia RankCache, em sua concepção, não foi pensada para ser superior em ambientes móveis, apenas cabeado. Mesmo assim, observa-se certa superioridade. É possível que a modificação de certos parâmetros e adequação do modelo utilizado possa melhorar ainda mais o desempenho da proposta RankCache em ambientes móveis.

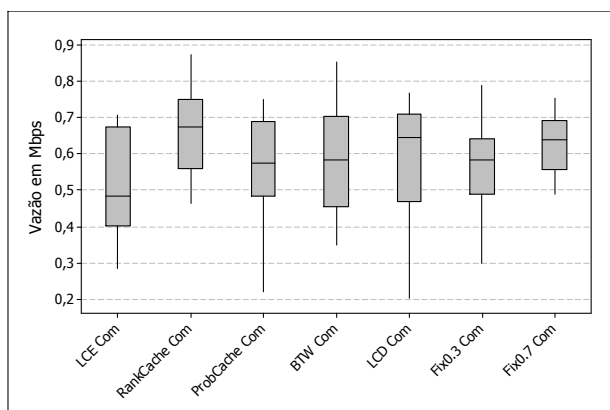


Figura 6.13 Boxplot da vazão das estratégias de *caching* com mobilidade

A Figura 6.14 mostra a vazão média para cada estratégia no cenário sem mobilidade de provedor. O RankCache também é superior, porém com uma folga maior, pois esse cenário, apesar de sem fio, como não há mobilidade, assemelha-se aos cenários totalmente cabeados, e na Seção 4.3, já foi mostrada a superioridade do RankCache.

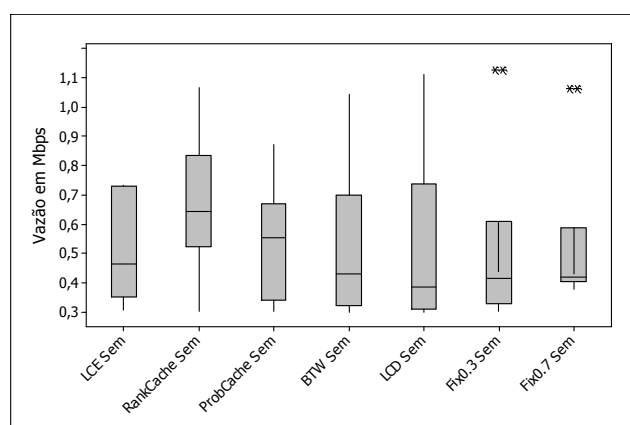


Figura 6.14 Boxplot da vazão das estratégias de *caching* sem mobilidade

6.3. Considerações Finais

Este capítulo demonstrou estatisticamente que há correlação entre mobilidade e *caching*. Um ambiente móvel pode afetar estratégias de *caching* e diferentes estratégias de *caching* possuem desempenhos diferentes. Os testes realizados em cenários de redes móveis utilizando o algoritmo RankCache evidenciaram melhores resultados que os apresentados por esquemas similares para o propósito de *caching* na rede cabeada, presentes na literatura. Isso mostra que o problema em questão é relevante e que faz sentido propor uma solução para integrar mobilidade de provedor com *caching* na rede. O próximo capítulo apresentará a solução proposta na tese para o problema de mobilidade de provedor, tratando mobilidade e *caching* de modo integrado.

7

MOBCACHE

Este capítulo apresenta a integração entre a solução para mobilidade de provedor, o ranqueamento de *cache* e uma estratégia para redução de perdas de pacotes durante a mobilidade de provedor. A essa solução deu-se o nome de MobCache. O capítulo mostra, através de avaliação, que a proposta apresenta bons resultados em comparação com outras propostas da literatura.

7.1. Introdução

No capítulo anterior, demonstrou-se que diferentes algoritmos de armazenamento de *chunks* possuem diferentes desempenhos em ambientes com alto grau de mobilidade. Isso quer dizer que é possível criar soluções que melhorem o desempenho de serviços usando *caching* em ambientes móveis. Como nenhum dos algoritmos usados nos experimentos foi concebido pensando em ambientes móveis, é possível que novas estratégias de *caching* com foco na mobilidade consigam um desempenho bem superior. Não é do nosso conhecimento que existam propostas na literatura que englobem os mundos sem fio e cabeados ao mesmo tempo.

O MobCache consiste em um arcabouço que une o RankCache, a proposta de mobilidade apresentada no Capítulo 5 e uma proposta de prevenção de perdas de pacotes apresentadas neste capítulo. Essa última consiste em o provedor enviar os pacotes de dados sem receber os pacotes de Interesse. Durante o *handover*, vários pacotes de Interesse são perdidos e, no CCN convencional, esses pacotes precisam ser reenviados pelo consumidor. No MobCache, o provedor calcula o número de pacotes de Interesse perdidos durante o *handover* e, ao conectar-se à outra rede, envia os pacotes de dados correspondentes.

A Figura 7.1 apresenta a sinalização completa do esquema de mobilidade de provedor com o MobCache. Essa figura toma como base o cenário destacado na Figura 7.2. Inicialmente, o consumidor está se comunicando com o Provedor pela rota “PoA3/CCN4/CCN1/PoA1”. Algumas mensagens são iguais às da

proposta de mobilidade de provedor, as mensagens [5] e [9] são acrescentadas à solução com a integração com o RankCache. O detalhamento de algumas mensagens será repetido para a explicação ser autocontida e facilitar o entendimento.

- (A) O provedor monitora as sessões ativas, armazenando apenas um pacote de Interesse de cada sessão.
- [1] O consumidor envia um pacote de Interesse para requisitar um novo conteúdo do provedor. O pacote de Interesse segue o caminho até o provedor, sendo que esse caminho já está configurado previamente através de alguns mecanismos de disseminação de prefixos. No simulador desenvolvido nesta tese, foi implementado o OSPFN (Wang L. H., 2012) para esse propósito.
- [2] O Provedor recebe a requisição do conteúdo e envia o primeiro pacote de dados ao consumidor, informando tamanho do conteúdo, número de pacote e habilita uma *flag* de requisição de caminho. Essa *flag* possui 1 bit e informa à API dos roteadores que o próximo pacote de Interesse deverá ser marcado com o nome do roteador correspondente. Essa API é a mesma implementada na aplicação real do CCN (CCNx.org) para monitorar pacotes de Interesse.
- [3] Quando o consumidor envia o segundo pacote de Interesse, os roteadores que estão com o monitoramento ativo marcam o seu nome no pacote de Interesse. Assume-se que cada roteador possui um nome único na rede. Se o pacote passasse pelos roteadores CCN01, CCN03 e CCN04, nesta sequência, o campo de informações de rotas ficaria: “/CCN01” após o primeiro, “/CCN03/CCN01” após o segundo e “/CCN04/CCN03/CCN01” após o terceiro. Esse processo é realizado apenas uma vez durante cada sessão iniciada pelo consumidor. Todas essas informações são armazenadas na Tabela de Sessão dentro do Provedor. Cada entrada da tabela possui um tempo de vida curto que é atualizado com a chegada de novos pacotes de Interesse. Caso o tempo de vida expire, a entrada é removida. Essa operação ajuda a manter a tabela simples e pequena.
- [4] Os próximos pacotes de Interesse e de Dados trocados pelo consumidor e provedor seguem o padrão normal CCN.
- (B) No Evento B, o Provedor detecta que está prestes a mover-se para outra rede. Essa detecção pode ser feita por meio do monitoramento de níveis de sinal ou usando alguma técnica de predição.
- [5] Após detectar o *handover* iminente, o provedor calcula o número de pacotes de Dados que poderiam ter sido requisitados durante o processo de mobilidade. Esse cálculo é feito para todas as sessões ativas da Tabela de Sessões.
- (C) No período de ocorrência do *handover*, os pacotes de Interesse enviados para o provedor são perdidos.

- [6] [7] Quando o provedor conecta-se à rede alvo, o novo PoA envia os pacotes de Interesse armazenados para o provedor. Além disso, o provedor envia um pacote de Interesse Virtual na direção dos consumidores, que também é enviado para todas as sessões ativas do provedor. O nome do pacote é formado pelas informações de rota obtidas na Tabela de Sessão (D).
- [8] O pacote virtual atualiza as rotas nas FIB pelas quais atravessa para a nova localização do provedor. O pacote é dito virtual, porque não faz requisição a nenhum conteúdo e serve apenas para atualizar as rotas entre o provedor e consumidor.
- [9] Nesse passo, são enviados os pacotes de Dados calculados no passo [5]. Esses pacotes foram nomeados como “Dados Gratuitos” porque não foram requisitados através de um *Interest*. Por meio dessa estratégia, os pacotes de dados que, possivelmente, foram perdidos durante o *handover* são entregues ao consumidor. Isso é feito de forma transparente para o consumidor.
- [11][12] Após a atualização das rotas, a comunicação entre provedor e consumidor segue normalmente.

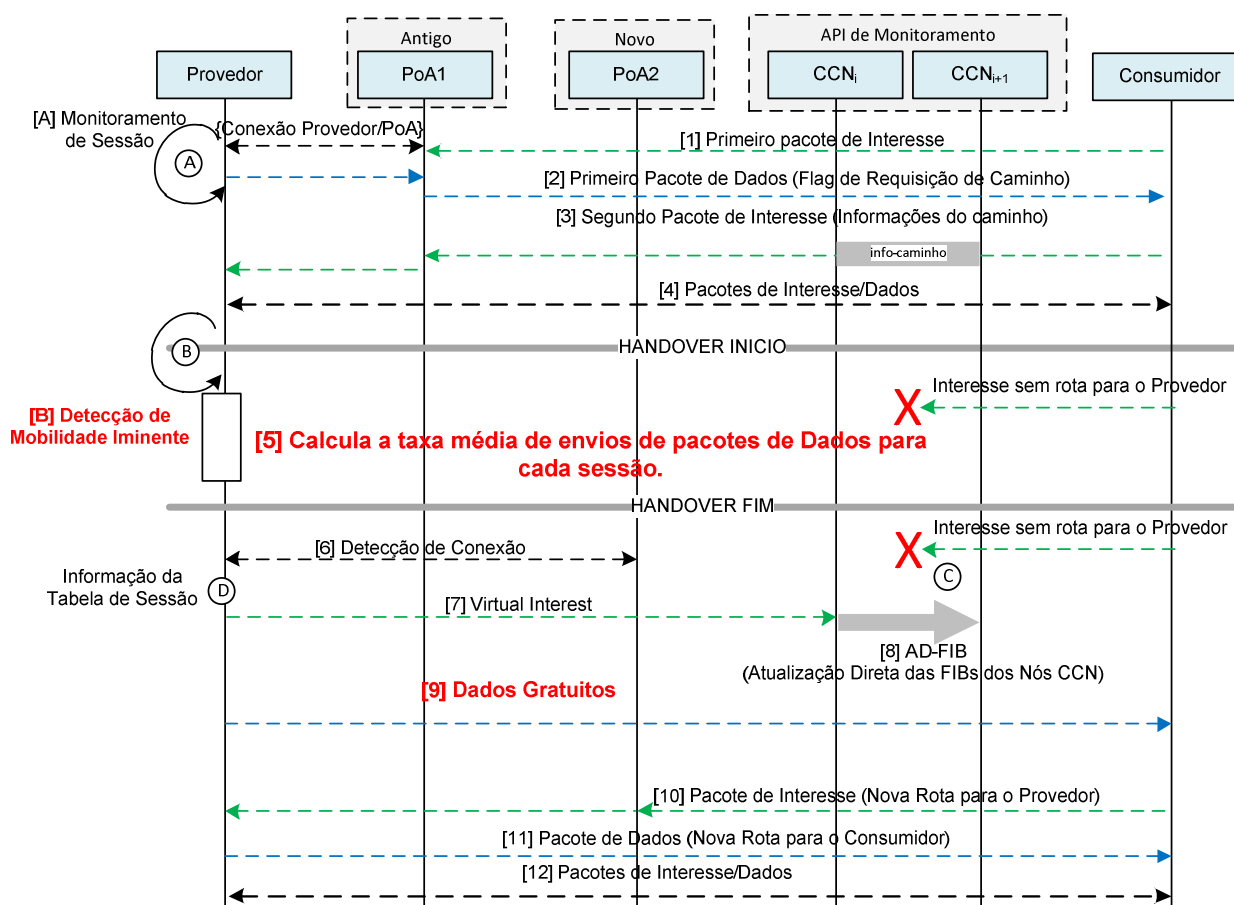


Figura 7.1 Sinalização do MobCache

No Capítulo 5 mostrou-se que cada *cache* possui uma Tabela de Sessão que armazena os conteúdos que estão sendo requisitados naquele momento. Para implementar a recuperação de pacotes no MobCache, essa tabela foi estendida com dois campos extras. Além de armazenar o índice de cada linha e o pacote de Interesse que representa a sessão ativa, a nova tabela armazena o número do último pacote de dados enviado antes do *handover* (UP – Último Pacote) e a taxa média de saída de dados (λ). A Eq. 11 foi elaborada para calcular o número de pacotes de Dados Gratuitos a serem enviados após o *handover* do provedor na *i*-ésima sessão.

$$DG(i) = L\lambda_i - E$$

Onde:

$DG(i)$: Número de pacotes de Dados Gratuitos na *i*-ésima sessão

L : O tempo que o provedor leva para reconectar na outra rede.

λ_i : Taxa média de transmissão dos pacotes de Dados do provedor para a *i*-ésima sessão.

E : Margem de erro.

Para o provedor obter o valor de $DG(i)$, precisa monitorar o tempo de duração de *handover* e verificar a taxa de saída de pacotes de dados. O cálculo resultante é o número de pacotes de Interesse perdidos durante o *handover*. Como esse é um valor aproximado, é possível haver discrepâncias entre o número real de pacotes de Interesse que realmente foram enviados pelo consumidor e o valor estimado. Por esse motivo, acrescentou-se à fórmula uma margem de erro (E) que será reduzida do número de pacotes totais que precisam ser transmitidos. Após finalizar o *handover* e obter o valor de $DG(i)$, o provedor verifica o último pacote enviado (UP), Figura 7.2, para transmitir ao consumidor os pacotes de Dados Gratuitos. Na próxima seção será apresentada a avaliação do algoritmo proposto.

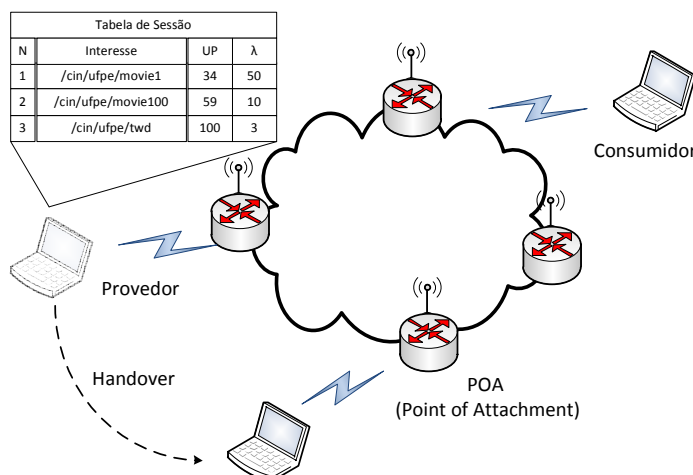


Figura 7.2 Nova Tabela de Sessão (estendida da Figura 5.1) com informações de UP e λ

7.2. Avaliação de Desempenho

O desempenho do MobCache foi analisado por meio de simulação e modelagem analítica. O MobCache foi avaliado através de simulação para capturar características e métricas mais realistas, tanto usando QoS (Qualidade de Serviço) quanto com QoE (Qualidade de Experiência). Os resultados do MobCache foram comparados com alguns trabalhos da literatura como o MobiNDN (Zhou, 2014), PBMA (Rao Y. G., 2014a), PNPCCN (Woo, 2014) e PCNDN (Rao Y. Z., 2013). A proposta também foi avaliada através de modelagem analítica. Foi criado um modelo de custo do *handover* e os resultados do MobCache foram comparado com duas outras propostas, o PCNDN (Rao Y. Z., 2013) e CCN padrão.

7.2.1. Avaliação por meio de simulação

A avaliação por simulação é ideal para analisar algumas métricas de redes, como perdas de pacotes, atrasos, vazão. Por este motivo, o MobCache foi avaliado também através de simulação. Foram implementadas no simulador as estratégias mencionadas anteriormente, MobiNDN, PBMA, PNPCCN e PCNDN. Os três primeiros são representantes dos grupos E-C, I-MIP e AD-FIB respectivamente e o PCNDN é a mesma estratégia usada, como comparação, na modelagem. Essa última não atualiza a rota entre consumidor e provedor, ela foi utilizada na comparação por apresentar uma estratégias de recuperação do conteúdo perdido durante o *handover*. Essas estratégias foram escolhidas, em cada grupo, por seus bons desempenhos em relação aos parâmetros Latência e Custo de Sinalização apresentados na Tabela 3.3 do Capítulo 3.

7.2.1.1. Avaliação de QoS (Quality of Service).

Para as simulações foram utilizados os mesmos parâmetros da Tabela 5.1 mostrados no Capítulo 5. A Figura 7.3 mostra a vazão entre todas as propostas. O MobCache está em destaque, com linha contínua. É possível notar, visualmente, que na maior parte das simulações o MobCache consegue obter uma ganho maior que as outras propostas, sendo também a proposta mais estável. A maioria oscila muito: em uma simulação está com bom desempenho; em outra, com baixo desempenho.

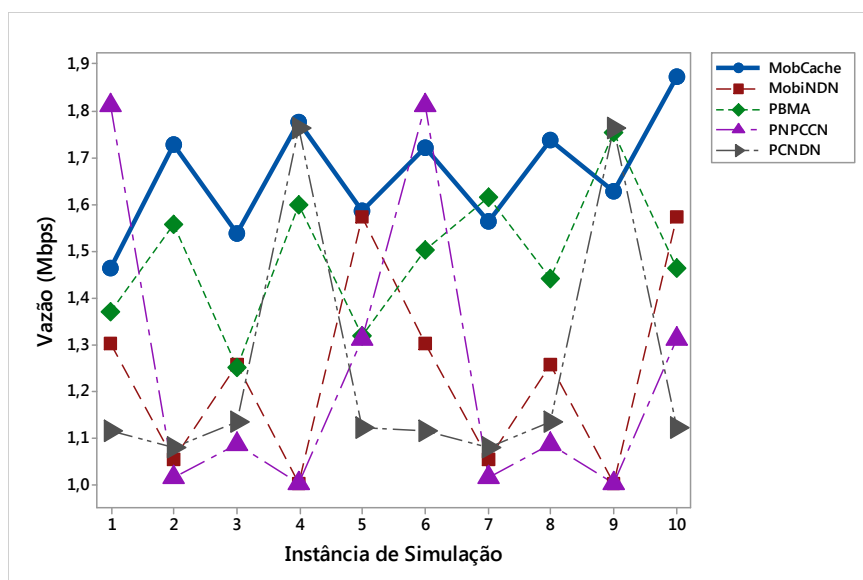


Figura 7.3 Gráfico de vazão em cada instância de simulação

A Figura 7.4 mostra o *boxplot* comparativo entre as propostas em relação à vazão. É possível perceber que o MobCache possui vazão superior às outras propostas. Isso porque além da eficácia na recuperação de pacotes, o MobCache distribui os conteúdos na rede de forma mais eficiente devido ao RankCache, melhorando a entrega de pacotes e aumentando o fluxo de pacotes. A proposta com menor desempenho foi a PNPCCN. O desempenho dessa proposta depende da topologia e funciona melhor em topologias hierárquicas, pois ao migrar para a outra rede, faz-se a suposição que o custo de comunicação é semelhante à anterior. Porém, em topologias não hierárquicas, a rede de destino pode não possuir uma comunicação válida com a rede de origem ou essa comunicação pode ser muito custosa.

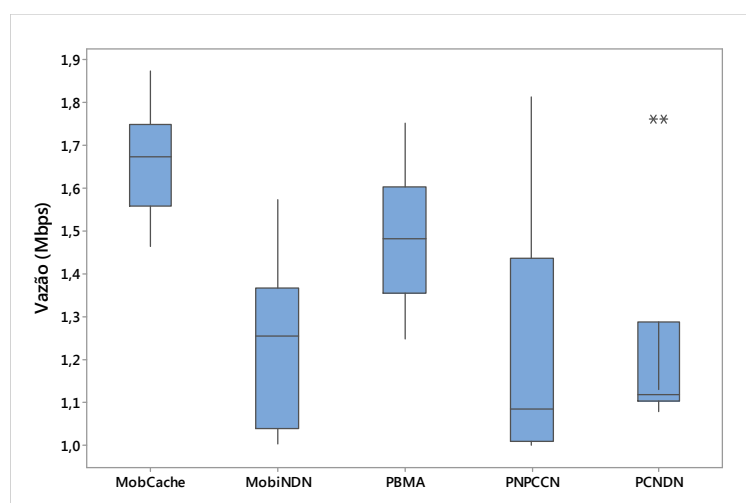


Figura 7.4 Boxplot comparativo da vazão - Cenário 1

Fazendo-se uma avaliação estatística comparativa dos resultados apresentados na Figura 7.3 e na Figura 7.4, obtém-se um *p-value* de 0,004. Isso mostra que, com 95% de confiança, a proposta MobCache possui a maior vazão

que a segunda colocada, o PBMA. Em relação à média das duas vazões, o ganho é de 12%. Quantitativamente, a proposta MobCache obteve 1,6614 Mbps e o PBMA 1,4863 Mbps de vazão. A Figura 7.5 mostra a soma total de Pacotes Retransmitidos pelo consumidor, variando-se a velocidade média dos provedores de 3 m/s a 30 m/s. O MobCache e o PCNDN são as duas únicas propostas com detecção e tratamento de perdas. Se levarmos em consideração um provedor de baixa e/ou média velocidade, ou seja, de 3m/s a 6m/s, essas duas propostas possuem um desempenho semelhante. A partir de 12 m/s, o MobCache começa a se mostrar superior ao PCNDN com poucos pacotes retransmitidos. Isso acontece porque em altas velocidades acontece muito deslocamento entre redes e o *buffering* feito pelo PCNDN prejudica o seu desempenho, devido à triangulação de pacotes de Interesse entre o antigo PoA e o novo. Portanto, em um cenário com grande número de *handovers*, percebe-se que a operação de recuperação de pacotes do PCNDN é bastante custosa. Por outro lado, como a recuperação do MobCache é simples e rápida, torna-se mais eficiente para provedores em altas velocidades.

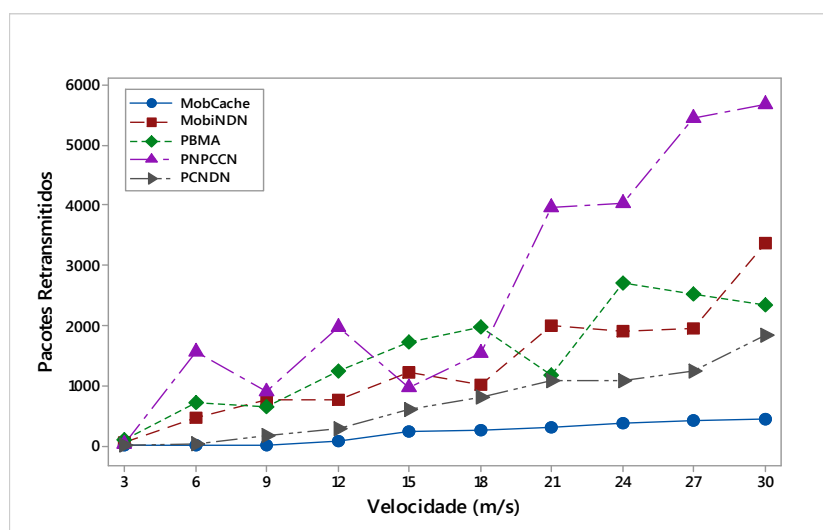


Figura 7.5 Soma de pacotes de Interesse retransmitidos pelo Provedor

A Tabela 7.1 mostra um resumo dos dados coletados nesse cenário, utilizando-se a métrica *pacotes de Interesse retransmitidos*. Percebe-se que, sem uma estratégia de contenção de perdas, algumas propostas chegam a perder mais de 5000 pacotes quando o provedor está em alta velocidade. Na avaliação geral, o MobCache é superior e precisa retransmitir 3 vezes menos pacotes que o PCNDN, segundo colocado.

Tabela 7.1 Estatística descritiva do número de pacotes retransmitidos

Proposta	Média	Desvio Padrão	Mínimo	Mediana	Máximo
MobCache	218,5	174,9	8	260,5	446,1
PCNDN	714	602	14	703	1832
MobiNDN	1354	971	64	1113	3379
PBMA	1515	876	102	1492	2701

PNPCCN	2612	2008	44	1771	5677
--------	------	------	----	------	------

A Figura 7.6 mostra o gráfico com o número de pacotes de controle na rede. Esses pacotes são enviados pelas propostas de mobilidade de provedor e de atualização das FIBs. Nesse cenário, não foram contabilizados pacotes de Interesse, de Dados ou de descoberta de prefixos (OSPFN) e variou-se o número de dispositivos requisitando informações. Em todas as simulações foram usados três provedores de conteúdo móveis, com velocidade de 30 m/s. Nessa velocidade há um número grande de *handovers* e isso gera muitos pacotes de controle na rede. Esse teste mostra o impacto do número de mensagens extras na rede. Propostas do tipo E-C e I-MIP geram grande número de mensagens na rede. O PCNDN gera muitos pacotes de controle devido ao processo de triangulação de Interesses armazenados no antigo PoA. Propostas AD-FIB geram menos *overhead*, mas ainda assim, são maiores do que as geradas pelo MobCache, que possui apenas uma mensagem de atualização de FIB (Interesse Virtual).

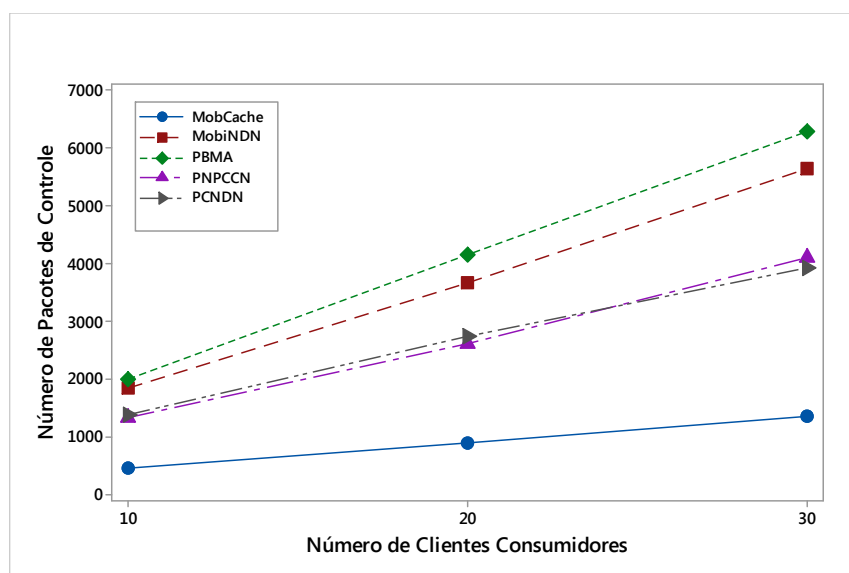


Figura 7.6 Número de Pacotes de Controle durante as simulações

7.2.1.2. Avaliação de QoE (Quality of Experience).

As métricas de QoE servem como extensão aos parâmetros de QoS e fornecem avaliação das redes de modo mais próximo à percepção dos usuários, permitindo avanços nas transmissões de aplicações de áudio e vídeo e podem proporcionar melhorias nos protocolos. Uma das métricas mais usadas de QoE é o PSNR (*Peak Signal to Noise Ratio*). O PSNR estima a qualidade do vídeo, em decibéis, comparando a transmissão original com a obtida no receptor. Para cada faixa de valores de PSNR, há uma qualificação para o vídeo que foi recebido pelo usuário conforme mostra a Tabela 7.2.

Para avaliação do PSNR foi utilizado o vídeo Akiyo (University). Todas as simulações foram feitas com o vídeo da mesma duração, codec, número de quadros (300 quadros) e resolução (352x288). Foi escolhido o vídeo Akiyo porque é um dos mais populares em simulações de QoE e, com isso, os resultados podem servir de comparação para outros trabalhos. O *framework* Evalvid (J. Klaue, 2003) foi integrado ao simulador desenvolvido por esta tese para permitir envio e avaliação de vídeos.

Tabela 7.2 Qualidade estimada de vídeo a partir do PSNR.

PSNR (dB)	> 37	31 – 37	25 – 31	20 – 25	< 20
Qualidade	Excelente	Bom	Aceitável	Pobre	Péssimo

Ao avaliar o QoE das estratégias de *caching*, percebeu-se que os vídeos recebidos pelo consumidor chegava com um auto grau de degradação, inclusive no MobCache. Notou-se que o fato de distribuir os *chunks* na rede de modo aleatório causava recuperação de pacotes fora de ordem. Isso acontece porque os *caches* não têm ciência da ordem dos quadros (*frames*) de um vídeo. Nas simulações com QoS esse problema acontece, mas não é relevante, uma vez que, diferente de um vídeo, a recuperação de um arquivo convencional não é sensível ao tempo de chegada entre os pacotes, nas rede TCP/IP a reordenação de quadros é feita nativamente por protocolos de transporte confiáveis como o TCP. Existem duas soluções possíveis para esse problema. A primeira é o aumento do buffer do consumidor, o que eleva o tempo de reordenação dos quadros, e a segunda é fazer um esquema de ordenação de quadros nos *caches* ao longo do caminho de entrega.

Para testar as duas hipóteses, foram elaboradas simulações empregando ambas as propostas. A Figura 50 mostra como se comportam as perdas de quadros com o MobCache em relação ao aumento do *buffer* no consumidor. Esse buffer é usado exclusivamente para o tráfego de vídeo e não armazena outro tipo de *chunk*. Na Figura 50, é possível notar que o aumento no tamanho do buffer implica na redução das perdas de quadros. No entanto, o vídeo demora mais para ser reconstruído, isso acontece por causa do tempo necessário para a ordenação dos quadros no consumidor.

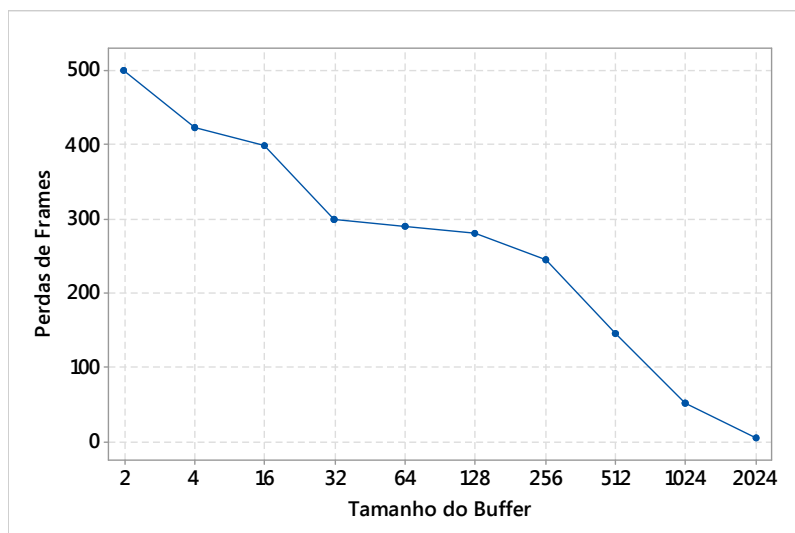


Figura 7.7 Tamanho do buffer vs perda de quadros (frames)

Idealmente, os quadros deveriam chegar ordenados no consumidor. Por isso, a segunda solução para o problema da distribuição dos *chunks* de vídeo é a ordenação dos quadros no caminho de entrega do conteúdo. Para isso, emprega-se uma ordenação crescente a partir do primeiro quadro transmitido. Dessa forma, os quadros de menor ordem ficam mais próximos do consumidor que os de maior ordem. Para isso, após ocorrer um acerto, o *chunk* é armazenado no próximo salto do pacote de Dados apenas se o número do quadro for menor do que os quadros do mesmo vídeo armazenados no *cache*. Além disso, foi implementado um esquema de remoção de *chunks*, nos *caches*, proporcional ao número do quadro, ou seja, os quadros iniciais da transmissão do vídeo são removidos mais rapidamente, isso agiliza a propagação de quadros de maior ordem do núcleo em direção à borda da rede. Todo esse processo é iniciado após o RankCache aceitar o armazenamento do *chunk* em questão. Essa operação modifica o processo original do RankCache e foi projetada apenas para esse experimento. O esquema faz com que os quadros fiquem ordenados a partir da borda da rede em direção ao núcleo.

A Figura 7.8 exibe o *boxplot* do PSNR com e sem essa estratégia de ordenação. Com a estratégia, é possível verificar que o PSNR do MobCache com a extensão é maior do que sem a estratégia. A média do PSNR com o MobCache sem a extensão é de 28 (Aceitável), segundo a Tabela 7.2, e com a extensão é de 35 (BOM), segundo a mesma tabela. Apesar da melhoria, o MobCache com ordenação gera muito *overhead*, pois, é necessário verificar a numeração de todos os pacotes de entrada e compará-los com os pacotes que estão no *cache*. Porém, é uma estratégia que obteve melhor desempenho em termos da métrica PSNR que o aumento do buffer e pode ser otimizada em trabalhos futuros.

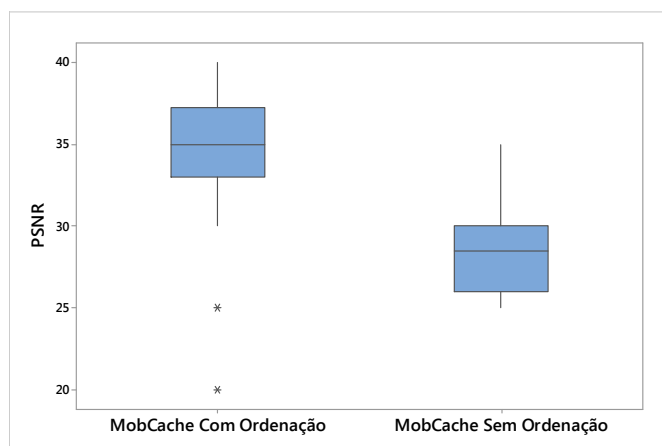


Figura 7.8 PSNR do MobCache com e sem ordenação

Após a análise inicial, foram feitas simulações e posteriormente avaliação de QoE com várias propostas da literatura. Os vídeos foram enviados um de cada vez pelo provedor. Durante a transmissão do vídeo o provedor permanecia em constante movimento de um ponto de acesso para outro, conforme o modelo RandomWayPoint. Após a obtenção do vídeo pelo consumidor, o próprio simulador computado o PSNR do vídeo recebido. Além das estratégias já avaliadas, inseriu-se o MobCache com ordenação, chamado de “MC com” nos gráficos e o MobCache com buffer, chamado de “MC Buffer”. O “MC sem” refere-se ao MobCache sem qualquer estratégia adicional.

A Figura 7.9 mostra o resultado do PSNR das soluções para mobilidade de provedor consideradas nesta tese. Pode-se perceber que o MobCache com as otimizações de aumento do buffer e o esquema de ordenação está com desempenho melhor que o dos concorrentes da literatura. Percebe-se que várias propostas possuem PSNR inferior à 20dB e são classificadas como “péssimo”, segundo a Tabela 7.2. A pior estratégia é o PCNDN, pois apesar de ter propostas de recuperação de pacotes perdidos durante a mobilidade, não atualiza a FIB e, portanto, demora para criar novas rotas, prejudicando a entrega de conteúdo de vídeo. O MobCache com estratégia de ordenação possui o melhor desempenho, mas como foi dito anteriormente, acarreta sobrecarga de processamento devido ao processo de ordenação. O MobCache com buffer é a segunda melhor proposta e sua implementação é mais simples.

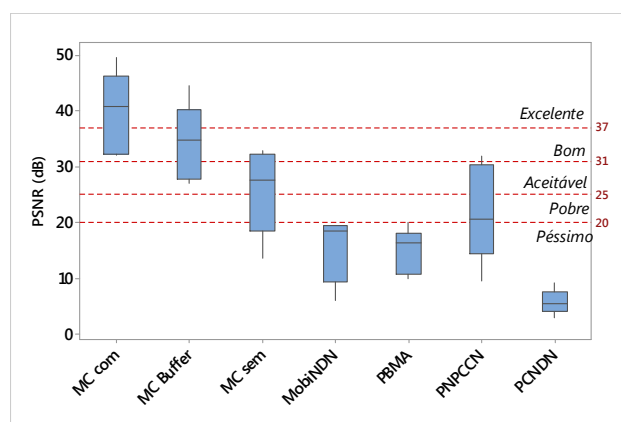


Figura 7.9 PSNR de todas as propostas

A Tabela 7.3 apresenta a estatística descritiva, em termos de PSNR, do experimento da Figura 52. Vale ressaltar que os valores apresentados na Tabela 7.3 correspondem a média obtida a partir de trinta repetições. O MobCache com ordenação possui ganhos de 16% em relação ao MobCache com estratégia de buffer e 53% em relação ao MobCache sem nenhuma estratégia, que por sua vez, possui 17% de ganho em relação ao PNPCCN, quarto colocado. Os ganhos do MobCache com ordenação em relação ao PNPCCN, segundo colocado entre as outras propostas, é de 80%. Portanto, mesmo sem estratégias adicionais o MobCache possui ganhos significativos às demais propostas. Na Figura 7.10 é possível obter resultados visuais de todas as propostas.

Tabela 7.3 Estatística descritiva do PSNR das simulações usando as estratégias de mobilidade

Estratégias	Média	Desvio	Mínimo	Mediana	Máximo
MC com	39,6	7,41	32,07	40,84	49,61
MC Buffer	34,2	6,98	27,07	34,84	44,61
MC sem	25,91	7,85	13,59	27,77	32,87
PNPCCN	22,11	8,78	9,59	20,77	31,87
MobiNDN	15,29	5,88	5,99	18,54	19,63
PBMA	14,85	4,07	9,98	16,3	20,02
PCNDN	5,849	2,202	3,024	5,456	9,159

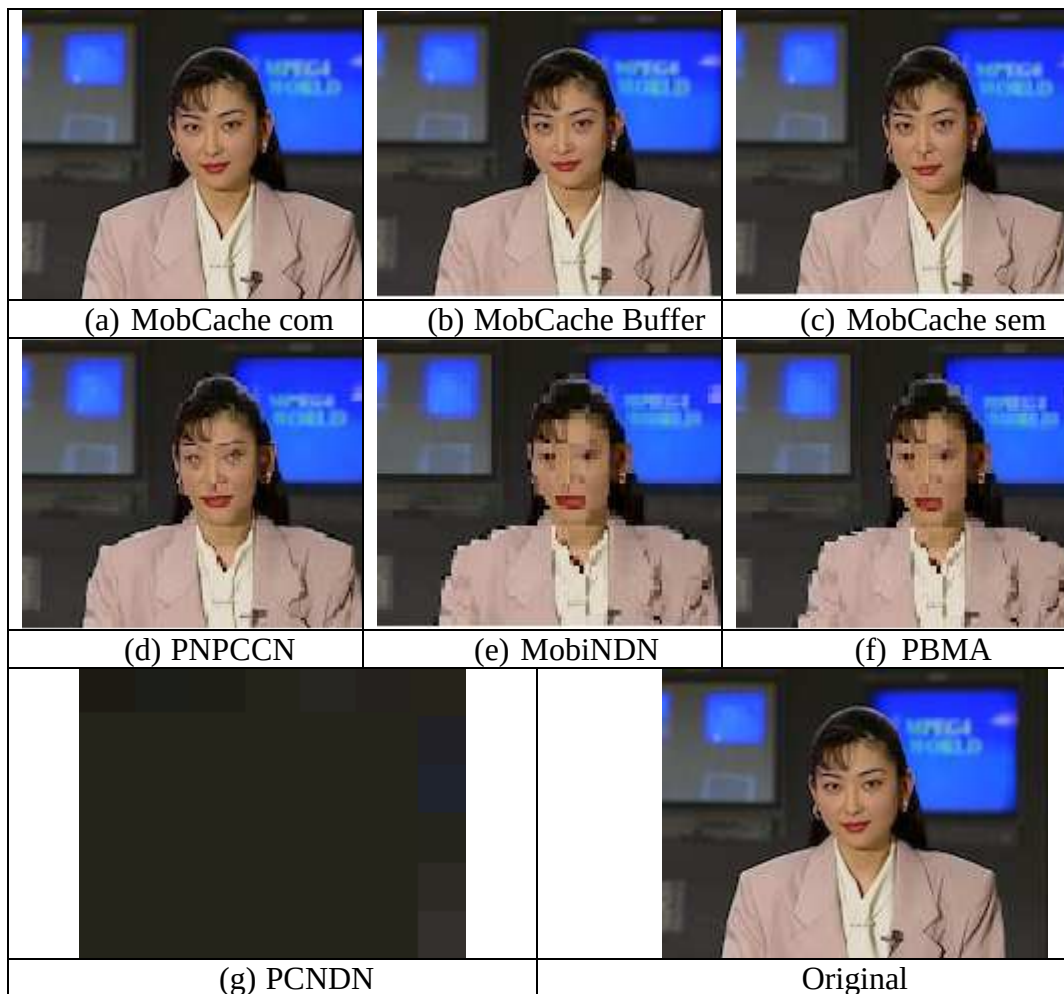


Figura 7.10 Pior quadro de cada proposta recuperado de uma das simulações

A Figura 7.10 mostra os piores quadros de cada proposta, ou seja, aqueles que obtiveram os menores valores para o PSNR, recuperados de uma das simulações. Na Figura 7.10, é possível observar os quadros das simulações com o MobCache (a), (b) e (c) com nitidez, mesmo com uma leve distorção no MobCache sem extensão, na Figura 7.10 (c). Nas outras propostas, as perdas causam distorções que prejudicam bastante a recuperação dos quadros. No PCNDN, o quadro com pior PSNR não apresenta qualquer informação.

O autor desta tese desconhece propostas de decisão de armazenamento de *cache* ou de mobilidade de provedor fazem avaliação de QoE. A avaliação apresentada nessa subseção mostrou que tráfego de vídeo precisa ser avaliado nos trabalhos de redes orientadas a conteúdo, isso porque apresenta problemas novos que não estão sendo tratados pelas propostas atuais. A seguir, será apresentada a avaliação através de modelagem analítica.

7.2.2. Avaliação analítica

Nesta subseção será detalhada a modelagem analítica. Foi feita a comparação da estratégia proposta com o CCN convencional e um esquema de *cache* proativo (Rao Y. Z., 2013) chamado PCNDN. O foco do PCNDN está na detecção e recuperação de pacotes perdidos durante o *handover* e foi escolhido, para comparação, por seu bom desempenho. Assim como nas simulações, será usado o modelo chamado *Random Waypoint* (Bettstetter, 2004) para representar a mobilidade do provedor. Como mostrado na Figura 7.11, esse modelo de mobilidade escolhe um ponto de origem (PO) em um dado espaço e move-se para esse ponto com uma velocidade constante. Após aguardar um tempo $\Delta\tau$, o provedor escolhe um ponto de destino (PD) aleatoriamente e se move até esse ponto com uma velocidade constante, e assim por diante. Chama-se transição o movimento de PO para PD.

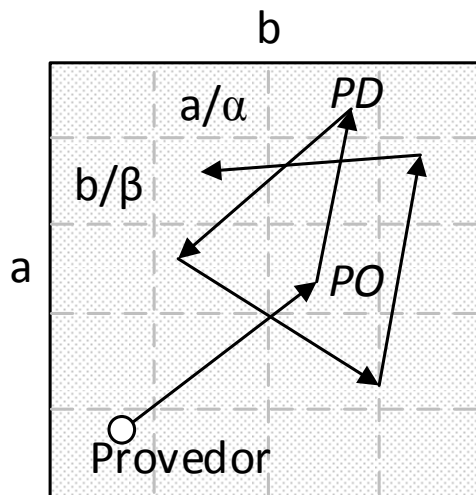


Figura 7.11 Modelo Random Waypoint

Duas métricas são importantes nesse modelo de mobilidade: a distância da transição (PO para PD), denominada D_T , e o tempo necessário para uma transição, denominado T_T . Se a velocidade do provedor (V_p) for constante durante o processo de movimento e a célula de movimentação for dada por um espaço retangular $\mathbf{a} \times \mathbf{b}$, o valor esperado da distância das transições $E(D_T)$ é dado pela Eq. 12 (Bettstetter, 2004).

$$E(D_t) = \frac{1}{15} \left[\frac{a^3}{b^2} + \frac{b^3}{a^2} + \sqrt{a^2 + b^2} \left(3 - \frac{a^2}{b^2} - \frac{b^2}{a^2} \right) \right] + \frac{1}{6} \left[\frac{b^2}{a} \operatorname{arcosh} \frac{\sqrt{a^2 + b^2}}{b} + \frac{a^2}{b} \operatorname{arcosh} \frac{\sqrt{a^2 + b^2}}{a} \right] \quad \text{Eq. 12}$$

Onde a função *arcosh* é a função inversa do cosseno hiperbólico dado pela Eq. 13.

$$\operatorname{arcosh}(x) = \ln(x + \sqrt{x^2 - 1}) \quad \text{Eq. 13}$$

Deste modo, o valor esperado do tempo de transição pode ser derivado como:

$$E(T_t) = E\left(\frac{D_t}{v_p}\right) \quad \text{Eq. 14}$$

$$E(T_t) = E(D_t) \times E\left(\frac{1}{v_p}\right) \quad \text{Eq. 15}$$

$$E(T_t) = \frac{1}{v_p} \times E(D_t) \quad \text{Eq. 16}$$

Para o cálculo do custo é preciso saber qual o valor esperado do número de *handovers* durante a movimentação. A variável aleatória correspondente é denotada por *C*. Supondo-se que c_i é uma instância da variável aleatória na transição *i*, o valor esperado de *C* é dado pela equação Eq. 17 (Bettstetter, 2004).

$$E(C) = \lim_{n \rightarrow \infty} \frac{1}{n} \left(\sum_{i=1}^n c_i \right) \quad \text{Eq. 17}$$

A variável aleatória *C* é a resultante de uma função determinística do ponto de partida inicial aleatório $P_1 = (P_{x1}, P_{y1})$ de um nó e seu próximo destino aleatório $P_2 = (P_{x2}, P_{y2})$. Como está sendo considerado que o espaço de cobertura é espaço retangular com lados **a** x **b** e com **α** linhas e **β** colunas, então sua área de cobertura é dada pela Eq. 18.

$$A(\alpha, \beta) = \frac{a}{\alpha} \times \frac{b}{\beta} \quad \text{Eq. 18}$$

Usando-se uma área retangular, o número de mudanças de células depende apenas da posição dos *waypoints* P1 e P2. Supõe-se que a posição horizontal da célula é indexada por $\xi = 1, 2, \dots, \alpha$ e sua posição vertical é dada por $\psi = 1, 2, \dots, \beta$. A posição dessa célula no espaço retangular é dada por (ξ, ψ) . Um ponto qualquer na posição (x, y) possui a posição na célula $\xi = \lfloor x\alpha/\alpha \rfloor$ e $\psi = \lfloor y\beta/\beta \rfloor$. Se um nó move de um *waypoint* na célula (ξ_1, ψ_1) para um *waypoint* na célula (ξ_2, ψ_2) , o número de mudanças de células é dada pela distância Manhattan (Black, 2006) entre as células, ou seja:

Eq. 19

$$c(\xi_1, \psi_1, \xi_2, \psi_2) = |\xi_1 - \xi_2| + |\psi_1 - \psi_2|$$

O valor esperado de C pode ser computado como a média de $c(\xi_1, \psi_1, \xi_2, \psi_2)$ levando-se em consideração todos os possíveis pares de células (Bettstetter, 2004):

Eq. 20

$$E(C) = \frac{1}{\alpha^2 \beta^2} \sum_{\xi_1=1}^{\alpha} \sum_{\psi_1=1}^{\beta} \sum_{\xi_2=1}^{\alpha} \sum_{\psi_2=1}^{\beta} c(\xi_1, \psi_1, \xi_2, \psi_2)$$

Assumindo que a duração de uma sessão (D) de comunicação do provedor é definida como o intervalo de tempo entre a primeira e a última requisição de conteúdo. Pode-se dizer que o número de mudanças (denominado M) durante uma sessão pode ser computado através da Eq. 21.

Eq. 21

$$E(M) = D \times \frac{E(C)}{E(T_T)}$$

Para avaliar o MobCache, usou-se a métrica custo total de *handover*. Essa métrica é definida como soma das latências dos pacotes trocados durante a mobilidade do provedor. A Figura 7.12 ilustra as mensagens trocadas no processo de *handover* dentro de um cenário genérico de mobilidade. As mensagens apresentadas na Figura 7.12 são usadas para montar as equações de custo. Na modelagem existem dois tipos de pacotes: o “MI” e o “MD”, onde o primeiro refere-se aos pacotes de Interesse trocados durante a mobilidade e o segundo, aos pacotes de dados.

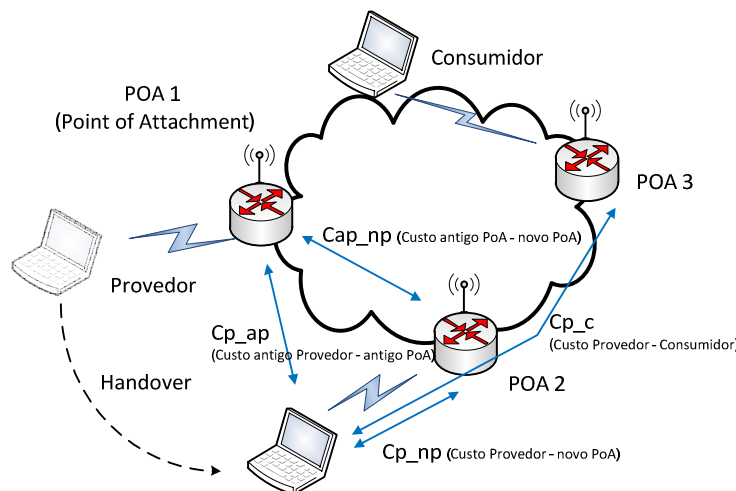


Figura 7.12 Mensagens utilizadas para o cálculo do custo de *handover*

Como foi dito anteriormente, a proposta foi comparada com o CCN convencional e com um esquema de *cache* proativo chamado PCNDN (Rao Y. Z., 2013). Portanto, serão apresentados os modelos do custo de *handover* do MobCache, do CCN convencional e do PCNDN. O custo total do *handover* no PCNDN (C_{PCNDN}) é dado pela Eq. 22, o do CCN (C_{CCN}) é dado pela Eq. 27 e do MobCache (C_{MC}) pela Eq. 31.

a) Custo de *handover* do PCNDN

Esta estratégia utiliza o armazenamento de pacotes de Interesse no antigo PoA. Esses pacotes são encaminhados ao provedor após a mobilidade e, portanto, não precisam ser retransmitidos pelo consumidor. O custo total de *handover* do PCNDN é dado pela somatória entre o custo de envio dos pacotes de Interesse e o custo de envio dos pacotes de dados (Eq. 22).

$$C_{PCNDN} = \sum_{i=1}^N Z_{PCNDN_MI}(i) + \sum_{i=1}^N Z_{PCNDN_MD}(i)$$

Eq. 22

A somatória do custo das mensagens de **Interesse** (MI) no PCNDN é dada pela Eq. 23. Essa equação é formada pelo produto entre o número de *handovers* (M), tamanho dos pacotes de Interesse (T_{MI}) e a somatória dos custos dos pacotes que participam da triangulação entre provedor, PoA1 e PoA2, para recuperação dos pacotes de Interesses mostradas na Figura 5.6. A outra parcela da fórmula representa o produto entre o número de pacotes perdidos (N_L) e o custo do envio dos pacotes de Interesse do antigo PoA (C_{ap_np}) para a nova localização do provedor (C_{p_np}).

$$\sum_{i=1}^N Z_{PCNDN_MI}(i) = M \times T_{MI} \times [(C_{p_ap} + C_{p_np} + C_{ap_np} + C_{p_c}) + N_L \times (C_{ap_np} + C_{p_np})]$$

Eq. 23

O custo da somatória das mensagens de **Dados** (MD) durante a mobilidade no PCNDN é dado pela Eq. 24. Essa equação é formada pelo produto entre o número de *handovers* (M), o tamanho do pacotes de Dados

(T_{MD}) e o custo total de envios de pacotes de dados do provedor (C_{p_np}) para o antigo PoA (C_{ap_np}). Percebe-se que o modelo para o custo de pacotes de dados não envolve o custo de envio partindo do provedor até o consumidor (C_{p_c}). Esse é o parâmetro mais oneroso e que influenciará negativamente no custo total do CCN convencional, apresentado mais adiante.

$$\sum_{i=1}^N Z_{PCNDN_MD}(i) = M \times T_{MD} \times [N_L \times (C_{ap_np} + C_{p_np})] \quad \text{Eq. 24}$$

Portanto, o custo total é dado pela somatória dos custos dos pacotes de Interesse e de Dados durante a mobilidade, conforme Eq. 25.

$$C_{PCNDN} = M \times T_{MI} \times [(C_{p_ap} + C_{p_np} + C_{ap_np} + C_{p_c}) + N_L \times (C_{ap_np} + C_{p_np})] + M \times T_{MD} \times [N_L \times (C_{ap_np} + C_{p_np})] \quad \text{Eq. 25}$$

Um parâmetro importante no cálculo do custo de *handover* é o número de pacotes perdidos. Esse número é uma variável aleatória, denominada N_L , que depende da taxa de transmissão dos pacotes do consumidor e do valor médio do tempo de transição entre as redes. Assumindo-se que a taxa de transmissão de pacotes segue uma distribuição exponencial com parâmetro λ , pode-se calcular N_L como:

$$N_L = E(T_t) \times \lambda$$

Eq. 26

b) Custo de *handover* do CCN

O custo total de *handover* do CCN é dado pela Eq. 27:

$$C_{CCN} = \sum_{i=1}^N Z_{CNN_MI}(i) + \sum_{i=1}^N Z_{CNN_MD}(i) \quad \text{Eq. 27}$$

Onde, a somatória do custo das mensagens de **Interesse** (MI) no CCN padrão é dada por:

$$\sum_{i=1}^N Z_{CNN_MI}(i) = M \times T_{MI} \times [N_L \times (C_{p_np} + C_{p_c})] \quad \text{Eq. 28}$$

É possível observar na Eq. 28 que, no custo do CCN, existem menos parcelas do que no custo do PCNDN da Eq. 23. Porém, o custo da mensagem

que sai do provedor até o consumidor (C_{p_c}), está sendo influenciado diretamente pelo número de pacotes perdidos (N_L) na Eq. 28, mas não na Eq. 23. Isso faz com que o custo do CCN se eleve muito mais para cada pacote perdido, o que não acontece no PCNDN.

E o custo da somatória das mensagens de **Dados** (MD) no CCN padrão é:

$$\sum_{i=1}^N Z_{CCN_MD}(i) = M \times T_{MD} \times [N_L \times (C_{p_{np}} + C_{p_c})] \quad \text{Eq. 29}$$

Portanto:

$$C_{CCN} = M \times T_{MI} \times [N_L \times (C_{p_{np}} + C_{p_c})] + M \times T_{MD} \times [N_L \times (C_{p_{np}} + C_{p_c})] \quad \text{Eq. 30}$$

c) Custo de *handover* do MobCache

$$C_{MC} = \sum_{i=1}^N Z_{MC_MI}(i) + \sum_{i=1}^N Z_{MC_MD}(i) \quad \text{Eq. 31}$$

A somatória do custo das mensagens de **Interesse** (MI) no MobCache é dada pela Eq. 32. O custo do *handover* com o MobCache não envolve a triangulação entre o antigo e o novo PoA. Portanto, esses custos ($C_{p_{np}}$ e $C_{ap_{np}}$) não entram no modelo. A variável P_E representa a probabilidade de erro do MobCache, ou seja, a chance do MobCache errar o número correto de pacotes de Interesse perdidos. Se o MobCache conseguir calcular com sucesso o número de pacotes perdidos, ele poderá enviar todos os pacotes de Dados corretamente e o consumidor não precisará reenviar pacotes de Interesse não satisfeitos. Quanto menor for o P_E , menor será o número de pacotes de Interesse perdidos que precisarão ser reenviados.

$$\sum_{i=1}^N Z_{MC_MI}(i) = M \times T_{MI} \times [(C_{p_{np}} + C_{p_c}) + N_L \times (C_{p_{np}} + C_{p_c}) \times P_E]$$

O custo da somatória das mensagens de dados durante a mobilidade no MobCache é dado pela Eq. 33. Nessa equação é possível ver que há relação entre o custo total entre provedor e consumidor (C_{p_c}) e o número de pacotes perdidos (N_L), como acontece no CCN. No MobCache, se o cálculo de dados gratuitos for incorreto, alguns pacotes serão perdidos e precisarão ser reenviados pelo consumidor, aumentando o custo do *handover*. Porém, o parâmetro P_L torna-se o contra peso desse cálculo. Com um P_L baixo, ou seja, com baixa probabilidade de erro, a influência da relação $N_L \times C_{p_c}$ no custo total fica bastante atenuada.

$$\sum_{i=1}^N Z_{RC_HD}(i) = M \times T_{MD} \times [N_L \times (C_{p_np} + C_{p_c}) \times P_L]$$

Eq. 33

Portanto:

$$C_{RC} = M \times T_{MI} \times [(C_{p_np} + C_{p_c}) + N_L \times (C_{p_np} + C_{p_c}) \times P_L] \\ + M \times T_{MD} \times [N_L \times (C_{p_np} + C_{p_c}) \times P_L]$$

Eq. 34

7.2.2.1. Resultados do Modelo

Para avaliar a proposta usou-se como área de cobertura $1000m \times 1000m$, que é um valor comumente usado para delimitar redes celulares. O parâmetro $A(\alpha, \beta)$, mostrado na Eq. 18, foi configurado como $\frac{\alpha}{\alpha} \times \frac{\beta}{\beta} = \frac{5 \times 10^4 m}{50} \times \frac{5 \times 10^4 m}{50} = 1000m \times 1000m$. A Tabela 7.4 mostra o resumo dos parâmetros usados na modelagem incluindo seus respectivos valores.

Tabela 7.4 Resumo dos parâmetros

Parâmetro	Descrição	Valor
C_{p_ap}	Custo de transmissão entre o provedor e o antigo ponto de acesso. (em saltos)	1
C_{pa_pn}	Custo de transmissão entre o ponto de acesso antigo e o novo ponto de acesso do provedor. (em saltos)	1
C_{p_np}	Custo de transmissão entre o provedor e o novo ponto de acesso. (em saltos)	1
C_{p_c}	Custo de transmissão por <i>byte</i> entre o novo ponto de acesso do provedor e o ponto de acesso do consumidor. (em saltos)	20 ou (de 1 a 900)

λ	Taxa de pacotes transmitidos pelo consumidor	80 ou (de 1 a 900 pacotes/s) ¹⁰ .
N_L	Variável aleatória que denota o número de pacotes perdidos. Depende da taxa de transmissão (λ) e do tempo de transição $E(T_t)$.	$E(T_t) \times \lambda$
P_E	Probabilidade de Erro do MobCache.	10% ou (de 1% a 100%)
T_{MI}	Tamanho de um único pacote de Interesse (em bytes)	40 bytes
T_{MD}	Tamanho de um único pacote de Dados (em bytes)	1040 Bytes
D	Duração da sessão	900s
V_p	Velocidade do provedor	30 ou (1 a 30m/s)
M	Número de mudanças de pontos de acesso durante a sessão.	Eq. 21

A Figura 7.13 apresenta o resultado do custo de *handover* em relação à variação da probabilidade de erro do MobCache. O cálculo errado do número de dados gratuitos obriga o consumidor a reenviar os pacotes de Interesse que foram perdidos. Essa operação eleva o custo do *handover*, por causa da distância entre o provedor e o consumidor. Quanto maior for a probabilidade de erro do MobCache, mais pacotes precisarão ser reenviados e maior será o custo total do *handover*. A Figura 7.13 mostra que com até 30% de probabilidade de erro, o MobCache é superior ao PCNDN e, a partir desse valor, o MobCache perde desempenho. Para essa avaliação, todos os outros parâmetros foram fixados conforme a Tabela 7.4.

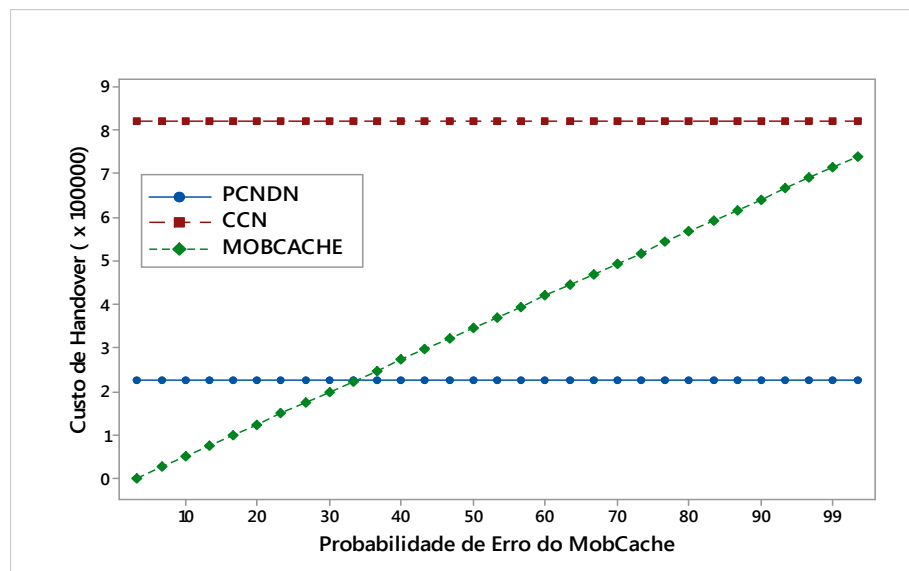


Figura 7.13 Custo de *handover* em relação à probabilidade de erro do MobCache

¹⁰ Significa que o padrão é usar 80 pacotes/s, mas em um dos gráficos variou-se de 1 a 900.

A Figura 7.13 mostra também que com 100% de erro, o MobCache tem o comportamento similar ao do CCN convencional, ou seja, todos os pacotes de Interesse perdidos precisarão ser retransmitidos. Para fins de comparação, foram feitas simulações com o MobCache no simulador e verificou-se que para cada 100 pacotes enviados durante o *handover*, um pacote precisava ser retransmitido pelo consumidor, ou seja, o valor médio de erros no MobCache gira em torno de 1%. Para a modelagem, optou-se por usar um valor 10 vezes maior de probabilidade de erro para capturar cenários intermediários, por isso, o uso de 10% na probabilidade de erro, ao invés de 1%, como nas simulações, ou o pior caso de 30%, como apresentado na Figura 7.13.

A Figura 7.14 mostra o resultado do custo total de *handover* em relação à mobilidade do usuário. Pode-se observar que o custo de *handover* aumenta de forma proporcional à velocidade do provedor. Isso porque, pelas equações Eq. 16 e Eq. 21, a velocidade do provedor V_p é diretamente proporcional ao número de mudanças de pontos de acesso (M). Como o aumento de M afeta o custo total de *handover*, quanto maior for a velocidade do provedor, maior será esse custo. Com o provedor a 30m/s, o PCNDN atingiu um custo 150% maior que o MobCache e o CCN padrão, 800% maior. Isso acontece porque, no CCN convencional, é necessário retransmitir todos os pacotes de Interesse perdidos e isso eleva bastante o custo. No PCNDN o custo é menor que no CCN padrão, porém o fato da proposta usar triangulação de mensagens para recuperação de pacotes de Interesse faz com que o custo se eleve mais quando há maiores perdas de pacotes, como no caso de aumento da velocidade do provedor. No MobCache não há triangulação, a proposta calcula o número provável de pacotes de Interesse perdidos e envia imediatamente os pacotes de Dados correspondentes para os consumidores.

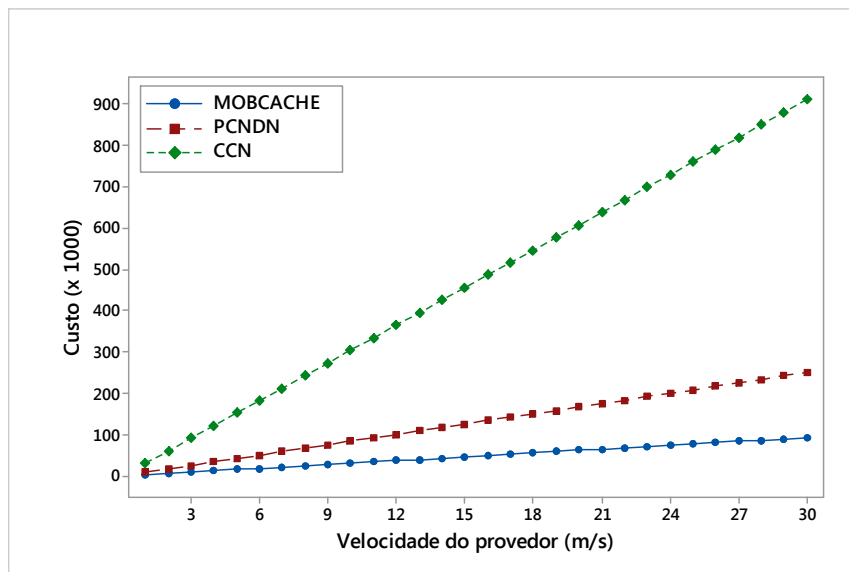


Figura 7.14 Custo total de *Handover* em Relação à velocidade do Provedor

A Figura 7.15 mostra o custo total de *handover* com a variação do parâmetro de taxa de transmissão de pacotes (1 a 900 pacotes/s). A velocidade nesse experimento foi fixada em 10 m/s e o tempo de duração da sessão é de 900s. Segundo a Figura 7.15, quanto maior for a taxa de transmissão maior será o custo do *handover*. Se o tempo de *handover* for fixado, quanto maior for a taxa de chegada de pacotes, maiores serão as perdas devido ao tempo de desconexão. Portanto, a Figura 7.15 mostra que para valores altos de λ , as perdas são maiores nas propostas CCN e PCNDN. Por outro lado, o MobCache consegue manter o custo bem menor, devido à sua estratégia eficiente de atualização das rotas.

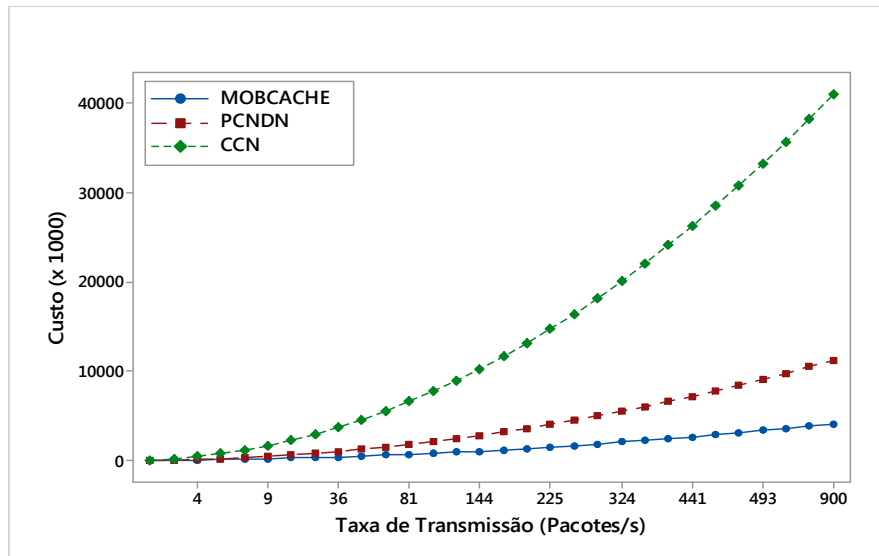


Figura 7.15 Custo total de *handover* em relação à taxa de transmissão

Um dos parâmetros que pode afetar o MobCache é o $C_{p,c}$ (Custo de transmissão entre o provedor e o consumidor). Esse parâmetro está diretamente relacionado ao P_E , ou seja, quanto menor for a probabilidade de erro, menor será a influência do $C_{p,c}$ no custo de *handover* do MobCache. Nessa avaliação, o P_E foi fixado em 10%. A Figura 7.16 mostra o custo de *handover* com a variação do $C_{p,c}$. O parâmetro foi configurado entre 1 e 900. Percebe-se que mesmo com o alto valor de $C_{p,c}$, o custo do CCN Padrão é 11 vezes maior que o MobCache e do PCNDN 3 vezes maior, ou seja, ganhos de 200%. Isso acontece porque, no MobCache, o aumento do $C_{p,c}$ só afeta o custo total se a probabilidade de erro for alta, o que não é o caso.

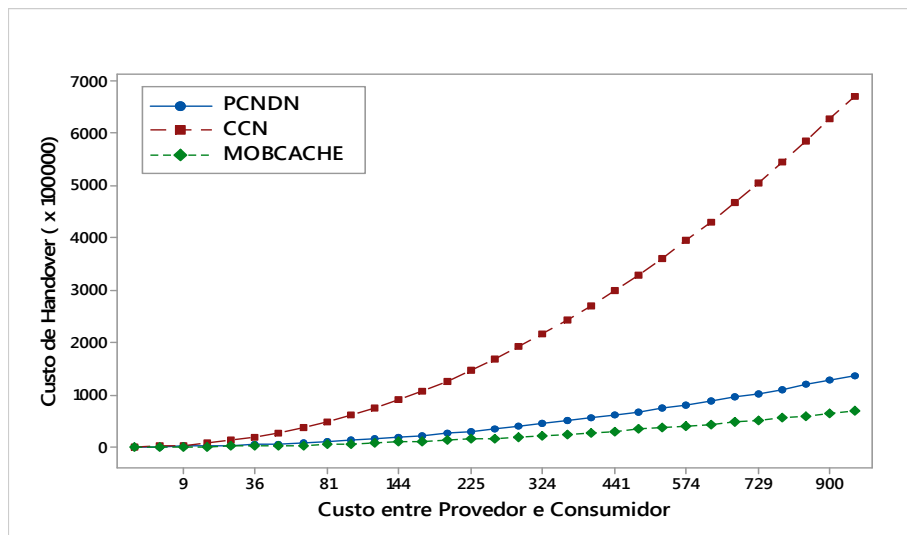


Figura 7.16 Custo de *handover* em relação ao custo de transmissão entre Provedor e Consumidor

7.2.3. Validação do Modelo Analítico

Ao criar um modelo analítico, é necessário verificar se o mesmo se comporta de forma suficientemente satisfatória com relação aos objetivos do estudo. A validação é realizada comparando-se o comportamento do modelo com o de um sistema real, ou simulado, quando ambos são submetidos às mesmas variáveis de entrada. Como na modelagem, o objetivo do estudo é o custo do *handover*, para validação, a mesma métrica foi utilizada. Para tal, foram utilizados na simulação e na modelagem os mesmos parâmetros da Tabela 7.5, extraídos da Tabela 7.4 Resumo dos parâmetros.

Tabela 7.5 Tabela de parâmetro para o Modelo e para a Simulação

Parâmetro	Descrição	Valor
$C_{p_{ap}}$, $C_{pa_{pn}}$, $C_{p_{np}}$	Custo de transmissão ponto a ponto. (em saltos)	1
C_{p_c}	Custo de transmissão por byte entre o novo ponto de acesso do provedor e o ponto de acesso do consumidor. (em saltos)	20
λ	Taxa de pacotes transmitidos pelo consumidor	80 pacotes/s.
P_E	Probabilidade de Erro do MobCache.	1%
T_{MI}	Tamanho de um único pacote de Interesse (em bytes)	40 bytes
T_{MD}	Tamanho de um único pacote de Dados (em bytes)	1040 Bytes
D	Duração da sessão	900s
V_p	Velocidade do provedor	30m/s

O cenário possui 20 nós disponíveis de forma aleatória em um espaço de 1000m x1000m. Como mostra a Tabela 7.5, a probabilidade de erro do

MobCache na modelagem foi fixada em 1%, semelhante à utilizada na simulação. A duração da sessão é de 900s e a velocidade do provedor varia de 1m/s até 30m/s. Cada salto entre dois roteadores possui o custo de 1 e a unidade de medida do custo é o número de saltos multiplicado pelos *bytes* de cada pacote (saltos x *bytes*).

A Figura 7.17 mostra a comparação entre os custos na modelagem e na simulação com o MobCache, variando-se a velocidade do provedor. É possível verificar que ambas se comportam de modo semelhante, o custo de *handover* aumenta em resposta ao aumento da velocidade do provedor. As simulações são mais dinâmicas, possuem elementos aleatórios e processos que não pode ser controlados, como tráfego de controle, atrasos devido ao enlace, filas de mensagens nos roteadores entre outros. Portanto, devido a essa dinamicidade dos simuladores há maior variação nos resultados se compararmos com a modelagem. Apenas estatisticamente é possível confirmar se ambas possuem o mesmo comportamento.

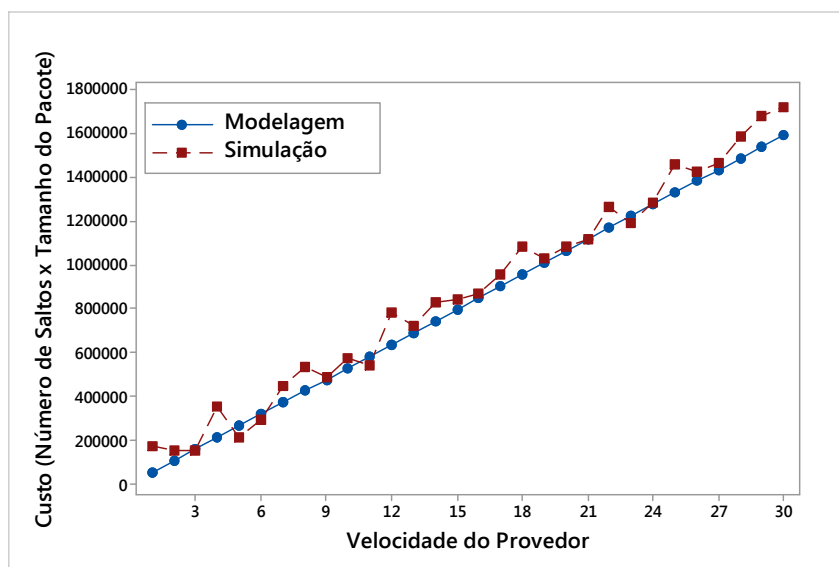


Figura 7.17 Comparação entre custo de *handover* na Modelagem e na Simulação do MobCache

A Figura 7.18 mostra o intervalo das médias dos resultados da modelagem e simulação. A média foi calculada a partir dos dados exibidos na Figura 7.17. O custo médio obtido com a modelagem foi de $8,26 \times 10^5$ e da simulação foi de $8,22 \times 10^5$, diferença de 2904 ou de 0,35%. A Figura 7.18 mostra que ambos os intervalos possuem pontos coincidentes, o que aumenta a suspeita de os resultados serem semelhantes. Porém, para ter um embasamento estatístico utilizou-se o teste T-Emparelhado para analisar a comparação entre os dois casos.

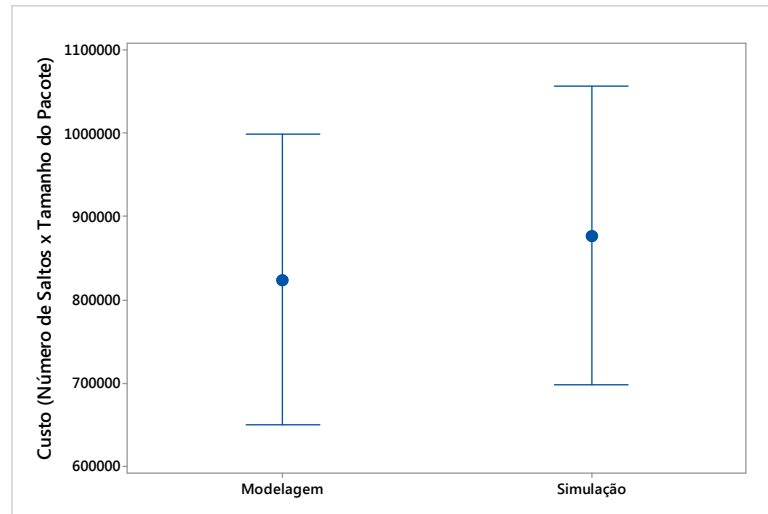


Figura 7.18 Intervalo das médias com 95% de confiança

O teste T-Emparelhado foi conduzido sobre as seguintes hipóteses.

- H_0 : A diferença entre valor médio do custo de *handover* da modelagem e da simulação é igual a zero.
- H_A : A diferença entre o valor médio do custo de *handover* da modelagem e da simulação é diferente de zero.

O teste T-Emparelhado pode ser usado porque as amostras coletadas da modelagem e da simulação seguem uma distribuição normal e são independentes. A Figura 7.19 mostra o boxplot da diferença entre as médias da modelagem e da simulação. O gráfico foi extraído a partir do teste T-Emparelhado e mostra que a hipótese nula (H_0) está dentro do intervalo estimado para a diferença das médias. Além disso, a Figura 7.19 destaca que o p-value é 0,787, ou seja, muito maior do que 0,05 e, com isso, não podemos rejeitar a hipótese nula. Portanto, com 95% de certeza podemos afirmar que não há diferença estatística entre o custo de *handover* com a simulação e com a modelagem.

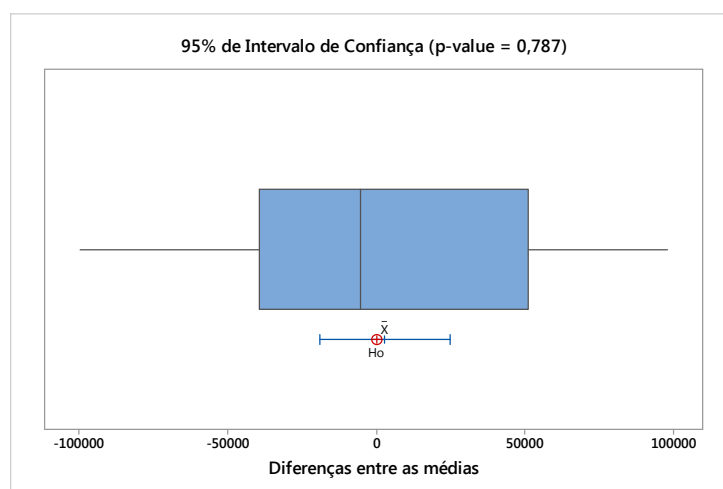


Figura 7.19 Boxplot das diferenças entre as médias

7.3. Considerações Finais

Neste capítulo foi proposto e avaliado o MobCache, um arcabouço para tratar a mobilidade de provedor nas CCNs visando melhorar a QoS levando-se em consideração tanto a parte cabeada da rede quanto a sem fio. Os resultados de QoS mostraram a superioridade da proposta em relação a outros trabalhos da literatura, com ganhos de mais de 300%. Além do QoS, o QoE também foi avaliado. Mostrou-se que propostas de distribuição de *chunks* podem afetar significativamente tráfegos de vídeos sensíveis ao atraso. As propostas foram avaliadas através da métrica PSNR, que reflete a qualidade de vídeo percebida pelo usuário. Estendeu-se o MobCache para melhorar seu desempenho nesses cenários e os resultados mostraram ganhos de 16%, 17% e 52% em relação às propostas da literatura. Em seguida, foi feita uma modelagem do MobCache e comparada com outras propostas. O MobCache também foi superior, apresentando ganhos de 150%, com variação de velocidade do provedor, e 300%, variando a distância entre o consumidor e provedor. Ao final do capítulo, os resultados da modelagem foram validados com os da simulação, utilizando os mesmos dados de entrada e obtendo-se resultados semelhantes. Essa relação foi comprovada estatisticamente através de teste de hipóteses com o T-Emparelhado. Os resultados do MobCache, de um modo geral, mostraram que a solução é promissora, pois obteve bons resultados em comparação com os trabalhos da literatura. Por outro lado, existem alguns pontos em aberto, como o desempenho em cenários com tráfego de vídeo. Apesar de ter sido proposto um aprimoramento, é necessário maior investigação nesse âmbito com variação de cenários e de métricas. O próximo capítulo finaliza a tese e mostra alguns possíveis direcionamentos futuros desta pesquisa.

8

CONSIDERAÇÕES FINAIS

Esta tese apresentou e avaliou propostas para os problemas de armazenamento eficiente de conteúdo na rede e de suporte à mobilidade do provedor na arquitetura ICN. Inicialmente, a tese mostrou os benefícios das redes ICN e suas principais desvantagens. O suporte à mobilidade de provedor foi apontado como um dos problemas do ICN, e esse se tornou o foco das soluções da tese. O texto seguiu com as principais características de cada tecnologia abordada no trabalho, como ICN, NDN, CCN, OSPFN, Mobilidade de Provedor. Mostrou-se como a mobilidade de consumidor obtém sucesso na continuidade da transmissão após o *handover* e destacou-se também o insucesso da mobilidade de provedor, apontando-se sua causa.

O Capítulo 3 mostrou os principais trabalhos relacionados ao tema da tese. Foi criada uma classificação própria para esses trabalhos baseada em como as tabelas internas dos roteadores são atualizadas. Os grupos destacados são: AD-FIB, E-C e I-MIP, ou seja, atualização direta da FIB, baseado em entidades centralizadas e inspirados no protocolo MIP, respectivamente. O Capítulo seguinte apresentou a primeira solução proposta pela tese, o algoritmo de ranqueamento de *cache*, o RankCache. Essa estratégia melhora a distribuição de pacotes nos roteadores CCN (*In-Path ou In-Network Caching*), os resultados das simulações mostraram ganhos entre 10% e 30%. A segunda solução proposta pela tese foi resolver o problema gerado pela mobilidade de provedor no CCN. A ideia utiliza um pacote de Interesse virtual para atualizar o caminho entre consumidor e o provedor, após a mobilidade desse último. Os resultados mostraram que a proposta obteve ganhos de até 142%.

O Capítulo 6 detalhou, através de simulações, a relação entre mobilidade e *caching*, para isso, foram aplicados testes de hipóteses e a relação foi comprovada com 95% de intervalo de confiança. Esse Capítulo representou um importante embasamento para a proposta final da tese, o MobCache. Essa última proposta foi criada através da união entre duas outras anteriores, o RankCache e a solução para mobilidade de provedor. O MobCache foi avaliado através de modelagem e simulação. Na simulação, mostrou-se que com o MobCache há redução de até três vezes no número de pacotes retransmitidos na rede em relação à segunda melhor proposta. Além do QoS, foi avaliado tráfego de vídeo

usando como referência o PSNR, que faz parte das métricas que estão mais próximas da percepção do usuário (QoE). Na avaliação de vídeo, os ganhos variaram de 16% a 80%. Após a simulação, mostrou-se a avaliação analítica da proposta, o foco da modelagem foi o custo de *handover*, os ganhos do MobCache em relação aos trabalhos da literatura foram de 150% a 300%. Esse capítulo terminou com a validação do modelo analítico usando comparação com a simulação. Aplicou-se também o teste de hipóteses e com 95% de confiança pode-se dizer que modelagem e simulação possuem a mesma saída para o mesmo grupo de entradas.

O autor deste trabalho acredita que as redes ICN representam um grande potencial para o futuro. Elas estão sendo testadas e validadas em várias partes do mundo (Foundation., 2017), principalmente em Internet das Coisas e Redes Veiculares. O gerenciamento de mobilidade é crucial para o bom funcionamento dessas redes e o problema da mobilidade de provedor precisa ser resolvido. A tese contribui com várias propostas, tanto no âmbito cabeado quanto no sem fio, mostrando seus reais ganhos na avaliação analítica e em simulações de QoS e QoE.

8.1. Contribuições

Este trabalho obteve algumas contribuições originais e significativas:

- RankCache: Criação de um Modelo e Algoritmo para melhorar a distribuição dos *chunks* em redes CCN cabeadas e aumentar a taxa de acertos nos *caches*.
- Proposta de mobilidade de provedor usando Interesse Virtual.
- MobCache: Proposta de solução para o problema de mobilidade de provedor usando elementos tanto das redes cabeadas como sem fio.
- Análise e demonstração estatística da relação entre Mobilidade e *Caching*.
- Modelo de avaliação para mobilidade de provedor usando *RandomWaypoint*.
- Desenvolvimento de um simulador completo ICN com módulos de avaliação de vídeo, ainda não vistos na literatura.

8.2. Trabalhos Futuros

A pesquisa sobre a mobilidade dos provedores ainda é recente se comparada com outras características ICN como *caching*, portanto, ainda há muitos pontos que precisam ser investigados. Vários trabalhos podem ser derivados desta tese e são caracterizados em diversas áreas como: Redes definidas por software, Mobilidade e aplicações de tempo real, IoT, VANETs e *Green Networking*. Outras questões, como decisões baseadas no QoS/QoE, suporte de sistemas legados, escalabilidade, segurança e privacidade, estão

envolvidas em quase todos os aspectos das áreas supracitadas. Em seguida, serão detalhadas algumas propostas mais importantes e relevantes que podem ser derivadas desta tese.

a) Redes Definidas por Software

O paradigma SDN (*Software-Defined Networking*) (Rowshanrad, 2014) separa o plano de controle do plano de dados, permitindo, assim, maior flexibilidade, gerenciamento e possibilidade de redes programáveis. O OpenFlow (McKeown, 2008) é a implementação mais bem conhecida e mais ampla de SDN. No OpenFlow, as aplicações são executadas nos chamados controladores. Eles implementam a lógica de rede e suas funções (por exemplo, firewall, balanceamento de carga, QoS etc.). A lógica da rede é traduzida em regras que são instaladas nos switches e roteadores. Os benefícios da sinergia entre SDN e CCN, em redes cabeadas, são evidenciados em (Nguyen, 2013) e (Sun, 2014). A visão que o SDN fornece da rede torna-se muito poderosa para o CCN, pois, uma vez que os controladores possuem essa visão global, soluções de encaminhamento de conteúdo podem ser mais eficientes e otimizadas. Soluções para o gerenciamento de mobilidade/*handover* via OpenFlow não são novas, no entanto, a mobilidade de provedor no CCN com base em SDN impõe questões específicas. Talvez um dos problemas dessa sinergia sejam os requisitos de escalabilidade, uma vez que a adoção de SDN implica, em geral, uma solução centralizada para CCN. Apesar disso, esforços têm sido dedicados aos controladores SDN distribuídos em literatura não-CCN. Outra questão é que a mobilidade do provedor pode gerar enormes quantidades de tráfego de interesse e pacotes de dados dependendo da distância entre controlador e provedor de conteúdo, além de atraso extra na instalação das novas regras. As alterações dos pacotes CCN e a falta de flexibilidade para lidar com a camada de aplicação são também limitações desta integração. Com a crescente adoção de tecnologias SDN torna-se evidente que a cooperação de ambas os paradigmas trará benefícios, porém, é necessária uma maior investigação sobre a estreita integração da CCN e da SDN para comprovar seus benefícios.

b) Mobilidade e Aplicações de Tempo Real

A demanda por aplicativos em tempo real cresceu paralelamente à busca por conteúdo. Este tipo de aplicação é amplamente estudado em redes convencionais, e seus requisitos estritos de QoS são conhecidos. No ICN, poucos estudos abordam este tipo de aplicação. Avaliações de aplicações VoIP foram apresentadas pela primeira vez em (Jacobson V. S., 2009), apesar de fornecer um custo extra no encaminhamento de pacotes, devido à nova camada de segurança adicionada, a aplicação obteve um resultado semelhante ao RTP/UDP. No entanto, o experimento mostrou apenas uma avaliação preliminar em cenários simplificados. Outros artigos têm mostrado o uso de CCN em jogos online. O trabalho (Chen J. A., 2012) apresenta uma implementação de um MMORPG (*Massively Multiplayer Online Role-Playing Game*) baseado no

CCNx, implementação oficial do CCN. O experimento mostrou que o protocolo pode atender aos rígidos requisitos de latência que este tipo de aplicação requer, além disso, foram apresentados ganhos de escalabilidade. Porém, em aplicações de áudio e vídeo, os dados são gerados em tempo real e imediatamente distribuídos na rede, e na presença de mobilidade, especialmente no provedor, a qualidade do serviço pode ser impactada. Poucos estudos têm esse foco de pesquisa. Portanto, embora conhecidos em redes tradicionais, esses tipos de aplicações precisam ser investigados em termos de mobilidade em redes ICN.

c) Internet das Coisas e Redes Veiculares

A Internet das coisas e veículos conectados (IoTV do inglês *Internet of connected Things and vehicles*) trazem vários recursos desafiadores para a disseminação convencional de conteúdo. Sensores e atuadores incorporados ao corpo humano, *smartphones* e veículos (bicicleta, carro, drones, etc) são potenciais provedores móveis de conteúdo. O conteúdo pode variar em velocidade (consumidor a pé ou de carro), parâmetros ambientais (por exemplo, umidade, temperatura, velocidade do vento, poluição), informações de segurança com relação a perigos em estradas. Toda esta disseminação de conteúdo possuem diferentes requisitos de QoS/QoE. As soluções de mobilidade do provedor para IoTV devem considerar uma ampla variedade de variáveis (grau de mobilidade, conectividade de rede, requisitos de QoS e QoE). Neste contexto, o comportamento de *cache* nesses ambientes é algo que precisa ser investigado.

A comunicação entre as aplicações VANETs deve ocorrer através de conexão intermitente e alta mobilidade do nó. Esse é um cenário topológico desafiador e altamente dinâmico. Essas redes também podem combinar informações de diferentes fontes. O uso de ICN em redes VANETs poderia trazer grandes benefícios (Amadeo M. C., 2016), por exemplo, o armazenamento nos elementos da rede pode acelerar o conteúdo de recuperação por replicação de dados em vários veículos. Alguns trabalhos (Wang L. W., 2012), (Tonneau, 2014) e (TalebiFard P. L., 2015) exploram os benefícios do uso do ICN em VANETs. No entanto, assim como outros dispositivos de produção de conteúdo, os veículos também sofrem com problema de mobilidade do provedor e dada a dinamicidade desses tipos de rede é uma grande oportunidade de pesquisas para o futuro das redes ICN.

d) Redes Verdes (*Green Networking*)

Soluções para eficiência energética (*Green Network* ou Redes Verdes) nas redes de computadores já foram projetadas para arquiteturas cabeadas ICN (Choi, 2012). O armazenamento nos nós CCN, evita a sinalização excessiva em relação aos provedores de conteúdo e, portanto, beneficia a engenharia de tráfego ciente de energia no *backbone* da rede. Antes de executar o *handover*, o conteúdo do provedor pode ser armazenado em *cache* em roteadores de conteúdo na rede de acesso para evitar interrupção de sessão enquanto o

dispositivo está sendo liberado, evitando assim perdas de pacotes e atraso para entregar conteúdo ao consumidor. Por outro lado, o consumo de energia do fornecedor com carga de bateria também deve ser levado em conta. Por exemplo, pode-se pensar no armazenamento em *cache* em dispositivos móveis vizinhos, em vez de usar roteadores de conteúdo na rede de acesso devido a maior potência necessária para enviar o conteúdo para um PoA. Assim, os dispositivos vizinhos com mais energia poderiam servir como um nó de retransmissão ou até mesmo ser responsável pelo *cache* e transmitir o conteúdo durante a mobilidade da fonte original. Certamente, para uma operação eficiente, são necessários incentivos e acordo de confiabilidade entre os dispositivos. Existem poucas pesquisas que relacionem as redes verdes e o consumo de energia com soluções de mobilidade de provedor, porém, elas são de suma importância para o sucesso do ICN no contexto de dispositivos com bateria limitada. Cenários desafiadores com requisitos contraditórios, como suporte QoS/QoE versus consumo de energia para mobilidade de provedores devem ser abordados em futuras pesquisas.

REFERÊNCIAS

- Ahlgren, B. D. (2012). A survey of information-centric networking. *IEEE Communications Magazine*, v. 50, n. 7, p. 26-3.
- Akyildiz, I. F. (2004). A survey of mobility management in next-generation all-IP-based wireless systems. *IEEE Wireless Communications*, pp. 16-28.
- Amadeo, M. C. (2012). CRoWN: Content-centric networking in vehicular ad hoc networks. *IEEE Communications Letters*, pp. 1380-1383.
- Amadeo, M. C. (2016). Information-centric networking for connected vehicles: a survey and future perspectives. *IEEE Communications Magazine*, 98-104.
- Arumaithurai, M. C. (2015). Prototype of an ICN based approach for flexible service chaining in SDN. In *Computer Communications Workshops (INFOCOM WKSHPS), 2015 IEEE Conference on*, pp. 5-6.
- Atzori, L. A. (2010). The Internet of things: A survey. *Computer networks*, p. 2787-2805.
- Augé, J. C. (2015). Anchor-less Producer Mobility in ICN. In *Proceedings of the 2nd International Conference on Information-Centric Networking*, pp. 189-190.
- Baccelli, E. M. (2014). Information centric networking in the IoT: Experiments with NDN in the wild. *arXiv preprint arXiv:1406.6608*.
- Bettstetter, C. H.-C. (2004). Stochastic properties of the random waypoint mobility model. *Wireless Networks*, pp. 555-567.
- Black, P. E. (2006). Manhattan distance. *Dictionary of Algorithms and Data Structures*.
- Bruno, F. C. (2014). Optimal content placement in icn vehicular networks. *International Conference and Workshop on the Network of the Future*, pp. 1-5.
- CCN-lite. (2016). Lightweight CCN Simulator. *Disponível em: <http://www.ccn-lite.net/>*, Acessado em: mar/2016.
- CCNx.org, P. (2016). Palo Alto Research Center (PARC). *Disponível em: <http://www.ccnx.org/>*, Acessado em : mar/2016.
- Cerf, V. G. (2005). A protocol for packet network intercommunication. *ACM SIGCOMM Computer Communication Review*, v. 35, n. 2, p. 71-82.

- Cha, M. K. (2009). Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Transactions on Networking (TON)*, 17(5), 1357-1370.
- Chai, W. K. (2011). Curling: Content-ubiquitous resolution and delivery infrastructure for next-generation services. *IEEE Communications Magazine*, 49(3), 112-120.
- Chai, W. K. (2013). Cache “less for more” in information-centric networks (extended version). *Computer Communications*, 36(7), 758-770. .
- Chambers, C. J. (2016). Future traffic demands and characteristics from a media perspective. *Phil. Trans. R. Soc. A* 374.2062.
- Chen, J. A. (2012). G-copss: A content centric communication infrastructure for gaming applications. *IEEE 32nd International Conference on Distributed Computing Systems*, pp. 355-365.
- Chen, M. S. (2014). Big data: A survey." *Mobile Networks and Applications*. *Mobile Networks and Applications*, v. 19, n. 2, p. 171-209.
- Cheriton, D. R. (2000). TRIAD: A new next-generation Internet architecture.
- Chiocchetti, R. R. (2013). ccnSim: an Highly Scalable CCN Simulator (2013). *In IEEE International Conference on Communications (ICC)*.
- Choi, N. G. (2012). In-network caching effect on optimal energy consumption in content-centric networking. *IEEE International Conference on Communications(ICC)*, pp. 2889-2894.
- Cisco, N. V. (2015). Forecast and Methodology, 2013–2018. Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf. Accessed at: fev.
- Edson Avelar, K. D. (2015). Armazenamento em Redes Orientadas a Conteúdo baseado. *Anais do SBRC*.
- Edwardes, M. &. (1994). Confidence intervals for a binomial proportion. *Statistics in medicine*, 13(16), 1693-1698.
- Farley, T. (2005). Mobile telephone history. *Telektronikk* 101.3/4.
- Foundation., N. S. (08 de 02 de 2017). *Named Data Networking (NDN) - A Future Internet Architecture*. Acesso em 08 de 02 de 2017, disponível em <https://named-data.net/>
- Ghodsi, A. S. (2011). Information-centric networking: seeing the forest for the trees. *In Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, pp. 1.

- Gladisch, A. D. (2014). Survey on mobility and multihoming in future Internet. *Wireless personal communications*, pp. 45-81.
- Gross, J. K. (2004). Cross-layer optimization of OFDM transmission systems for MPEG-4 video streaming. *Computer Communications*, 27(11), 1044-1055.
- Gundavelli, S. L. (2008). Proxy mobile ipv6 (No. RFC 5213).
- Han, D. L. (2014). Publisher mobility support in content centric networks. *IEEE International Conference on Information Networking*, pp. 214-219.
- Hermans, F. E. (2011). Mobile sources in an information-centric network with hierarchical names: An indirection approach. *7th SNCNW*.
- Hermans, F. N. (2012). Global source mobility in the content-centric networking architecture. In *Proceedings of the 1st ACM workshop on Emerging Name-Oriented Mobile Networking Design-Architecture, Algorithms, and Applications*, pp. 13-18.
- Hsu, H. &. (2008). Paired t test. *Wiley Encyclopedia of Clinical Trials*.
- Hu, B. Z. (2014). Provider mobility management based on domain proxies in content centric networks. *Communications and Information Technologies (ISCIT), 2014 14th International Symposium on. IEEE*.
- Izquierdo, L. R. (2006). Introduction to the formal analysis of social networks using mathematica. *University of California, Riverside*.
- J. Klaue, B. R. (2003). EvalVid - A Framework for Video Transmission and Quality Evaluation. *13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, , (pp. pp. 255-272). Illinois, USA.
- Jacobson, V. e. (2009). Networking named content. *Proceedings of the 5th international conference on Emerging networking experiments and technologies*.
- Jacobson, V. S. (2009). VoCCN: voice-over content-centric networks. In *Proceedings of the 2009 workshop on Re-architecting the Internet*, pp. 1-6.
- Jiang, X. B. (2012). A content provider mobility solution of named data networking. In *Network Protocols (ICNP)*.
- Jokela, P. N. (2004). Host identity protocol-extended abstract. *Wireless World Research Forum*.
- Jonathan B. Postel, C. A. (1981). The arpa Internet protocol. *Computer Networks*, pp.261–271.

- Khan, V. G. (1974). A protocol for packet network intercommunication. *IEEE Transactions on Communications*, 22:637–648.
- Kim H., K. J. (2012). Mobility support in content centric networks. In *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pp. 13-18.
- Kim, D. H. (2014). End-to-end mobility support in content centric networks. *International Journal of Communication Systems*.
- Kim, D. T.-X. (2015). Optimal Provider Mobility in Large-Scale Named-Data Networking. *TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS* , 4054-4071.
- Kleinrock 2002. (2002). Creating a mathematical theory of computer networks. *INFORMS, Operat. Res.* 50, pp. 125–131.
- Kleinrock, L. (1993). On the modeling and analysis of computer networks. *Proc. IEEE* 81 (8), pp. 1179–1191.
- Koponen, T. C. (2007). A data-oriented (and beyond) network architecture. In *ACM SIGCOMM Computer Communication Review*, Vol. 37, No. 4, pp. 181-192.
- Kumar, D. A. (2007). Globally optimal user-network association in an 802.11 WLAN & 3G UMTS hybrid cell. In *Managing Traffic Performance in Converged Networks*. Springer Berlin Heidelberg., pp. 1173-1187.
- Kurose, J. (2014). Information-centric networking: The evolution from circuits to packets to content. *Computer Networks* , v. 66, p. 112-120,.
- L. Berger, I. B. (2008). The OSPF opaque LSA. *RFC* 5250.
- L. Wang, O. W. (2013). MobiCCN: Mobility Support with Greedy Routing in Content-Centric Networks. *IEEE GLOBECOM*.
- Labiod, H. L. (2010). Analytical study of intradomain handover in multiple-mobile-routers-based multihomed NEMO networks. In *Computer Networks*, 3280–3294.
- Laoutaris, N. S. (2004). Meta algorithms for hierarchical web caches. *IEEE International Conference on Performance, Computing, and Communications*, pp. 445-452.
- Lara, A. K. (2014). Network innovation using openflow: A survey. *Communications Surveys & Tutorials, IEEE*, pp. 493-512.
- Law, E. L.-C. (2009). Understanding, scoping and defining user experience: a survey approach. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 719-728.

- Lee, J. S. (2012). Device mobility management in content-centric networking. *IEEE Communications Magazine*, pp. 28-34.
- Lee, K. W. (2010). Inter-domain handover scheme using an intermediate mobile access gateway for seamless service in vehicular networks. *International Journal of Communication Systems*, 1127-1144.
- Lehmann, M. M. (2016). Providing Producer Mobility Support in NDN Through Proactive Data Replication. *IEEE/IFIP Network Operations and Management Symposium, Istanbul*, pp. 383-391.
- Lilliefors, H. W. (1967). On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, pp. 399-402.
- Luo, H. a.-L. (2011). Quality of service provision in mobile multimedia-a survey. *Human-centric computing and information sciences*, v. 1, n. 1, p. 1-15.
- Luo, Y. E.-G. (2012). Mobility Support for Content Centric Networking: Case Study. In *Mobile Networks and Management*. Springer Berlin Heidelberg., pp. 76-89.
- Luo, Y. E.-G. (2014). Mobility support for content centric networking. *Telecommunication Systems*, pp. 1-18.
- McKeown, N. A. (2008). OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 69-74.
- Mosko, M. S. (2015). *CCNx 1.0 Protocol Architecture*.
- ndnSIM. (2016). NS-3 based Named Data Networking (NDN) simulator. Available at: <http://ndnsim.net/2.0/index.html>, Accessed at: mar/2016.
- ndnSIM. (2017). NS-3 based Named Data Networking (NDN) simulator. Disponivel em: <http://ndnsim.net/2.0/index.html>, Acessado em: mar/2017.
- Nguyen, X. N. (2013). Providing CCN functionalities over OpenFlow switches.
- Nikander, P. A. (2003). Mobile IP version 6 (MIPv6) route optimization security design. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, Vol. 3, pp. 2004-2008.
- Nikander, P. A. (2010). Host identity protocol (HIP): Connectivity, mobility, multi-homing, security, and privacy over IPv4 and IPv6 networks. *Communications Surveys & Tutorials*, pp. 186-204.
- NSF. (2017). *Future Internet Architecture Project*. Fonte: <http://www.nets-fia.net/>

- P. Jokela, A. Z. (2009). LIPSIN: Line Speed Publish/Subscribe Inter-networking. *Proc. ACM SIGCOMM, Barcelona, Spain*.
- Passarella, A. (2012). A survey on content-centric technologies for the current Internet: CDN and P2P solutions. *Computer Communications*, pp. 1-32.
- Perkins, C. E. (1997). Mobile ip. *IEEE Communications Magazine*, pp. 84-99.
- Podlipnig, S. &. (2003). A survey of web cache replacement strategies. *ACM Computing Surveys (CSUR)*, pp. 374-398.
- Psaras b, C. W. (2012). Probabilistic in-network caching for information-centric networks. *In Proceedings of the second edition of the ICN workshop on Information-centric networking. ACM*, pp. 55-60.
- Psaras, I. C. (2011). Modelling and evaluation of CCN-caching trees. *In NETWORKING Springer Berlin Heidelberg.*, pp. 78-91.
- Ramjee, R. L. (1999). IP micro-mobility support using HAWAII. *draft-ietf-mobileip-hawaii-00*.
- Rao, Y. D. (2014c). NLBA: A novel provider mobility support approach in mobile NDN environment. *Consumer Communications and Networking Conference (CCNC)*.
- Rao, Y. e. (2013). Proactive caching for enhancing user-side mobility support in named data networking. *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Seventh International Conference on. IEEE*.
- Rao, Y. G. (2014a). PBMA: A Proxy Based Mobility Support Approach in Named Data Networking. *In Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Tenth International Conference on*, pp. 938-941.
- Rao, Y. L. (2014b). LBMA: A novel Locator Based Mobility support Approach in Named Data Networking. *Communications, China, 11(4).*, pp. 111-120.
- Rao, Y. Z. (2013). Proactive caching for enhancing user-side mobility support in named data networking. *In Innovative Mobile and Internet Services in Ubiquitous Computing* , pp. 37-42.
- Ravindran, R. L. (2012). Supporting seamless mobility in named data networking. *IEEE International Conference Communications (ICC)*, pp. 5854-5869.

- Rayes, A. M. (2012). Internet of Things implications on ICN. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on, IEEE*, pp. 27-33.
- Rexford, J. &. (2010). Future Internet architecture: clean-slate versus evolutionary research. *Communications of the ACM*, pp. 36-40.
- RNP. (2016). Rede Nacional de Pesquisa/Arquitetura da Rede IPÊ. Disponível em: <http://www.rnp.br/servicos/conectividade/rede-ipe>. Acessado em: mar/2016.
- Ro, S. a. (2015). Inter-domain mobility support in Proxy Mobile IPv6 using overlap function of mobile access gateway. *Wireless Networks* , 899-910.
- Rowshanrad, S. e. (2014). A survey on SDN, the future of networking. *Journal of Advanced Computer Science & Technology*.
- Saha, D. M. (2004). Mobility support in IP: a survey of related protocols. *IEEE Network*, pp. 34-40.
- Saxena, D. R. (2016). Named Data Networking: A survey. *Computer Science Review*, pp. 15-55.
- Seufert, M. a.-G. (2015). A survey on quality of experience of HTTP adaptive streaming. *IEEE Communications Surveys & Tutorials*, 469-492.
- Sri, P. A. (2016). Big Data-Survey. *Indonesian Journal of Electrical Engineering and Informatics*, v. 4, n. 1.
- Sun, Q. e. (2014). SDN-based autonomic CCN traffic management. *Globecom Workshops (GC Wkshps)*.
- TalebiFard, P. L. (2015). Information-centric networking for VANETs. In *Vehicular ad hoc Networks. Springer International Publishing.*, pp. 503-524.
- TalebiFard, P. L. (2015). Information-centric networking for VANETs. *Springer International Publishing In Vehicular ad hoc Networks*, pp. 503-524.
- Tang, J. Z. (2014). A source mobility management scheme in content-centric networking. In *Consumer Communications and Networking Conference (CCNC)*, pp. 176-181.
- Tonneau, A.-S. N. (2014). A Survey on (mobile) wireless sensor network experimentation testbeds. *IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*.
- Tripathi, S. a. (2015). WLAN and UMTS QoS: A Survey. *Journal of Telecommunication, Switching Systems and Networks*, v.1, n.3, p.20-29.

- Trossen. (2011). Conceptual Architecture: Principles, Patterns and Sub-components Descriptions.
- University, A. S. (s.d.). *YUV Video Sequences*. Acesso em 2 de 1 de 2017, disponível em <http://trace.eas.asu.edu/yuv/>
- V. Jacobson, D. K. (2009). Networking Named Content. *Proc. ACM CoNEXT*, pp.1-12.
- Varga, A. (2017). *The OMNeT++ discrete event simulation system*. Fonte: <https://omnetpp.org/>
- Vetriselvan, V. P. (2014). Survey on the RIP, OSPF, EIGRP Routing Protocols. *International Journal of Computer Science and Information Technologies*, pp. 1058-1065.
- Wang, L. H. (2012). OSPFN: An OSPF based routing protocol for Named Data Networking. *University of Memphis and University of Arizona, Tech. Rep.*
- Wang, L. W. (2012). Data naming in vehicle-to-vehicle communications. In *Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 328-333.
- Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158), 209-212.
- Woo, T. P. (2014). Proactive neighbor pushing for enhancing provider mobility support in content-centric networking. *Sixth International Conf on Ubiquitous and Future Networks (ICUFN)*, pp. 158-163.
- Yao, H. F. (2016). An Optimal Routing Algorithm in Service Customized 5G Networks. *Mobile Information Systems*.
- Yaqub, M. A. (2016). Information-Centric Networks (ICN). *n Content-Centric Networks*. Springer Singapore., pp. 19-33.
- Zhang, G. L. (2013). Caching in information centric networking: a survey. *Computer Networks*, pp. 3128-3141.
- Zhang, L. e. (2014). Named data networking. *ACM SIGCOMM Computer Communication Review* , 66-73.
- Zhang, Y. H. (2014). Kite: A mobility support scheme for ndn. *Proceedings of the 1st international conference on Information-centric networking*. ACM.
- Zhou, Z. T. (2014). MobiNDN: A mobility support architecture for NDN. In *Control Conference (CCC), 2014 33rd Chinese* , pp. 5515-5520.