

**COMUNICAÇÃO DISPOSITIVO-A-DISPOSITIVO
ORIENTADA A ATRIBUTOS PARA
DESCARREGAMENTO DE DADOS EM REDES
CELULARES**

VINICIUS FONSÊCA E SILVA

**COMUNICAÇÃO DISPOSITIVO-A-DISPOSITIVO
ORIENTADA A ATRIBUTOS PARA
DESCARREGAMENTO DE DADOS EM REDES
CELULARES**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

ORIENTADOR: DANIEL FERNANDES MACEDO
COORIENTADOR: MARCELO DIAS DE AMORIM

Belo Horizonte

Março de 2021

© 2021, Vinicius Fonsêca e Silva.
Todos os direitos reservados

Silva, Vinicius Fonsêca e.

S586c Comunicação dispositivo-a-dispositivo orientada a atributos para descarregamento de dados em redes celulares [manuscrito] / Vinicius Fonsêca e Silva.- 2021. xxii, 104 f. il.

Orientador: Daniel Fernandes Macedo.

Coorientador: Marcelo Dias de Amorim

Tese (Doutorado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.

Referências: f.95-100

1. Computação – Teses. 2. Comunicação dispositivo-a-dispositivo – Teses. 3. Sistemas de comunicação sem fio. – Teses. 4. Telefonia celular – Teses I. Macedo, Daniel Fernandes. II. Amorim, Marcelo Dias de. III Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. IV. Título.

CDU 519.6*22(043)

Ficha catalográfica elaborada pela bibliotecária Belkiz Inez Rezende Costa CRB 6ª Região nº 1510



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE POS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

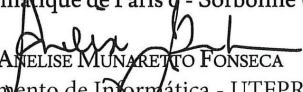
Comunicação Dispositivo-a-Dispositivo Orientada a Atributos para
Descarregamento de Dados em Redes Celulares

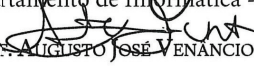
VINICIUS FONSÊCA E SILVA

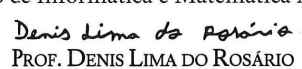
Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:


PROF. DANIEL FERNANDES MACEDO - Orientador
Departamento de Ciência da Computação - UFMG


PROF. MARCELO DIAS DE AMORIM - Coorientador
Laboratoire d'Informatique de Paris 6 - Sorbonne Université


PROFA. ANELISE MUNARETTO FONSECA
Departamento de Informática - UTFPR


PROF. AUGUSTO JOSÉ VENÂNCIO NETO
Departamento de Informática e Matemática Aplicada - UFRN


PROF. DENIS LIMA DO ROSÁRIO
Faculdade de Computação - UFPA


PROF. JOSÉ MARCOS SILVA NOGUEIRA
Departamento de Ciência da Computação - UFMG


PROF. VINICIUS FERNANDES SOARES MOTA
Departamento de Informática - UFES

Belo Horizonte, 26 de Março de 2021.

À minha tia Josiane, pelo apoio e confiança indispensáveis, constantes e incondicionais,

DEDICO

Agradecimentos

A Deus, que me iluminou e me protegeu durante toda essa etapa da minha vida.

A todos da minha família, em especial meus pais Josiara e Almir, minha irmã Fernanda e meu cunhado Thiago, minha tia Ana Cláudia e meu tio Mário Lúcio, pelo apoio oferecido sempre que precisei.

À Universidade Federal de Minas Gerais e ao Departamento de Ciência da Computação, pela oportunidade de realização do curso de Doutorado do Programa de Pós-Graduação deste departamento. Agradeço a todos os professores do departamento pelos conhecimentos adquiridos, os quais levarei pela vida toda.

Às agências de fomento CAPES, CNPq e Fapemig, bem como o projeto FUTEBOL, que foram indispensáveis durante toda a pesquisa, não somente do ponto de vista financeiro como também do provimento de recursos essenciais para o desenvolvimento deste trabalho.

A todos os meus amigos e colegas do Laboratório Winet, em especial os amigos Marcos Carvalho, Henrique Moura, Luis Cantelli, Erik Silva e Matheus Nunes, que acompanharam de perto o desenvolvimento deste trabalho e me ajudaram diversas vezes, principalmente por meio das várias conversas que tivemos ao longo da pesquisa. Sem essas conversas, inúmeros detalhes decisivos para o andamento do trabalho seriam quase impossíveis de serem solucionados.

Ao meu orientador, professor Daniel Macedo, e meu coorientador, professor Marcelo Dias de Amorim, que sempre me auxiliaram nos momentos de dificuldade. Agradeço pela sua paciência ao longo de toda a pesquisa, e acima de tudo, pelas constantes palavras de apoio e de certeza de que este trabalho se concretizaria com sucesso. Agradeço todo o conhecimento e conselhos passados por eles, os quais serão de grande valia durante toda a minha vida pessoal e profissional. Sem eles, esse trabalho não se concretizaria.

A todos aqueles que não citei aqui, mas que diretamente ou indiretamente contribuíram de alguma forma para a execução deste trabalho.

A todos, os meus sinceros agradecimentos.

*“Forte é quem, depois de tanto perder,
reergue-se e segue lutando.”*
(Ayrton Senna)

Resumo

O tráfego de dados de redes celulares mostra um crescimento exponencial ano após ano. O descarregamento oportunístico de dados móveis ameniza o problema movendo o tráfego para outras redes, podendo ser empregada para tal a comunicação Dispositivo-a-Dispositivo (*Device-To-Device* - D2D). Dessa forma, usuários que requisitam o mesmo dado podem compartilhá-lo entre si por meio de um enlace direto, eliminando a necessidade de múltiplas requisições na rede celular. Historicamente, grande parte das propostas de infraestruturas D2D/celular não considera a relação entre a mobilidade dos dispositivos e o conteúdo transmitido. Outra desvantagem é que tais propostas dependem de soluções de pré-carregamento de conteúdo. Isso faz com que o *cache* dos dispositivos receba conteúdo que pode não ser de interesse deles, mas somente de seus vizinhos, dessa forma aumentando o tráfego na rede celular. Visando aumentar o emprego de comunicação D2D oportunística e do *cache*, e com isso reduzir o tráfego na rede celular, o presente trabalho propõe dois modelos de prioridade para substituição de conteúdo no *cache* dos dispositivos, que computam valores de prioridade para cada conteúdo armazenado. A substituição tem como base um monitoramento da mobilidade de cada dispositivo, bem como do perfil do conteúdo requisitado por cada um. Tal processo ocorre de forma distribuída e independente por cada dispositivo, ocorrendo sempre que ele recebe um novo conteúdo e seu respectivo *cache* está sobrecarregado. Os modelos propostos consideram apenas os conteúdos que o respectivo dispositivo consumiu e tem armazenado no *cache*, eliminando o pré-carregamento de conteúdo. O primeiro modelo, denominado *Modelo de Prioridade de Propósito Geral* (MP-PG), considera cenários em que o conteúdo é indivisível e de qualidade fixa. Já o segundo modelo, denominado *Modelo de Prioridade Ciente de Qualidade* (MP-CQ), foca em cenários em que o conteúdo se divide em segmentos (*chunks*) de qualidade variável (por exemplo, vídeo adaptativo). O MP-CQ ainda otimiza a qualidade do conteúdo recebido pelos dispositivos. A avaliação do desempenho do MP-PG e do MP-CQ ocorreu por meio de simulações, variando-se parâmetros como o número de conteúdos

e usuários na rede. Os modelos foram comparados com uma solução clássica que remove os conteúdos mais antigos, desconsiderando a relação conjunta entre atributos de mobilidade e conteúdo. Os modelos também foram comparados com uma solução do estado da arte que considera esses atributos sob uma abordagem distinta. Os resultados comprovam a eficiência dos modelos em termos do emprego de comunicação D2D e do *cache*, além do custo sobre as mensagens de controle enviadas.

Palavras-chave: Redes D2D, Redes Celulares, Descarregamento de Dados.

Abstract

Data traffic from cellular networks shows an exponential growth year after year. *Opportunistic mobile data offloading* alleviates the problem by moving traffic to other networks, and Device-To-Device (D2D) communications can be used for such a purpose. In this way, users that request the same data can share it among themselves through a direct link, thus eliminating the need of multiple requests in the cellular network. Historically, most proposals of a D2D/cellular infrastructure do not consider the tradeoff between device mobility and the transmitted content. Another drawback is that these proposals rely on content preloading. This forces the device to cache content that might not be of interest of the respective user, but only of his neighbors, increasing the traffic on the network. To increase the use of opportunistic D2D communications and the device cache, this way reducing the traffic on the cellular network, this work proposes two priority models for cache replacement, computing priority values for the stored content. The cache replacement procedure is based on monitored data related to the mobility of each device, as well as their respective content download profile. Such a procedure is distributed among the devices, occurring every time they receive new content, and their respective cache is overloaded. The proposed models consider only content consumed and stored by each device, eliminating content preloading. The first model, called *General Purpose Priority Model* (MP-PG), considers scenarios with generic content of fixed quality that cannot be fragmented. The second model, called *Quality-Aware Priority Model* (MP-CQ), considers content splittable into segments (or chunks) of variable quality (e.g., adaptive video). MP-CQ also improves the content quality perceived by each user. The performance evaluation of MP-PG and MP-CQ occurred through simulations, varying parameters like the amount of content and users in the network. Both models were compared with a classic solution that removes the oldest content, hence not considering the joint relationship between mobility and content attributes. The proposed models were also compared with a state-of-the-art solution. Results prove the effectiveness of both models in terms of the use of D2D communications

and the cache, as well as the cost related to control messages.

Keywords: D2D Networks, Cellular Networks, Data Offloading.

Lista de Figuras

1.1	Exemplo do emprego de <i>offloading</i> D2D em uma rede celular.	2
2.1	Comunicação D2D <i>Inband</i> e <i>Outband</i> (Adaptado de Asadi et al. [2014]). . .	10
2.2	Tipos de Comunicação D2D (Adaptado de Shah et al. [2018]).	11
2.3	Tipos de UEs em Comunicação PS-D2D.	11
2.4	Sistema D2D/celular proposto por Pyattaev et al. [2015].	17
2.5	Componentes de um Sistema que emprega DASH (Adaptado de Kua et al. [2017])	19
4.1	MR-PG.	31
4.2	MR-CQ.	33
5.1	Variando o Intervalo de Monitoramento.	57
5.2	Variando os Pesos das EWMA's.	58
5.3	Variando o Tempo de Busca via D2D para o <i>Trace</i> de Contato <i>Infocom06</i>	59
5.4	Variando o Tempo de Busca via D2D para o <i>Trace</i> de Contato <i>Reality Mining</i>	59
5.5	Variando o Tempo de Busca via D2D para o Modelo de Mobilidade Sintético.	60
5.6	Variando o Número de Arquivos para o <i>Trace</i> de Contato <i>Infocom06</i>	61
5.7	Variando o Número de Arquivos para o <i>Trace</i> de Contato <i>Reality Mining</i>	61
5.8	Variando o Número de Arquivos para o Modelo de Mobilidade Sintético.	62
5.9	Número de Arquivos Baixados por Meio de Comunicação D2D, do Servidor e do <i>Cache</i> durante uma Execução do MP-PG.	63
5.10	Variando o Número de Usuários para o Modelo de Mobilidade Sintético.	64
5.11	Sobrecarga de Mensagens de Controle com a Variação do Número de Arquivos.	66
5.12	Sobrecarga de Mensagens de Controle com a Variação do Número de Usuários.	68
5.13	Sobrecarga de Bytes de Controle com a Variação do Número de Arquivos.	69
5.14	Sobrecarga de Bytes de Controle com a Variação do Número de Usuários.	70
5.15	Variando o Modelo e o Número de Arquivos para o <i>Trace</i> de Contato <i>Infocom06</i>	73

5.16 Variando o Modelo e o Número de Arquivos para o <i>Trace</i> de Contato <i>Reality Mining</i>	73
5.17 Variando o Modelo e o Número de Arquivos para o Modelo de Mobilidade Sintético.	74
5.18 Variando o Modelo e o Número de Usuários para o Modelo de Mobilidade Sintético.	76
5.19 Taxa de Acerto Global de <i>Chunks</i> com a Variação do Número de Arquivos de Vídeo para o <i>Trace</i> de Contato <i>Reality Mining</i>	78
5.20 Taxa de Acerto Global de <i>Chunks</i> com a Variação do Número de Arquivos de Vídeo para o Modelo de Mobilidade Sintético.	78
5.21 Taxa de Acerto Global de <i>Chunks</i> com a Variação do Número de Arquivos de Vídeo para o <i>Trace</i> de Contato <i>Infocom06</i>	79
5.22 Taxa de Acerto Global de <i>Chunks</i> com a Variação do Número de Usuários para o Modelo de Mobilidade Sintético.	80
5.23 Taxa de Qualidade de <i>Chunks</i> com a Variação do Número de Arquivos de Vídeo para o <i>Trace</i> de Contato <i>Infocom06</i>	82
5.24 Taxa de Qualidade de <i>Chunks</i> com a Variação do Número de Arquivos de Vídeo para o <i>Trace</i> de Contato <i>Reality Mining</i>	82
5.25 Taxa de Qualidade de <i>Chunks</i> com a Variação do Número de Arquivos de Vídeo para o Modelo de Mobilidade Sintético.	83
5.26 Taxa de Qualidade de <i>Chunks</i> com a Variação do Número de Usuários para o Modelo de Mobilidade Sintético.	84
5.27 Sobrecarga de Mensagens de Controle com a Variação do Número de Arquivos de Vídeo para o Modelo de Mobilidade Sintético.	85
5.28 Sobrecarga de Mensagens de Controle com a Variação do Número de Usuários para o Modelo de Mobilidade Sintético.	86
5.29 Sobrecarga de Bytes de Controle com a Variação do Número de Arquivos de Vídeo para o Modelo de Mobilidade Sintético.	87
5.30 Sobrecarga de Bytes de Controle com a Variação do Número de Usuários para o Modelo de Mobilidade Sintético.	88

Lista de Tabelas

3.1	Comparação dos Trabalhos Apresentados com a Presente Proposta.	26
5.1	Parâmetros de Simulação.	44

Sumário

Agradecimentos	ix
Resumo	xiii
Abstract	xv
Lista de Figuras	xvii
Lista de Tabelas	xix
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	3
1.3 Contribuições	4
1.4 Organização do Texto	5
2 Referencial Teórico	7
2.1 Terminologia Básica	7
2.2 Comunicação Dispositivo-a-Dispositivo Oportunística em Redes Celulares	9
2.2.1 Comunicação D2D <i>Inband</i> e <i>Outband</i> : Vantagens e Desvantagens	12
2.3 Gerenciamento de <i>Cache</i> em Redes D2D/celular	13
2.4 Sistemas D2D/celular para Realização de <i>Caching</i> e <i>Offloading</i>	15
2.5 Transmissão de Vídeo Adaptativo por Meio de DASH	17
2.6 Conclusão	19
3 Revisão Bibliográfica	21
3.1 Gerenciamento de Conteúdo em <i>Cache</i> para Aplicações em Geral	21
3.2 Gerenciamento de Conteúdo em <i>Cache</i> para Aplicações de Vídeo	23
3.3 Comparação dos Trabalhos com a Presente Proposta	26
3.4 Conclusão	27

4	Substituição de Conteúdo em <i>Cache</i> baseada em Atributos de Mobilidade e Conteúdo	29
4.1	Modelos de Rede	29
4.1.1	Características Comuns ao MR-PG e ao MR-CQ	30
4.1.2	Características Específicas do MR-PG	31
4.1.3	Características Específicas do MR-CQ	32
4.2	Modelos de Prioridade para Substituição de Conteúdo no <i>Cache</i> dos Dispositivos	34
4.2.1	Modelo de Prioridade de Propósito Geral	34
4.2.2	Modelo de Prioridade Ciente de Qualidade	35
4.2.3	Calculando a Popularidade de um Arquivo em um Dispositivo	39
4.3	Conclusão	40
5	Avaliação dos Modelos de Prioridade	41
5.1	Preparação do Ambiente de Simulação	41
5.1.1	<i>Datasets</i> de Conteúdo e Mobilidade	42
5.1.2	<i>Baselines</i>	43
5.1.3	Parâmetros de Simulação	44
5.1.4	Métricas de Desempenho	48
5.1.5	Mensagens de Controle	49
5.2	Resultados	55
5.2.1	MP-PG	55
5.2.2	MP-CQ	77
5.3	Conclusão	88
6	Conclusões	91
6.1	Trabalhos Futuros	92
	Referências Bibliográficas	95
	Apêndice A Lista de Siglas	101
	Apêndice B Tabela de Variáveis Empregadas nos Modelos de Rede e de Prioridade	103

Capítulo 1

Introdução

Segundo um relatório anual recente feito pela Cisco, estima-se que, até 2023, mais de 70% da população mundial (cerca de 5.7 bilhões de pessoas) terá conectividade com uma rede celular (2G, 3G, 4G ou 5G), e 66% da população mundial (cerca de 5.3 bilhões de pessoas) terá acesso à Internet. Além disso, haverá cerca de 30 bilhões de dispositivos/conexões, sendo que 45% desses serão móveis, e cerca de 47% serão capazes de reproduzir vídeo [Cisco, 2020].

Ainda que as próximas gerações de redes sem fio sejam projetadas visando suprir o crescimento na demanda por tráfego e serviços, há o constante risco de que essa demanda não seja atendida em determinadas regiões, devido à grande variação na quantidade de dispositivos conectados a um mesmo ponto de acesso. Uma das redes sem fio que mais sofre com essa variação são as redes celulares, sendo que a alta demanda dos usuários gera prejuízos na qualidade dos serviços oferecidos, causando insatisfações [Schwarz & Rupp, 2016]. Um exemplo de serviço bastante utilizado em redes celulares atuais é o *streaming* de vídeo adaptativo por meio de DASH (*Dynamic Adaptive Streaming over HTTP*) [Stockhammer, 2011], que pode ter as imagens ou o áudio prejudicado até mesmo ao ponto de impossibilitar sua visualização [Zheng et al., 2015; Duanmu et al., 2017].

1.1 Motivação

O problema descrito acima, bem como tantos outros observados nas redes móveis nos últimos anos, fez com que a indústria e a academia buscassem soluções para suportar a crescente demanda. Uma solução para amenizar o problema apresentado é mover o tráfego de dados das redes celulares para outras interfaces de comunicação, ou seja, para fora da infraestrutura de rede principal. Esta estratégia é conhecida como

descarregamento oportunístico de dados das redes móveis (*opportunistic mobile data offloading*), a qual é uma técnica bastante explorada na literatura [Rebecchi et al., 2015].

O tráfego redirecionado por meio de *offloading* oportunístico passa a ocorrer em outra rede, a qual pode utilizar ou não a mesma tecnologia da rede original [Asadi et al., 2014]. As técnicas de descarregamento de dados das redes celulares variam do uso de femtocélulas (células domésticas com funcionalidades semelhantes às de uma célula tradicional das redes celulares, porém com alcance e capacidade de tráfego limitados) [Haldar et al., 2013] às redes Wi-Fi públicas [Mota et al., 2013] e, mais recentemente, a comunicação Dispositivo-a-Dispositivo (*Device-To-Device* - D2D) [Doppler et al., 2009]. O emprego de *offloading* por meio de comunicação D2D permite que diferentes usuários requisitando o mesmo conteúdo possam compartilhá-lo entre si, dispensando a necessidade de várias requisições na rede infraestruturada, conforme exemplificado na Figura 1.1.

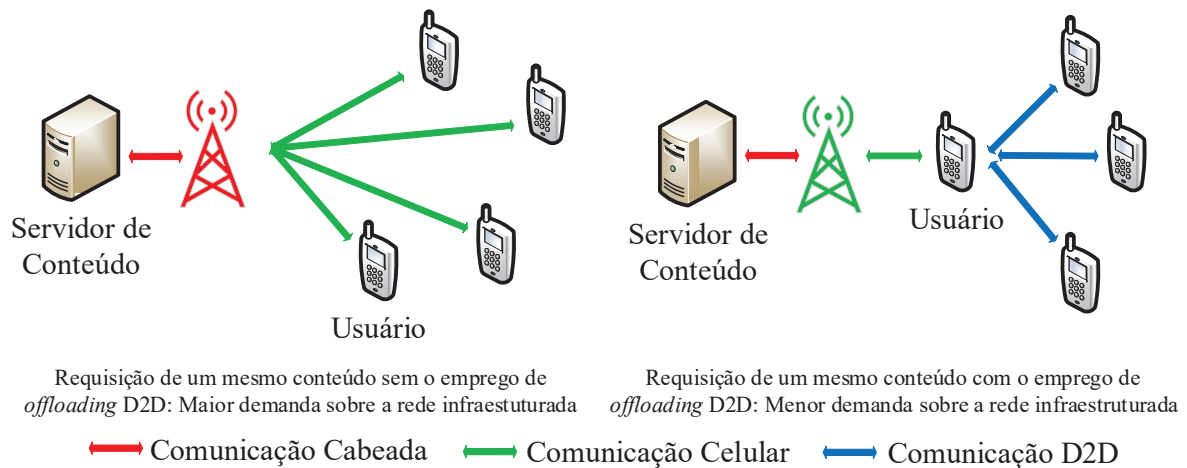


Figura 1.1. Exemplo do emprego de *offloading* D2D em uma rede celular.

Sem sombra de dúvida, o uso de uma rede híbrida D2D/celular, que permite que os dispositivos troquem informações com seus vizinhos e a operadora celular simultaneamente, já é um tema abertamente explorado pela literatura. Diferentes trabalhos propõem o uso de D2D como técnica de *offloading* em redes celulares [Asadi et al., 2014; Shah et al., 2018].

Nesse contexto, visando o compartilhamento de conteúdo entre os dispositivos, é necessário que o conteúdo seja armazenado em *cache* durante um tempo pré-determinado. O conteúdo pode ser pré-carregado no *cache* dos dispositivos, dessa forma antecipando a demanda dos usuários [Li et al., 2018].

Com o passar do tempo e a chegada de novos conteúdos nos dispositivos, há de se considerar que o *cache* desses pode ficar sobrecarregado. Isso exige que sejam tomadas decisões sobre quais conteúdos serão mantidos no *cache* e quais conteúdos serão descartados. Tais decisões visam aumentar a utilidade dos conteúdos armazenados, aumentando o emprego de comunicação D2D oportunística e do próprio *cache*. Dessa forma, há uma redução na demanda sobre a operadora.

1.2 Objetivos

Com base na contextualização acima, este trabalho visa atacar um dos principais problemas relacionados, que é o de gerenciamento do *cache* dos dispositivos. Mais especificamente, deverão ser definidas as diretivas para substituição de conteúdo no *cache* quando este estiver sobrecarregado. Com isso, define-se o seguinte objetivo geral:

Objetivo Geral: Reduzir a demanda sobre a operadora com o aumento do emprego de comunicação D2D oportunística e do *cache*, através de operações de monitoramento e gerenciamento do *cache* de baixo custo a cada entidade da rede.

Visando atingir o objetivo geral apresentado acima, o primeiro ponto a se observar é com relação às informações a serem monitoradas. Os atributos ligados à mobilidade permitem verificar quais conteúdos um dispositivo é capaz de transmitir por meio de comunicação D2D. Tal verificação pode ocorrer por meio do monitoramento dos contatos entre dispositivos. No entanto, conteúdos selecionados podem ser também de baixa popularidade na região monitorada, o que não contribui para o aumento do emprego de comunicação D2D e do *cache*.

Por outro lado, atributos ligados às características do conteúdo transmitido permitiriam selecionar conteúdos populares. A expressão da popularidade, por sua vez, poderia se dar em termos do número de requisições de cada conteúdo, por exemplo. No entanto, avaliar somente esse fator também não é suficiente, uma vez que o tamanho dos conteúdos populares poderia inviabilizar sua transmissão via comunicação D2D. Com base nas afirmações acima, o primeiro objetivo específico do presente trabalho é:

Objetivo Específico 1: Realizar análises dos conteúdos com base na relação conjunta entre atributos de mobilidade dos dispositivos e características do conteúdo transmitido.

Poucos trabalhos na literatura levam em consideração a relação conjunta entre esses atributos, e há de se considerar que eles mudam de forma localizada e ao longo do tempo. Outra deficiência da literatura é a escassez de trabalhos no contexto de redes híbridas D2D/celular que também focam em melhorar a qualidade do conteúdo transmitido, considerando aplicações em que o mesmo pode se dividir em segmentos (ou *chunks*). Exemplos de aplicações nesse sentido são as que realizam *streaming* de vídeo adaptativo por meio de DASH [Stockhammer, 2011]. Uma das formas de se melhorar a qualidade do conteúdo nessas aplicações é por meio da entrega de uma maior quantidade de *chunks* de maior resolução, o que contribui para o aumento da experiência do usuário. Com base nessa premissa, o segundo objetivo específico do presente trabalho é:

Objetivo Específico 2: Melhorar a qualidade do conteúdo transmitido, aliando essa melhoria ao aumento do emprego de comunicação D2D e do *cache*.

1.3 Contribuições

Visando atingir os objetivos específicos apresentados na Seção 1.2, esta seção define e destaca as duas principais contribuições deste trabalho.

A primeira contribuição envolve o objetivo específico 1 enunciado na Seção 1.2, focando em conteúdo não segmentado e cuja qualidade não varia ao longo do tempo:

Contribuição 1: Criação de um modelo que compute a prioridade de conteúdos no *cache* dos dispositivos.

O modelo enunciado na contribuição 1 é denominado *Modelo de Prioridade de Propósito Geral* (MP-PG). O MP-PG atribui valores de prioridade para cada conteúdo, computando valores maiores para conteúdos pequenos e populares ao mesmo tempo, ou seja, conteúdos que o dispositivo possa compartilhar por meio de comunicação D2D com sucesso ou ler do *cache* com maior frequência. Computa-se a popularidade com base em observações a nível global e local, ou seja, considerando todos os dispositivos e o raio de comunicação D2D onde cada um se encontra.

Apesar de atender a diferentes tipos de conteúdo, o MP-PG se limita a conteúdos não segmentados e cuja qualidade não varia ao longo do tempo, conforme mencionado anteriormente. Em cenários onde a qualidade varia, é necessário incluí-la como métrica de otimização ao substituir conteúdo no *cache*. Inspirando-se no MP-PG, define-se aqui a segunda contribuição do presente trabalho, que envolve não somente o objetivo específico 1 enunciado na Seção 1.2, mas também o objetivo específico 2, relacionado à qualidade do conteúdo:

Contribuição 2: Criação de um modelo que compute a prioridade de conteúdos representados em qualidade variada no *cache* dos dispositivos.

O modelo enunciado na contribuição 2 é denominado *Modelo de Prioridade Ciente de Qualidade* (MP-CQ). O MP-CQ foca em conteúdo que pode se dividir em segmentos (*chunks*) que o servidor e os dispositivos podem fornecer com qualidade variada. O MP-CQ visa principalmente aumentar a qualidade do conteúdo fornecido, ao mesmo tempo mantendo-se um alto emprego de comunicação D2D e do *cache*.

Os modelos de prioridade enunciados nas contribuições 1 e 2 permitem definir quais conteúdos deverão se manter no *cache* de cada dispositivo, de forma a reduzir a demanda sobre a operadora. Com isso, aumenta-se o emprego de comunicação D2D oportunística e do *cache*, em alguns casos aumentando inclusive a qualidade do conteúdo consumido pelos dispositivos.

O monitoramento das informações empregadas pelo MP-PG e pelo MP-CQ ocorre de forma distribuída entre o servidor de conteúdo e os dispositivos, reduzindo assim a sobrecarga de controle (*overhead*) sobre a rede. Além disso, os dispositivos compartilham conteúdo de forma independente via comunicação D2D com seus vizinhos. Dessa forma, desobriga-se a operadora de tomar todas as decisões, evitando sua sobrecarga.

É importante observar que as operações de substituição de conteúdo ocorrem somente quando um dispositivo recebe um novo conteúdo, e seu respectivo *cache* está sobrecarregado. Além disso, tais operações consideram apenas o que cada usuário consumiu ao longo do tempo, dispensando dessa forma a necessidade de soluções de pré-carregamento de conteúdo em *cache*, como feito tradicionalmente na literatura. É importante também ressaltar que o emprego de soluções de pré-carregamento faz com que os dispositivos estejam sujeitos a receber conteúdo que não seja de interesse deles, mas apenas de seus vizinhos. Conseqüentemente, há uma tendência de aumento na demanda sobre a rede celular.

1.4 Organização do Texto

O restante deste trabalho está organizado da seguinte forma: O Capítulo 2 contextualiza os conceitos empregados durante toda a pesquisa. O Capítulo 3 apresenta os trabalhos relacionados que levaram à presente proposta. O Capítulo 4 descreve em mais detalhes os modelos de prioridade desenvolvidos. O Capítulo 5 avalia os modelos de prioridade e discute os resultados obtidos. Por fim, o Capítulo 6 faz as considerações finais e conclui este trabalho.

Capítulo 2

Referencial Teórico

Este capítulo apresenta os conceitos empregados neste trabalho. A Seção 2.1 apresenta a terminologia básica. Considerando que o presente trabalho emprega comunicação D2D oportunística para realização de *offloading*, a Seção 2.2 faz uma descrição sobre a comunicação D2D aliada às redes celulares, com uma classificação detalhada dos tipos de comunicação, bem como as vantagens e desvantagens de cada um.

Visando lidar com o problema de gerenciamento do *cache* dos dispositivos, a Seção 2.3 descreve técnicas de *caching* em redes D2D/celular, e a Seção 2.4 destaca exemplos de sistemas da literatura que realizam *caching* e *offloading*. Como a avaliação considera o *streaming* de vídeo adaptativo por meio de DASH (*Dynamic Adaptive Streaming over HTTP*), a Seção 2.5 apresenta uma descrição detalhada a respeito dessa aplicação. Por fim, a Seção 2.6 conclui este capítulo.

2.1 Terminologia Básica

Esta seção apresenta algumas definições básicas empregadas na tese [Peterson & Davie, 2007; Tanenbaum & Wetherall, 2010]. Tais definições permitem deixar mais claras e mais bem definidas as funções de cada entidade da rede relacionada às soluções propostas.

- **Programa de Aplicação (ou “Aplicação” somente ou “Aplicativo”)**: Um programa executado pelo usuário (Ex.: YouTube) e oferecido por uma determinada empresa (Ex.: Google). Cada aplicação apresenta requisitos de desempenho específicos, devido à forma como foi desenvolvida pela respectiva empresa. Um dos inúmeros serviços oferecidos pelas aplicações aos seus usuários

é o *streaming de vídeo*, empregado por programas como o *YouTube* e o *Dailymotion*.

- **Protocolo de Aplicação:** Um conjunto de funções e procedimentos desempenhados e empregados por uma ou mais aplicações, segundo um padrão especificado por uma organização (Ex.: IEEE, ISO). Um exemplo de protocolo é o HTTP (*Hypertext Transfer Protocol*), que aplicações como o Mozilla Firefox e o Google Chrome empregam. Cada protocolo tem um direcionamento a um ou mais serviços, dependendo de sua complexidade.
- **Serviço:** Do ponto de vista do gerenciamento das entidades que compõem uma infraestrutura de rede, um serviço define um conjunto de objetivos a serem alcançados. Em outras palavras, cada serviço define um conjunto de tarefas que devem ocorrer na rede (Ex.: Autenticação, NAT, compressão de dados, roteamento etc.). Do ponto de vista do usuário, um serviço engloba um conjunto de aplicações com o mesmo objetivo, como por exemplo, as aplicações que realizam *streaming* de vídeo mencionadas acima.
- **Servidor de Aplicação (ou Servidor de Conteúdo):** Entidade da rede que compõe uma aplicação, sendo uma determinada empresa (Ex.: Google, Microsoft) responsável pelo seu desenvolvimento e gerenciamento. O servidor de aplicação, em suma, recebe requisições dos usuários e envia o conteúdo requisitado.
- **Qualidade de conteúdo:** Característica que define o quão bem um conteúdo está representado em qualquer entidade da rede. Por exemplo, a qualidade de um arquivo de vídeo pode ser em termos da sua resolução, ou seja, quantos pontos (*pixels*) compõem cada quadro de vídeo. Quanto maior a resolução, maior será a qualidade do respectivo vídeo, ou seja, melhor será a sua representação. A qualidade do conteúdo é um dos fatores que impactam diretamente na Qualidade de Experiência (*Quality of Experience - QoE*) do usuário [Oyman & Singh, 2012].
- **Segmentos de conteúdo (ou *chunks*):** São partes de um conteúdo de tamanho e qualidade fixos. No caso de arquivos de áudio e vídeo, cada *chunk* tem uma duração pré-definida (e.g. um segundo).

2.2 Comunicação Dispositivo-a-Dispositivo Oportunística em Redes Celulares

No contexto de redes oportunísticas, ao se fazer a entrega de um conteúdo, assume-se que não existe um caminho pré-definido entre os dispositivos de origem e destino, e que um repasse desse conteúdo pode ocorrer quando dois nós se encontram. Tais redes caracterizam-se por uma topologia que se altera de forma dinâmica, criando períodos de desconexão longos e frequentes entre os usuários. Duas das principais redes contidas nesse contexto são as *Pocket Switched Networks* (PSNs) [Hui et al., 2005], compostas por dispositivos móveis carregados por pessoas, e as redes tolerantes a atrasos e interrupções (*Delay and Disruption Tolerant Networks* - DTNs) [Burleigh et al., 2003], sendo essa última voltada principalmente para comunicação interplanetária, dotada de longos atrasos e longos períodos de interrupção [Mota et al., 2014].

É nesse mesmo contexto que surgiu o conceito de *comunicação Dispositivo-a-Dispositivo* (*Device-To-Device* - D2D), permitindo que dois dispositivos compartilhem entre si conteúdos de interesse comum, dispensando a necessidade de múltiplas requisições na rede infraestruturada. O padrão 3GPP *Device-To-Device Proximity Services* (D2D ProSe [Lin et al., 2014]) define como comunicação D2D a comunicação direta entre dispositivos próximos, com mínima ou nenhuma intervenção da estação base [Asadi et al., 2014]. O D2D ProSe permite utilizar redes celulares como tecnologia de comunicação, dessa forma, tanto a rede celular quanto a rede D2D utilizam a mesma faixa de frequência e o mesmo recurso de rádio para transmitir conteúdo, definindo assim a comunicação D2D *inband*. Por outro lado, o D2D ProSe também pode explorar o espectro não licenciado (a comunicação D2D *outband*). A comunicação *outband* poderia utilizar, por exemplo, a tecnologia Bluetooth, o Wi-Fi em modo *ad-hoc* ou o Wi-Fi Direct.

Asadi et al. [2014] dividem a comunicação *inband* entre dois sub-tipos. No primeiro, as comunicações D2D e celular compartilham os mesmos recursos de hardware e a mesma faixa de frequência (*underlay*), portanto havendo disputa pelo uso dos recursos. Já no segundo o compartilhamento persiste, porém a divisão de recursos é pré-definida (*overlay*), sendo que cada tipo de comunicação possui intervalos de tempo dedicados. Dessa forma, não há disputa pelo recurso entre as comunicações D2D e celular.

Para a comunicação D2D *outband*, Asadi et al. [2014] também fazem uma divisão entre duas vertentes: controlada e autônoma. Na comunicação D2D *outband*

controlada, existe uma entidade centralizada pertencente à rede celular que gerencia toda comunicação D2D. Já a comunicação D2D *outband* autônoma não possui tal entidade, permitindo que os dispositivos criem redes D2D sem a necessidade de uma interação prévia com a rede celular.

A Figura 2.1 esquematiza a comunicação D2D *inband* e *outband*, mostrando o emprego dos recursos de rádio ao longo do tempo.

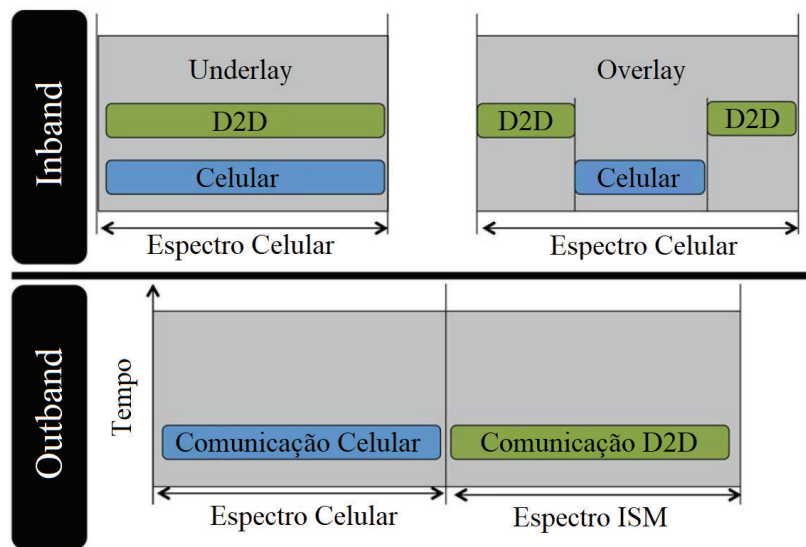


Figura 2.1. Comunicação D2D *Inband* e *Outband* (Adaptado de Asadi et al. [2014]).

Posteriormente, Shah et al. [2018] inseriu o agrupamento de tipos de comunicação D2D de Asadi et al. [2014] em um único tipo geral, denominado *Commercial D2D*. Nesse mesmo nível, os autores definem um novo ramo de classes, denominado *Public Safety D2D* (PS-D2D), no qual se insere a comunicação D2D voltada para segurança pública. O 3GPP realizou a padronização da comunicação PS-D2D em 2015, em sua *release* de número 12, com estudos a respeito dos cenários, requisitos, e a melhor tecnologia empregada [3GPP, 2015]. Esse tipo de comunicação reúne dispositivos que estão fora do alcance da rede celular, em locais onde não existe infraestrutura favorável ou em cenários emergenciais onde a infraestrutura local está danificada. Na comunicação PS-D2D, um dispositivo pode se conectar com vizinhos próximos, que funcionam como *relays* e possuem melhor conectividade com a rede celular.

De forma semelhante à comunicação D2D *outband* apresentada por Asadi et al. [2014], divide-se a comunicação PS-D2D entre *controlada* e *autônoma*, e a comunicação D2D *inband* entre *centralizada* e *distribuída*.

A Figura 2.2 mostra uma nova divisão dos diferentes tipos de comunicação D2D, também considerando a visão dada por Shah et al. [2018].

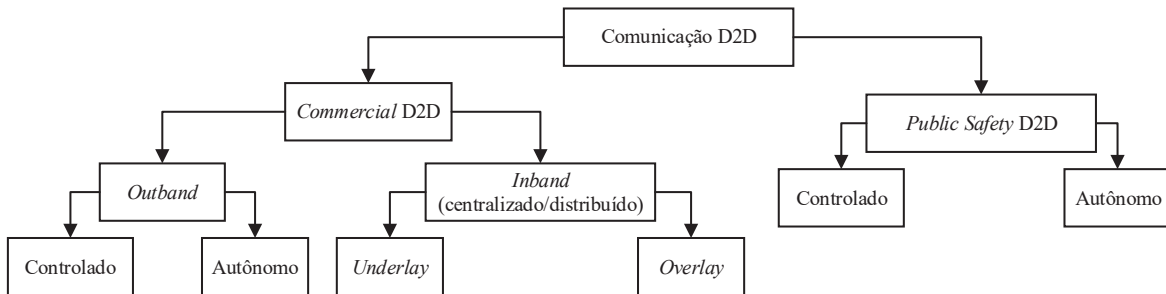


Figura 2.2. Tipos de Comunicação D2D (Adaptado de Shah et al. [2018]).

Em comunicações PS-D2D, dividem-se os UEs (*User Equipments*, ou seja, os dispositivos de usuário, tais como *smartphones*, *tablets* etc.) em três classes distintas, conforme mostrado na Figura 2.3:

- UE Celular (*Cellular UE* – CUE), diretamente conectados à rede celular;
- UE Isolado (*Isolated UE* – IUE), fora do alcance da rede celular e por isso não está conectado à infraestrutura via interface celular;
- UE *Relay* (*Relay UE* – RUE), conectado à infraestrutura e está no raio de alcance dos IUEs, servindo como intermediário para os IUEs enviarem dados à rede celular, sendo que haverá um enlace D2D entre um IUE e um RUE.

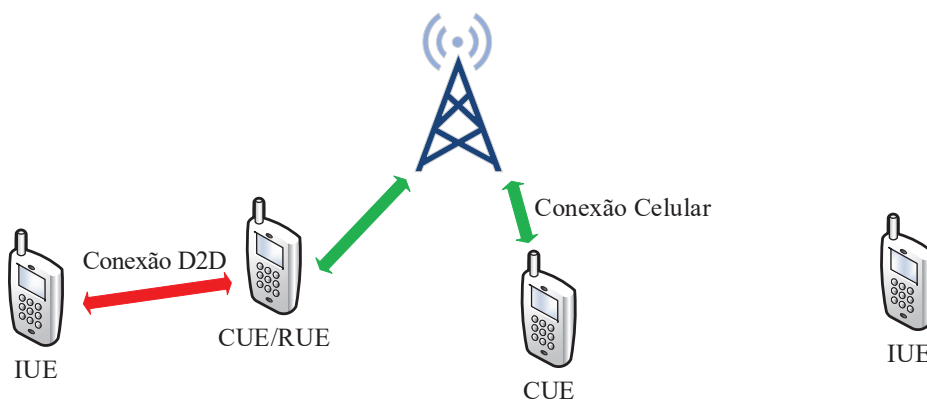


Figura 2.3. Tipos de UEs em Comunicação PS-D2D.

2.2.1 Comunicação D2D *Inband* e *Outband*: Vantagens e Desvantagens

Tendo em vista que, para o presente trabalho será empregada a comunicação D2D no contexto de redes celulares, esta seção apresenta as vantagens e desvantagens entre os tipos de comunicação D2D *inband* e *outband*.

2.2.1.1 Comunicação *Inband*

Segundo Asadi et al. [2014], na comunicação D2D *inband underlay*, a principal vantagem é o aproveitamento otimizado do espectro juntamente com a comunicação celular, quando esta última não estiver fazendo uso do espectro. Por outro lado, há a imposição de vários desafios devido à escassez de recursos de hardware, o que leva a uma “disputa” entre o modo de comunicação D2D e o modo de comunicação celular. Há também o risco de interferência entre dispositivos realizando comunicação D2D e outros realizando comunicação celular no mesmo intervalo de tempo, uma vez que todos utilizam a mesma faixa de frequência. A interferência acaba por levar a soluções complexas de controle dos recursos, o que poderia inviabilizar sua implementação dependendo da tecnologia empregada.

Por fim, considerando a comunicação D2D *inband overlay*, existe também o risco de alocação ineficiente dos recursos para cada modo de comunicação (D2D e celular), o que pode causar um desperdício desses recursos e diminuir a eficiência de uso do espectro.

2.2.1.2 Comunicação *Outband*

Considerando a comunicação D2D *outband*, uma das vantagens é a possibilidade de uso simultâneo das duas interfaces de comunicação (D2D e celular), o que aumenta a capacidade da rede em relação à comunicação D2D *inband*. No entanto, o problema da interferência se mantém, pois a rede D2D emprega faixas de frequência que outras redes concorrentes também empregam. Uma das faixas de frequência mais usadas na comunicação D2D *outband* são as faixas ISM (*Industrial, Scientific and Medical*), uma vez que não precisam de uma licença prévia para uso. O principal problema das faixas ISM é o fato de que inúmeras outras tecnologias operam nessa mesma faixa (Ex.: Wi-Fi 802.11, Bluetooth, ZigBee etc.), e por isso podem prejudicar a comunicação D2D ou vice-versa.

Outro grande desafio em comunicação D2D *outband* é o consumo de energia, uma vez que o dispositivo emprega interfaces de comunicação adicionais e de forma paralela

à interface celular, portanto exigindo mais recursos de hardware para operar. Isso poderia inviabilizar o emprego desse modo de comunicação em um cenário comercial com grande quantidade de dispositivos. Por fim, outra questão enfrentada é o uso eficiente das interfaces D2D e celular, o que leva à necessidade de se criar mecanismos que permitam definir quais interfaces serão empregadas em cada instante.

2.3 Gerenciamento de *Cache* em Redes D2D/celular

Diversos serviços para redes D2D/celular, tanto do ponto de vista do seu gerenciamento como do usuário, têm sido apresentados na literatura nos últimos anos [Asadi et al., 2014; Liu et al., 2015; Shah et al., 2018].

Alguns dos serviços apresentados são: a) Comunicação entre dispositivos em situações de emergência ou desastre; b) *Caching* de conteúdo; c) Armazenamento distribuído; d) Aumento da cobertura de rede celular; e) Comunicação local em grupo; f) Serviços localizados; g) Descarregamento de dados oportunístico (*offloading*); h) Jogos *multiplayer* para grupos de usuários próximos entre si; i) Serviços para redes veiculares; j) Serviços de saúde (*eHealth*); k) Comunicação em *smart grids*; l) Serviços para cidades inteligentes (*smart cities*); e m) Marketing baseado em localização.

Dentre os serviços acima, dois dos mais abordados na literatura é o *caching* de conteúdo e o descarregamento de dados oportunístico (*offloading*). O emprego de comunicação D2D para *caching* e *offloading* em redes celulares tem sido proposto desde o período em que apenas a comunicação D2D *inband* era empregada, em especial no contexto de redes oportunísticas [Doppler et al., 2009].

Nesse contexto, pesquisadores da área de arquitetura de computadores e provedores de serviço têm feito estudos amplos focando nos problemas ligados ao gerenciamento de *cache*. Existem atualmente diferentes técnicas clássicas de substituição de conteúdo em *cache* (*cache replacement*) de fácil implementação, tais como o *Menos Frequentemente Usado* (*Least Frequently Used - LFU*), o *Menos Recentemente Usado* (*Least Recently Used - LRU*), e o *LRS* (*Least Recently Stored*).

O *LRS* é uma técnica gulosa que funciona da seguinte forma: quando o dispositivo faz o *download* de um novo arquivo, ele tenta armazená-lo em seu respectivo *cache*. Caso o espaço de armazenamento disponível não seja suficiente, o dispositivo remove o arquivo com o tempo de armazenamento mais antigo. O dispositivo repete este passo até que o espaço liberado seja suficiente para o novo arquivo.

Apesar das técnicas acima serem de fácil implementação, é necessário considerar que o armazenamento de conteúdo em *cache* visando aumentar o emprego de comunicação D2D oportunística e do próprio *cache* traz novos desafios. Tais desafios envolvem a mobilidade dos dispositivos, o interesse do usuário em cada conteúdo e o armazenamento limitado [Wang et al., 2017].

De fato, a mobilidade tem um papel dual em comunicação D2D, já que ao mesmo tempo em que pode causar tempos de conexão mais curtos entre dispositivos, ela pode prover conteúdo em *cache* para mais dispositivos. Diferente de soluções de armazenamento em *cache* para Web baseadas na popularidade do conteúdo, a comunicação D2D oportunística deve considerar a popularidade em conjunto com outros atributos de conteúdo e as características dos contatos entre os dispositivos. Por exemplo, em uma rede caracterizada por tempos de contato mais curtos, conteúdos populares porém grandes podem ser inviáveis de se transmitir entre os dispositivos. As técnicas clássicas citadas acima ignoram as características de mobilidade, popularidade e outros atributos de conteúdo.

Na direção oposta, a literatura propõe técnicas do estado da arte de substituição de conteúdo que consideram atributos inerentes à mobilidade dos dispositivos e do conteúdo trafegado, porém sob uma abordagem distinta da que é adotada pelo presente trabalho. Uma dessas propostas é chamada SACC (*Social-Aware Cooperative Caching*) [Wu et al., 2020]. O SACC emprega uma técnica de substituição de conteúdo em *cache* que computa uma métrica de popularidade para cada arquivo.

O cálculo da popularidade ocorre em duas etapas e sempre que um dispositivo recebe um novo arquivo e o *cache* não possui espaço suficiente para ele, ou seja, está sobrecarregado. Na primeira etapa, reúnem-se os valores computados para a habilidade de cada dispositivo da rede em compartilhar arquivos com cada um dos vizinhos com quem ele teve contato. Cada dispositivo calcula localmente as habilidades de compartilhamento para cada vizinho. Eles tomam como base métricas sociais computadas entre eles ao longo do tempo, como a força do laço social, a confiabilidade e a probabilidade de encontros. Além da habilidade de compartilhamento, na primeira etapa também reúnem-se informações sobre os arquivos contidos no *cache* de cada dispositivo da rede.

Na segunda etapa, computa-se a popularidade para cada arquivo contido no *cache* do dispositivo que está realizando a substituição de conteúdo. O cálculo da popularidade toma como base as informações reunidas na primeira etapa, sendo que para cada arquivo avaliado, combina-se a habilidade de compartilhamento entre dispositivos que possuem o arquivo e dispositivos que não o possuem. Por fim, o dispositivo que está realizando a substituição de conteúdo descarta os arquivos do

cache com base nos valores de popularidade resultantes, do menos para o mais popular, até que a sobrecarga do *cache* deixe de ocorrer.

De forma paralela às técnicas de substituição de conteúdo, a maioria das soluções que habilitam armazenamento em *cache* em comunicação D2D também movem conteúdo de servidores centralizados para o *cache* de dispositivos selecionados, e são denominadas *estratégias de pré-carregamento de conteúdo em cache (cache placement)* [Li et al., 2018]. As soluções propostas baseiam-se em *clusters*, onde servidores centralizados agrupam dispositivos móveis e definem os centros de *cluster* como *cache* para outros membros do respectivo *cluster*. Por outro lado, uma série de algoritmos buscam o conteúdo em dispositivos vizinhos, e são denominados como *estratégias de entrega de conteúdo armazenado em cache (cache delivery)*. Nesses casos, cada dispositivo define quais conteúdos serão mantidos em *cache* [Li et al., 2018]. Leitores interessados podem se referir a Wang et al. [2017] e a Li et al. [2018] para uma discussão mais aprofundada a respeito de políticas de armazenamento em *cache* móvel em redes D2D.

2.4 Sistemas D2D/celular para Realização de *Caching* e *Offloading*

Nesta seção, destacamos dois sistemas D2D/celular para a realização de *caching* e *offloading*, propostos na literatura por Bao et al. [2013] e Pyattaev et al. [Pyattaev et al., 2013a,b; Andreev et al., 2014; Pyattaev et al., 2014; Andreev et al., 2015; Pyattaev et al., 2015]. Em ambos, o tipo de comunicação D2D empregado é o *outband*, descrito na Seção 2.2. Destacamos esses sistemas pela sua proximidade com o objetivo geral do presente trabalho, que é o aumento do emprego de comunicação D2D e do *cache*, conforme descrito na Seção 1.2. Tal proximidade permite simplificar a integração desses sistemas com os modelos de prioridade propostos, contribuindo inclusive na maximização do desempenho dos modelos.

Bao et al. [2013] desenvolveram um protótipo denominado *DataSpotting*, no qual a operadora móvel coordena o estabelecimento de grupos D2D em função da localização e do conteúdo requisitado. Aqui, consideram-se cenários em que a concentração de dispositivos é alta, e dispositivos que estão na mesma região se comunicam empregando Wi-Fi no modo *ad-hoc*. A determinação de quais dispositivos irão se conectar se baseia na localização e no conteúdo armazenado por cada dispositivo, informações essas que a operadora armazena ao longo do tempo. Experimentos realizados com *smartphones* Android mostraram que a largura de banda entre dispositivos próximos (distância

menor que 3 metros) pode chegar a 2,5 MB/s e diminuir para 0,5 MB/s entre dispositivos com distância superior a 25 metros.

No *survey* elaborado por Shah et al. [2018], um dos principais conjuntos de trabalho citado é o desenvolvido por Pyattaev et al., cujo cenário de foco é uma infraestrutura celular integrada a uma interface de comunicação D2D *outband* via Wi-Fi Direct [Pyattaev et al., 2013a,b; Andreev et al., 2014; Pyattaev et al., 2014; Andreev et al., 2015; Pyattaev et al., 2015]. Nesses trabalhos, os autores avaliaram o benefício trazido pela comunicação D2D *outband* como técnica de *offloading*, sob aspectos como a capacidade geral da rede, bem como a eficiência de energia do sistema como um todo. Os resultados obtidos culminaram em um protótipo, desenvolvido com hardware real [Pyattaev et al., 2014, 2015]. Os autores avaliaram os gargalos da comunicação D2D, definindo limites de desempenho e identificando as limitações da tecnologia. Os autores empregaram uma pilha LTE completa com hardware real.

Uma das principais entidades pertencentes ao sistema proposto pelos autores acima é denominada *D2D Server*, a qual possui várias informações sobre conexões em andamento e localização dos dispositivos, além de mediar a formação de grupos D2D. Para cada aplicação em execução, existe a interação do *D2D Server* com o respectivo servidor de aplicação (*Application Server*) via rede cabeada, ou seja, fora do espectro licenciado pela rede celular. Esse servidor possui uma base de todo conteúdo disponibilizado para compartilhamento por seus dispositivos. A disponibilização se faz por meio de mensagens de publicações (*publish*) ao *Application Server*, informando a disponibilidade de um conteúdo antes requisitado pelo dispositivo relacionado. Quando outro dispositivo deseja fazer uma requisição do mesmo conteúdo, realiza-se uma busca no *Application Server*, o qual informa ao *D2D Server* a disponibilidade local daquele conteúdo. A partir daí, com base em uma estimativa local de distância entre os dispositivos, o *D2D Server* coordena a conexão entre eles via D2D para realizar o compartilhamento.

A Figura 2.4 mostra um esquemático do sistema proposto pelos autores.

Após a formação de um grupo D2D, a operadora faz requisições periódicas de vazão e outras métricas de QoS para cada dispositivo envolvido, visando melhorar a coexistência dos grupos D2D e outras redes que utilizam a mesma faixa de espectro [Pyattaev et al., 2014, 2015].

Um ponto negativo encontrado com frequência em sistemas D2D/celular é a ausência de uma preocupação desses quanto às características operacionais de um enlace D2D, ao se compartilhar um conteúdo específico. Os modelos de prioridade propostos no presente trabalho consideram, dentre outros atributos, a mobilidade dos

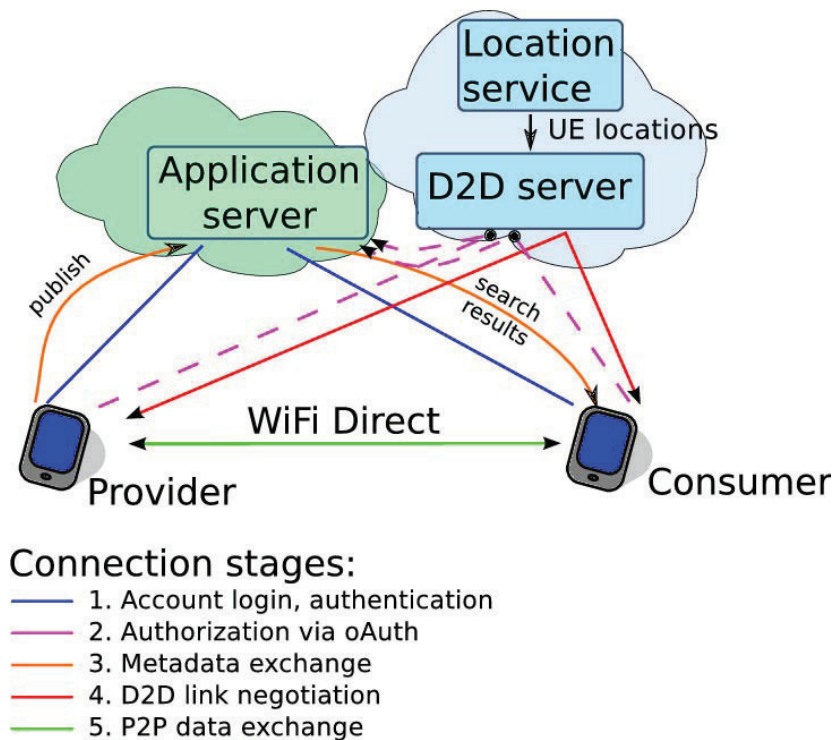


Figura 2.4. Sistema D2D/celular proposto por Pyattaev et al. [2015].

dispositivos, de forma a determinar se um conteúdo é passível de transmissão por meio de comunicação D2D por determinado dispositivo, e por isso deve se manter armazenado em *cache*.

Uma vez que esses sistemas em geral focam na criação e gerenciamento de grupos D2D, e os modelos de prioridade propostos neste trabalho (Seção 1.3) focam no problema de gerenciamento do *cache* dos dispositivos, ambos podem ser empregados de forma conjunta, contribuindo para um aumento ainda maior do desempenho em termos do emprego de comunicação D2D oportunística e do *cache*.

2.5 Transmissão de Vídeo Adaptativo por Meio de DASH

Com a convergência de uma vasta gama de serviços multimídia para a transmissão de seu conteúdo por meio de redes móveis, houve um aumento na demanda por tráfego na rede [Bouazizi et al., 2010]. Essa demanda tende a aumentar ainda mais nos próximos anos, em termos do número de usuários e do tamanho do conteúdo trafegado [Cisco, 2020].

O aumento constante da capacidade e da vazão de redes móveis, por meio da

inserção no mercado de novas tecnologias e protocolos de comunicação, nem sempre condiz com a demanda estabelecida ao longo do tempo. Com isso, a operadora pode ficar sobrecarregada em determinados momentos, impedindo que alguns usuários recebam o conteúdo desejado [Schwarz & Rupp, 2016]. Até mesmo as gerações mais recentes de redes celulares ainda encaram problemas de longa data, tais como interferências de sinal entre os dispositivos e os pontos de acesso da operadora, por exemplo [Gupta & Jha, 2015].

Os desafios acima fizeram com que a indústria e a academia desenvolvessem técnicas que garantissem a entrega do conteúdo a todos os usuários que o solicitassem, ainda que em qualidade inferior em determinados momentos. O *Moving Picture Experts Group* (MPEG) foi uma das entidades envolvidas nessa tarefa, dando assim origem ao DASH (*Dynamic Adaptive Streaming over HTTP*) [Stockhammer, 2011; Sodagar, 2011; Kua et al., 2017].

O DASH, como o próprio nome indica, transmite conteúdo multimídia por meio de requisições HTTP. O servidor envia partes consecutivas do conteúdo por meio de requisições sequenciais enviadas pelo dispositivo, até que ele obtenha o conteúdo completo. Em paralelo às requisições, o dispositivo monitora as condições da rede. Dessa forma, caso a qualidade da rede caia, o dispositivo requisita as partes subsequentes em qualidade inferior. Essa técnica permite manter uma reprodução do conteúdo sem interrupções, evitando degradações na Qualidade de Experiência (*Quality of Experience - QoE*). De forma contrária, melhorias na qualidade da rede permitem que partes do conteúdo em maior qualidade sejam requisitadas.

Um sistema DASH possui os seguintes componentes, conforme esquematizado na Figura 2.5:

- Um servidor HTTP que possui um mesmo conteúdo multimídia com representações/qualidades distintas, dividindo cada representação em segmentos (*chunks*);
- Uma descrição das representações, enviada pelo servidor ao usuário (o cliente DASH), visando orientá-lo nas requisições de segmentos;
- Um monitor do enlace entre o usuário e o servidor HTTP, que irá tomar decisões sobre qual representação requisitar ao longo do tempo;
- Um cliente HTTP que envia as requisições de segmentos ao servidor;
- Um reprodutor de conteúdo multimídia que decodifica e apresenta o vídeo ao usuário.

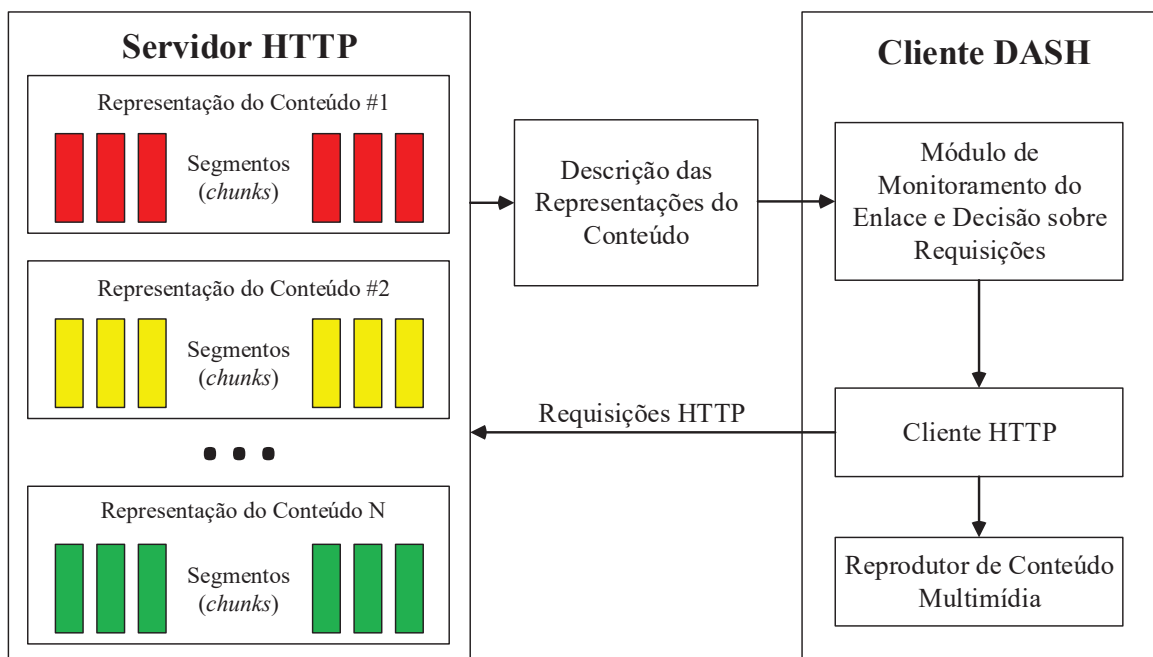


Figura 2.5. Componentes de um Sistema que emprega DASH (Adaptado de Kua et al. [2017])

2.6 Conclusão

Este capítulo apresentou os principais conceitos que contextualizam a tese. Inicialmente foi apresentada a terminologia básica. Em seguida, foram apresentados os conceitos inerentes à comunicação D2D integrada às redes celulares, com suas vantagens e desvantagens. Tendo em vista que a capacidade de hardware, bem como as opções de interface de rede em um mesmo dispositivo aumentaram nos últimos anos, a literatura começou a ter maior interesse por soluções envolvendo comunicação D2D *outband*. Por essa razão, o presente trabalho visa seguir nessa mesma direção.

Além de apresentar os conceitos inerentes às redes D2D e redes celulares, foi dado foco especial a um dos principais serviços ligados a essas redes, que é o *caching* de conteúdo, e a uma das principais aplicações empregadas, que é a transmissão de vídeo adaptativo por meio de DASH.

Capítulo 3

Revisão Bibliográfica

Este capítulo reúne os principais trabalhos em torno da presente proposta. A busca por trabalhos tomou como base o principal problema atacado pelo presente trabalho, que é o gerenciamento do *cache* dos dispositivos.

Destacam-se neste capítulo trabalhos que propõem técnicas de pré-carregamento (*placement*) e substituição (*replacement*) de conteúdo em *cache*, considerando conteúdos inteiros ou divididos em segmentos (*chunks*), bem como conteúdos disponibilizados em diferentes qualidades. Foram empregadas diferentes combinações dos seguintes termos de busca: *D2D*, *cache*, *placement*, *replacement*, *dash*, *video*, *chunks*, *mobility* e *popularity*.

Os trabalhos dividem-se em duas partes. A primeira parte apresenta técnicas de gerenciamento de conteúdo em *cache* voltadas para aplicações em geral (Seção 3.1). Já a segunda parte apresenta técnicas de gerenciamento de conteúdo em *cache* voltadas para aplicações de vídeo (Seção 3.2). Por fim, a Seção 3.3 faz uma comparação de todos os trabalhos apresentados com a presente proposta.

3.1 Gerenciamento de Conteúdo em *Cache* para Aplicações em Geral

A mobilidade em comunicação D2D permite uma maior cobertura da rede, porém pode causar contatos curtos entre dispositivos [Mota et al., 2014]. Como resultado, encaminhar conteúdos maiores pode se tornar inviável. Nesta seção, destacam-se trabalhos recentes que também consideram as características de mobilidade e o perfil de *download* de conteúdo de dispositivos vizinhos, propondo o que aqui são denominadas *políticas de armazenamento em cache cientes socialmente e cientes de*

mobilidade (social and mobility-aware caching policies) [Hosny et al., 2016; Song & Choi, 2019; Shan & Zhu, 2019; Yuan et al., 2019; Zhang et al., 2019; Ali et al., 2020; Wu et al., 2020; Vasilakos et al., 2016]. Esses trabalhos propõem políticas para todos os tipos de conteúdo, entretanto, elas não consideram conteúdo provido em diferentes qualidades.

Hosny et al. [2016] consideram que usuários com o mesmo padrão de mobilidade devem armazenar o mesmo conteúdo em *cache* e mostram que usuários com mobilidade aleatória são eficientes em armazenar mais conteúdo na rede. Song & Choi [2019] exploram a mobilidade dos usuários para modelar o pré-carregamento de conteúdo nos dispositivos dos usuários na forma de um problema de programação inteira (*integer programming*). Shan *et al.* consideram a mobilidade e a estrutura social dos dispositivos para propor uma estratégia de armazenamento em *cache* baseando-se em *clusters* de usuários com os mesmos interesses [Shan & Zhu, 2019; Yuan et al., 2019]. Zhang et al. [2019] consideram a mobilidade, a estrutura social e a popularidade do conteúdo. Os autores propõem a previsão de interesse do usuário para definir o conteúdo no *cache* dos dispositivos.

De forma similar, Ali et al. [2020] consideram o perfil de *download* dos usuários e grupos de mobilidade desses para otimizar o *cache* dos dispositivos. A solução dos autores monitora a mobilidade dos usuários para definir um perfil de mobilidade para cada um, definindo grupos de usuários com perfis similares. Wu et al. [2020] consideram as relações sociais entre usuários para propor um modelo de armazenamento de conteúdo em *cache* baseado em comunidade. Ao se formar comunidades de usuários com interesses similares, os autores melhoram a precisão da estimação da popularidade do conteúdo, otimizando o processo de armazenamento em *cache* e com isso aumentando o uso de comunicação D2D oportunística.

A presente proposta considera modelos para substituição de conteúdo em *cache* após exceder sua capacidade, dessa forma aumentando as chances de se prover um conteúdo por meio de comunicação D2D oportunística. A maioria dos trabalhos apresentados acima dispensam substituição de conteúdo em *cache* ou empregam algoritmos clássicos quando o *cache* está cheio. Nesse caso, há a probabilidade de se descartar de um dispositivo conteúdos potencialmente compartilháveis com seus vizinhos.

Portanto, é importante decidir, com a chegada de novos conteúdos maiores que o espaço disponível em *cache*, quais deles o dispositivo deve manter armazenado em *cache* ou descartar. A solução proposta por Wu et al. [2020] é um passo nessa direção, propondo uma nova estratégia de substituição de conteúdo em *cache* que estima a popularidade baseando-se apenas em informações coletadas pelos dispositivos em sua

respectiva comunidade, bem como entre comunidades.

Vasilakos et al. [2016] propõem um modelo de pré-carregamento de conteúdo em *cache* baseado em mobilidade que foca em conteúdos menos populares ou personalizados. Os autores também desenvolvem um esquema baseado na popularidade do conteúdo como uma extensão do modelo original. Eles também propõem soluções de substituição de conteúdo em *cache*, baseadas em uma fórmula de ganho em duas versões distintas: Uma considerando mobilidade, e outra considerando tanto mobilidade quanto popularidade de conteúdo. A principal desvantagem da proposta é o fato de que não se considera a qualidade do conteúdo.

Chandrasekaran et al. [2018] propõem a mobilidade do dispositivo como um serviço (*as a service*) para distribuir conteúdo no *cache* dos dispositivos. Eles selecionam um subconjunto de dispositivos como ajudantes (*helpers*) baseado na sua respectiva mobilidade, de forma a armazenar conteúdo em *cache* e distribuí-lo para seus vizinhos por meio de comunicação D2D. A política de substituição de conteúdo em *cache* é uma adaptação do clássico *Menos Frequentemente Usado* (*Least Frequently Used* - LFU). Apesar da proposta dos autores considerar tanto a mobilidade do dispositivo quanto a popularidade do conteúdo, ela falha em não considerar cenários com conteúdo fornecido em qualidades distintas. Além disso, a proposta dos autores centraliza todas as decisões no Servidor de Borda (*Edge Server*), um fato que pode eventualmente sobrecarregar a rede principal. Por fim, ao se adotar uma abordagem de seleção de *helpers*, a proposta dos autores tende a limitar a variabilidade de conteúdo na rede.

3.2 Gerenciamento de Conteúdo em *Cache* para Aplicações de Vídeo

Uma vez que a presente proposta também foca em melhorar a qualidade do conteúdo, é razoável considerar que uma das aplicações que poderão se beneficiar é o *streaming de vídeo adaptativo baseado em chunks*, onde os vídeos dividem-se em segmentos (*chunks*) de qualidades distintas, e o servidor de conteúdo envia *chunks* aos usuários de acordo com a sua respectiva qualidade de conexão ao longo do tempo (Ex.: Abordagens baseadas em DASH – *Dynamic Adaptive Streaming over HTTP* [Stockhammer, 2011]).

Baseando-se nas afirmações acima, nesta seção destacam-se trabalhos recentes que propõem soluções de pré-carregamento e substituição de conteúdo em *cache* (*cache placement/replacement*) considerando aplicações de vídeo [Li et al., 2018; Wang et al.,

2019; Ge et al., 2016; Mehrabi et al., 2019a,b; Mehr & Medhi, 2019; Ye et al., 2017; Baccour et al., 2020].

Li et al. [2018] propõem um mecanismo de pré-carregamento e substituição de conteúdo em *cache* que centraliza todas as decisões no servidor *Edge*, que se baseia na topologia da rede e estatísticas de requisição de forma a armazenar o conteúdo mais popular no *cache*. A proposta dos autores faz pré-carregamento por meio de compartilhamento oportunístico de conteúdo entre dispositivos vizinhos, bem como substituição de conteúdo por meio de soluções clássicas tais como FIFO (*First In First Out*), LRU (*Least Recently Used*) ou baseadas na popularidade do conteúdo. A proposta dos autores também considera o conteúdo dividido em *chunks*, mas não considera a variabilidade da qualidade.

Wang et al. [2019] consideram as relações sociais entre usuários para propor uma abordagem de pré-busca de vídeo (*video prefetching*) para pré-carregamento de conteúdo em *cache*. A proposta dos autores seleciona vídeos considerando sua respectiva popularidade e privacidade. O modelo é atualizado periodicamente e independente das previsões realizadas. Operações de substituição de conteúdo em *cache* também ocorrem por meio do clássico LRU, e os dispositivos compartilham informações de monitoramento por meio de comunicação D2D. Da mesma forma que os trabalhos anteriores, a proposta dos autores não considera a qualidade na modelagem da pré-busca de vídeo. Além disso, eles não dão foco especial em substituição de conteúdo em *cache*, um fato que impacta diretamente na taxa de acerto no *cache*.

Ge et al. [2016] focam em armazenamento em *cache* com o emprego de DASH e o conceito de *Mobile Edge Computing* (MEC) para redes 5G, concentrando todas as decisões no servidor *Edge*. Os autores propõem um mecanismo de pré-carregamento/substituição de *chunks* de vídeo no *cache* visando selecionar os melhores *chunks* de vídeo, e avaliam sua proposta em um protótipo baseado em uma rede celular LTE-A (*Long Term Evolution Advanced*). A principal desvantagem da proposta dos autores é o fato de que o *cache* localiza-se apenas no servidor *Edge*, o que impede os dispositivos de selecionarem o próprio conteúdo com base nas suas observações locais. Além disso, a proposta dos autores não emprega comunicação D2D oportunística como uma forma de compartilhamento de conteúdo, além de não considerar a mobilidade dos dispositivos na modelagem da solução.

Mehrabi et al. [2019a] propõem um mecanismo de pré-carregamento de conteúdo em *cache* para aplicações de *streaming* de vídeo que distribui conteúdo de vídeo entre o servidor *Edge* e os dispositivos, tomando decisões e armazenando conteúdo em sua maior parte no *cache* do servidor *Edge*. A solução maximiza a qualidade de experiência

(*Quality of Experience* - QoE), criando *chunks* de qualidade inferior com base em *chunks* de qualidade superior (uma técnica denominada *transrating*). Uma vez que o foco principal é o fator de QoE, a solução dos autores não considera as características do conteúdo nem a mobilidade dos dispositivos, e a substituição de conteúdo em *cache* ocorre por meio de soluções clássicas baseadas no LRU, com ou sem comunicação D2D.

Mehrabi et al. estendem sua proposta anterior ao contexto de *Green Mobile Edge computing* (*Green MEC*), onde a principal diferença para o MEC tradicional é a inclusão de fontes de energia intermitentes e renováveis [Mehrabi et al., 2019b]. Em seu trabalho, os autores focam não apenas em otimizar o QoE, mas também em otimizar o consumo de energia e o tráfego no *backhaul*.

Visando melhorar o QoE por meio de aplicações de *streaming* de vídeo baseadas em DASH, Mehr & Medhi [2019] propõem um *framework* denominado *Smart Cache*, que se baseia em previsões de vazão de forma a fazer pré-carregamento de vídeo no *cache* dos dispositivos. Para a tarefa de previsão, o *framework* proposto se baseia em valores prévios de vazão coletados pelos dispositivos ao longo do tempo. Os autores avaliaram o *framework* proposto em um testbed, demonstrando a queda no número de pré-carregamentos de *cache* não utilizados. Entretanto, os autores não empregam D2D como uma opção para compartilhamento de vídeo, além de não considerar atributos de mobilidade e conteúdo em sua modelagem.

Ye et al. [2017] focam em aplicações baseadas em DASH e adotam o conceito de *Scalable Video Coding* (SVC) visando armazenar vídeos de qualidades distintas no *cache* do servidor *Edge*. Com o emprego do SVC, cada vídeo tem uma camada base e uma ou mais camadas de melhoria, o que pode sofrer ajustes para melhorar (ou reduzir) a qualidade do vídeo de acordo com as condições do canal celular e a popularidade do vídeo. As operações de substituição de conteúdo em *cache* também ocorrem no servidor *Edge* por meio da adaptação do LRU e do LFU para aplicar o fator de qualidade, e com isso os autores propõem o MRQ (*Most Recent Quality*) e o MFQ (*Most Frequent Quality*) para substituição de conteúdo no *cache*. Como se pode observar, uma das principais desvantagens da proposta dos autores é a centralização do *cache* no servidor *Edge*, o que limita a variabilidade de conteúdo oferecido na rede. Além disso, os autores não consideram o emprego de comunicação D2D.

Baccour et al. [2020] propõem um mecanismo de pré-carregamento/substituição de conteúdo em *cache* que seleciona *chunks* com maior probabilidade de consumo. De forma similar aos trabalhos anteriores, a proposta dos autores centraliza o *cache* e todas as decisões no servidor *Edge*. O servidor *Edge* armazena o mesmo *chunk* em qualidades distintas, replicando-os em servidores *Edge* vizinhos. A proposta dos autores realiza substituição de conteúdo em *cache* ao se priorizar *chunks* com maior

probabilidade de consumo, bem como *chunks* com maior taxa de bits/qualidade. A principal desvantagem é a ausência do D2D como canal de comunicação.

3.3 Comparação dos Trabalhos com a Presente Proposta

A Tabela 3.1 apresenta um sumário das principais características dos trabalhos deste capítulo, incluindo a presente proposta. Para fins de simplificação, a seguir definem-se acrônimos para cada característica considerada:

- EPC - Estratégia de Pré-carregamento de Conteúdo;
- BM - Baseado em Mobilidade;
- BDC - Baseado em Dados de Conteúdo;
- BQ - Baseado em Qualidade;
- NESC - Nova Estratégia de Substituição de Conteúdo.

Tabela 3.1. Comparação dos Trabalhos Apresentados com a Presente Proposta.

Trabalho	EPC	BM	BDC	BQ	NESC
Hosny et al. [2016]	✓	✓	✗	✗	✗
Song & Choi [2019]	✓	✓	✗	✗	✗
Yuan et al. [2019]	✓	✓	✗	✗	✗
Shan & Zhu [2019]	✓	✓	✗	✗	✗
Zhang et al. [2019]	✓	✓	✓	✗	✗
Ali et al. [2020]	✓	✓	✓	✗	✗
Wu et al. [2020]	✓	✓	✓	✗	✓
Vasilakos et al. [2016]	✓	✓	✓	✗	✓
Chandrasekaran et al. [2018]	✓	✓	✓	✗	✗
Li et al. [2018]	✓	✓	✓	✗	✗
Wang et al. [2019]	✓	✓	✓	✗	✗
Ge et al. [2016]	✓	✗	✓	✓	✗
Mehrabi et al. [2019a]	✓	✗	✗	✓	✗
Mehrabi et al. [2019b]	✓	✗	✗	✓	✗
Mehr & Medhi [2019]	✓	✗	✗	✓	✗
Ye et al. [2017]	✓	✗	✓	✓	✓
Baccour et al. [2020]	✓	✗	✓	✓	✓
Proposta	✗	✓	✓	✓	✓

Como pode ser visto, e até onde se sabe, o presente trabalho é o único que propõe uma solução de gerenciamento de *cache* que dispensa técnicas de pré-carregamento de conteúdo (*cache placement*), considerando apenas conteúdos já consumidos por cada dispositivo.

Conforme mencionado anteriormente, as técnicas de pré-carregamento de conteúdo armazenam conteúdo no *cache* que pode não ser de interesse do respectivo dispositivo, mas apenas de seus vizinhos, causando assim uma tendência de aumento no tráfego sobre a rede celular. Além disso, o emprego dessas técnicas aumenta a probabilidade de receber conteúdos que não serão consumidos por nenhum dispositivo, uma vez que elas estão sujeitas a erros nas previsões acerca do que os usuários irão consumir em intervalos futuros. Por fim, é importante observar que tais estratégias requerem decisões centralizadas, em geral por parte da operadora, o que aumenta o risco de sobrecarga dessa entidade.

A presente proposta reduz a sobrecarga na rede celular ao buscar apenas conteúdo requisitado pelos usuários, o que também é um dos principais objetivos ao se empregar comunicação D2D oportunística. Dessa forma, aumenta-se inclusive a utilidade dos conteúdos armazenados, garantindo o seu consumo em ao menos uma oportunidade. Essa mesma garantia não pode ser feita por meio das técnicas de pré-carregamento de conteúdo. Propõe-se aqui uma solução distribuída, considerando atributos de conteúdo, mobilidade e qualidade para substituição de conteúdo no *cache* (*cache replacement*) e *download* de conteúdo por meio de comunicação D2D oportunística.

Por fim, como também visto nessa seção, diferentes propostas dependem do conceito de *Mobile Edge Computing*, que centraliza a maior parte (ou todas) as decisões e o *cache* no servidor *Edge*. Na presente proposta, distribui-se o fardo do monitoramento entre o servidor de conteúdo e os dispositivos, permitindo a esses últimos decidirem de forma independente quais conteúdos serão mantidos no seu respectivo *cache*.

3.4 Conclusão

Este capítulo reuniu as principais propostas focadas no problema de gerenciamento do *cache* dos dispositivos, de forma a definir o melhor conteúdo a se manter no *cache* ao longo do tempo.

A maior parte dos trabalhos desconsidera a relação conjunta entre atributos inerentes à mobilidade dos dispositivos e do conteúdo trafegado. Estas informações permitem determinar a capacidade e a importância de se transmitir um conteúdo via

D2D em determinado dispositivo. Além disso, a maior parte dos trabalhos centraliza o monitoramento e todas as decisões operacionais em uma única entidade da rede, o que prejudica a escalabilidade em cenários com uma grande quantidade de dispositivos.

O presente trabalho visa distribuir as tarefas de monitoramento entre o servidor de conteúdo e os dispositivos, de forma a determinar quais conteúdos são mais adequados para compartilhamento via comunicação D2D. Com isso, a solução se torna mais escalável e mais simplificada, ao mesmo tempo reduzindo a sobrecarga na operadora e até mesmo melhorando a qualidade do conteúdo trafegado em alguns cenários.

Capítulo 4

Substituição de Conteúdo em *Cache* baseada em Atributos de Mobilidade e Conteúdo

Este capítulo descreve os modelos de prioridade MP-PG e MP-CQ, apresentados na Seção 1.3 para o problema de gerenciamento de *cache*. A Seção 4.1 descreve dois modelos de rede, sendo que cada modelo de prioridade emprega um modelo de rede específico. A Seção 4.2, por sua vez, apresenta a modelagem matemática do MP-PG e do MP-CQ. Por fim, a Seção 4.3 conclui este capítulo.

4.1 Modelos de Rede

Esta seção descreve dois modelos de rede, denominados *Modelo de Rede de Propósito Geral* (MR-PG) e *Modelo de Rede Ciente de Qualidade* (MR-CQ), empregados pelo MP-PG e pelo MP-CQ respectivamente.

O MR-PG caracteriza conteúdos não segmentados (sem divisão em *chunks*) e cuja qualidade não varia. Diferentes tipos de conteúdo podem se classificar nessas características, tais como texto, imagens e vídeo. O MR-PG também define a política empregada para a busca e *download* dos conteúdos na rede, bem como quais ações executar quando o MP-PG decide pela remoção de um conteúdo.

O MR-CQ, por sua vez, tem foco especial em aplicações de *streaming* de vídeo adaptativo. Esse modelo caracteriza conteúdos divididos em segmentos (*chunks*), cujo fornecimento aos usuários pode ocorrer em qualidade variável de acordo com as condições da rede. Devido à forma de representação dos conteúdos no MR-CQ, as

políticas de busca e *download* na rede e de remoção do *cache* são distintas às do MR-PG.

É importante notar que os modelos de prioridade MP-PG e MP-CQ, bem como os modelos de rede MR-PG e MR-CQ podem coexistir no mesmo sistema, uma vez que cada um define de forma independente as políticas de análise, busca e *download* do conteúdo. Portanto, ao se considerar um cenário com dispositivos reais, bastaria verificar o tipo do conteúdo desejado para se empregar os modelos de prioridade e de rede mais adequados.

A Seção 4.1.1 apresenta as características comuns aos dois modelos de rede. Em seguida, as seções 4.1.2 e 4.1.3 apresentam as características específicas de cada um, visando implementar os modelos de prioridade propostos. Por fim, o Apêndice B apresenta uma descrição sumarizada de todas as variáveis empregadas em ambos os modelos de rede.

4.1.1 Características Comuns ao MR-PG e ao MR-CQ

Modela-se a rede como um Servidor de Conteúdo (SC) e um conjunto $D = \{d_1, d_2, \dots, d_{|D|}\}$ de dispositivos móveis, denotando-se d_i como o i -ésimo dispositivo contido em D . Cada dispositivo possui um *cache* de tamanho fixo de C_{\max} bytes. Considera-se que o SC possui um conjunto de arquivos $A = \{a_1, a_2, \dots, a_{|A|}\}$ a partir do qual os nós fazem requisições. Denota-se a_j como o j -ésimo arquivo no conjunto A , com tamanho s_j que varia de um a C_{\max} bytes e não se altera ao longo do tempo.

Tanto o SC quanto os dispositivos monitoram as requisições na rede em intervalos de monitoramento discretos. Cada dispositivo também monitora a duração do contato D2D com seus vizinhos.

Cada modelo de rede define um comportamento específico de busca e *download* dos arquivos, devido à sua estruturação. No entanto, por simplicidade, considera-se que o dispositivo baixa o arquivo completo, independentemente do canal de comunicação por onde o *download* dele ocorreu ou a sua respectiva qualidade (caso se trate de conteúdos com qualidade variável). O *download* de todo o arquivo por meio de comunicação D2D oportunística é garantido por ambos os modelos de rede, uma vez que o dispositivo monitora a duração do contato com seus vizinhos, visando avaliar a sua capacidade de transmissão por meio desse canal. Após o *download*, assume-se que o usuário consome o arquivo durante um tempo pré-determinado.

É importante ressaltar que o procedimento adotado para *download* de um arquivo, para cada modelo de rede, não é a única opção possível. Como trabalho

futuro propõe-se uma avaliação mais aprofundada sobre os efeitos das soluções propostas, considerando outros perfis de comportamento do usuário, bem como outros algoritmos de *download* de arquivos.

4.1.2 Características Específicas do MR-PG

No MR-PG, considera-se que os arquivos contidos em A são indivisíveis e de qualidade invariável. A Figura 4.1 apresenta o MR-PG.

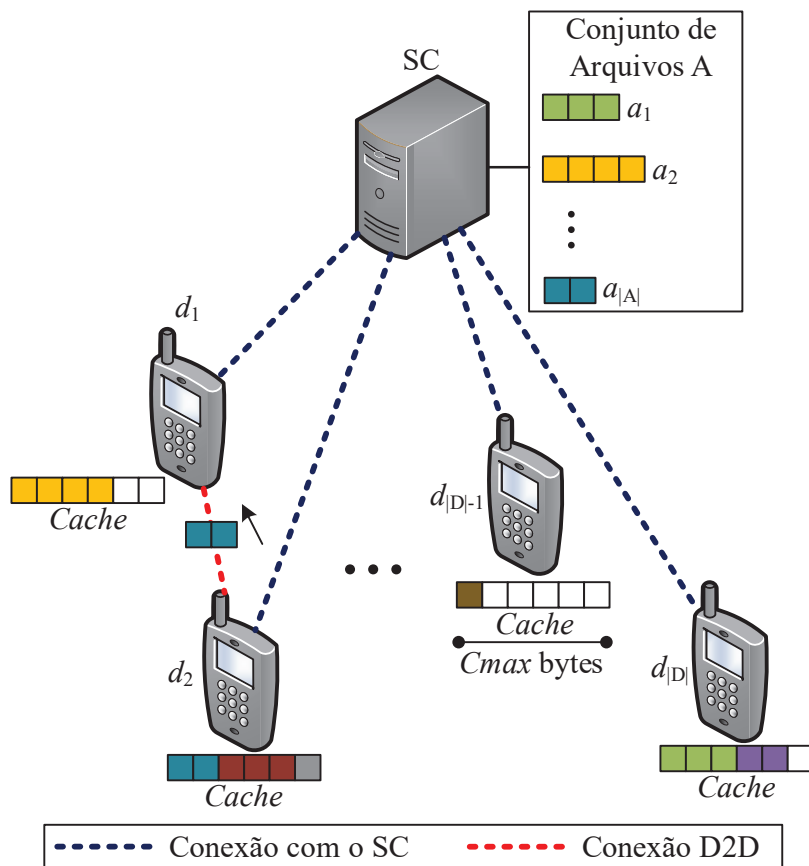


Figura 4.1. MR-PG.

Para cada *download* de um arquivo a_j , o dispositivo segue os passos abaixo:

1. O dispositivo busca por a_j na vizinhança D2D durante um tempo pré-determinado, enviando periodicamente mensagens *broadcast* aos seus vizinhos com informações de a_j .
2. Caso o dispositivo encontre a_j em qualquer um dos seus vizinhos, ele realiza um compartilhamento via comunicação D2D.

3. Caso o dispositivo não encontre a_j após o intervalo de tempo pré-determinado, ele encaminha a requisição para o SC.

Após o *download* de a_j , independente do canal de comunicação, caso o *cache* do dispositivo esteja sobrecarregado, inicia-se o procedimento de substituição de conteúdo no *cache*. Com base nos valores de prioridade calculados pelo MP-PG, inserem-se os arquivos no *cache* do mais pro menos prioritário, até que todo o espaço disponível seja preenchido. Os arquivos remanescentes são removidos por completo, devido à sua natureza indivisível.

4.1.3 Características Específicas do MR-CQ

No MR-CQ, considera-se que o SC provê o mesmo vídeo em $|Q|$ qualidades distintas. Em A , cada arquivo a_j está associado a uma ID de qualidade q_k contido em um conjunto $Q = \{q_1, q_2, \dots, q_{|Q|}\}$. Consequentemente, para a mesma informação provida pelo SC, haverá $|Q|$ respectivos arquivos de vídeo em A , cada um com uma qualidade distinta.

Além disso, divide-se cada arquivo de vídeo a_j contido em A em segmentos (ou *chunks*) de tamanho e duração fixos, definindo-se o tamanho de cada *chunk* pela respectiva ID de qualidade q_k contida em Q . Dessa forma, para cada vídeo desejado, os dispositivos podem receber *chunks* de diferentes qualidades por meio de comunicação D2D oportunistica ou a partir do SC.

A Figura 4.2 apresenta o MR-CQ. O dispositivo segue os passos abaixo para cada *download* de vídeo:

1. O dispositivo busca pelo vídeo desejado na vizinhança D2D durante um tempo pré-determinado, enviando periodicamente mensagens *broadcast* aos seus vizinhos com informações do vídeo.
2. Caso o dispositivo encontre um ou mais vizinhos, seleciona-se aquele que possui o vídeo desejado com a maior qualidade e o maior tempo de contato esperado.
3. Se o dispositivo requerente já possui uma parte do vídeo desejado em seu *cache*, e a qualidade do vídeo contido no *cache* do vizinho selecionado é maior, o dispositivo requerente faz o seu *download* por meio de comunicação D2D e descarta o vídeo com menor qualidade. Caso o dispositivo requerente não encontre nenhum vídeo no seu *cache*, ele faz o *download* do vídeo do vizinho selecionado da mesma forma.
4. Se o dispositivo não encontrar nenhum vizinho durante o tempo de busca pré-determinado, ou nenhum dos vizinhos encontrados possuírem o vídeo

desejado em maior qualidade, ele faz a leitura dos *chunks* disponíveis no seu *cache* (caso haja).

5. Caso haja *chunks* faltando após um *download* via D2D ou uma leitura no *cache*, o dispositivo faz o *download* deles a partir do servidor. A quantidade de *chunks* baixados pode variar de um a todos os *chunks* do vídeo desejado, sendo este último o caso em que o dispositivo não encontrou nenhuma parte do vídeo em seu *cache* ou na vizinhança D2D.

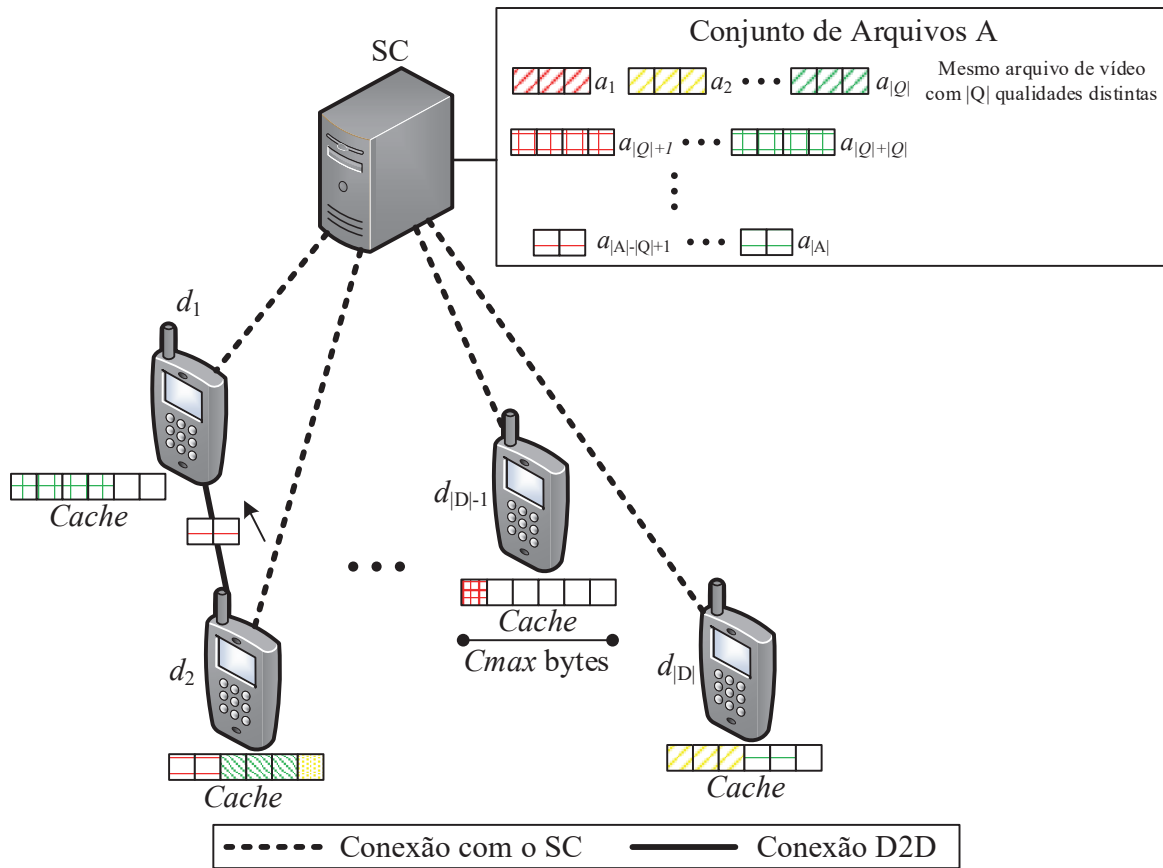


Figura 4.2. MR-CQ.

No *cache* do dispositivo, a organização dos *chunks* de cada vídeo se dá de forma contígua, agrupando *chunks* consecutivos de mesma qualidade em *blocos de chunks*. O procedimento de substituição de conteúdo ocorre de forma semelhante à do MP-PG, inserindo vídeos no *cache* do mais pro menos prioritário. Porém, caso seja necessário remover parte do vídeo de forma que o mesmo caiba no espaço disponível, o dispositivo começa por remover os últimos *chunks*, ou seja, os últimos segundos do respectivo vídeo, de forma que não haja lacunas na parte preservada no *cache*.

4.2 Modelos de Prioridade para Substituição de Conteúdo no *Cache* dos Dispositivos

Esta seção apresenta uma descrição detalhada de cada um dos modelos de prioridade propostos neste trabalho, mais especificamente como cada modelo computa valores de prioridade para cada arquivo transmitido na rede. A Seção 4.2.1 descreve o *Modelo de Prioridade de Propósito Geral* (MP-PG), ao passo que a Seção 4.2.2 descreve o *Modelo de Prioridade Ciente de Qualidade* (MP-CQ). A Seção 4.2.3 apresenta o cálculo da popularidade de um arquivo no *cache* de um dispositivo, sendo essa a única variável que é comum aos dois modelos. Por fim, o Apêndice B apresenta uma descrição sumarizada de todas as variáveis empregadas nos modelos de prioridade.

4.2.1 Modelo de Prioridade de Propósito Geral

A prioridade para um arquivo a_j no *cache* de um dispositivo d_i , aqui denotado como ζ_{a_j, d_i} , é um número real com um valor máximo indefinido. Computa-se esse valor por meio da multiplicação de dois termos principais, que aqui são denominados *Potencial de Envio* (PE) e *Potencial de Disseminação* (PD):

$$\zeta_{a_j, d_i} = \text{PE} \times \text{PD}. \quad (4.1)$$

O primeiro termo, PE, trata da mobilidade, relacionando o tamanho s_j do arquivo a_j a um limiar de tamanho λ_{d_i} , que depende do padrão de contato de d_i . Esse limiar é o tamanho máximo de arquivo que d_i deve ser capaz de transmitir com sucesso por meio de comunicação D2D oportunística. Portanto, arquivos que são maiores que λ_{d_i} recebem valores menores seguindo um fator exponencial. A Seção 4.2.1.1 mostra em detalhes como se obter o valor de λ_{d_i} . Para o momento, computa-se o valor de PE da seguinte forma:

$$\text{PE} = \frac{\lambda_{d_i}}{s_j}. \quad (4.2)$$

O segundo termo, PD, por outro lado, trata da popularidade. Denota-se a popularidade de um arquivo a_j para o dispositivo d_i como p_{a_j, d_i} (a Seção 4.2.3 explica passo a passo o cálculo dessa variável). Como mencionado anteriormente, os dispositivos são menos propensos a compartilhar arquivos grandes e pouco populares por meio de comunicação D2D, então deve-se computar valores de prioridade maiores para arquivos que são pequenos e populares ao mesmo tempo. Para tanto, pondera-se

a popularidade de a_j com seu respectivo tamanho s_j :

$$PD = \frac{p_{a_j, d_i}}{\log_{C_{\max}} s_j}. \quad (4.3)$$

É importante notar que para os maiores arquivos ($s_j = C_{\max}$ e $\log_{C_{\max}} s_j = 1$), o valor resultante dependerá apenas do valor computado para a popularidade p_{a_j, d_i} .

A seguir, a Seção 4.2.1.1 apresenta uma descrição detalhada de como se calcula o valor de λ_{d_i} .

4.2.1.1 Calculando o Limiar de Tamanho de Arquivos para cada Dispositivo

O valor do limiar de tamanho de arquivo para um dispositivo d_i (λ_{d_i}) depende de dois atributos principais: a média móvel exponencialmente ponderada (*Exponentially Weighted Moving Average* - EWMA) para o tempo de contato entre d_i e seus vizinhos, que aqui denota-se como $\overline{t_{d_i}}$, e a vazão alcançável de d_i por meio de comunicação D2D oportunística, denotado como ε . Por meio da multiplicação desses dois valores, define-se o maior arquivo que d_i deverá ser capaz de transmitir com sucesso para seus vizinhos:

$$\lambda_{d_i} = \overline{t_{d_i}} \times \varepsilon. \quad (4.4)$$

O dispositivo d_i atualiza o valor de $\overline{t_{d_i}}$ sempre que ele se desconecta de um vizinho. O novo valor de $\overline{t_{d_i}}$ é uma EWMA, ponderada com α , que depende do seu valor anterior e da duração do último contato t_{last, d_i} :

$$\overline{t_{d_i}} = \alpha \times t_{\text{last}, d_i} + (1 - \alpha) \times \overline{t_{d_i}}. \quad (4.5)$$

Quanto maior a EWMA para o tempo de contato ($\overline{t_{d_i}}$), maior será o tamanho de arquivo que d_i deverá ser capaz de compartilhar com sucesso com seus vizinhos por meio de comunicação D2D oportunística.

4.2.2 Modelo de Prioridade Ciente de Qualidade

A prioridade para um vídeo composto por *chunks* de uma ou mais qualidades (denotado como v_l , com l variando de um a $\frac{|A|}{|Q|}$, sendo esse último o número de vídeos distintos em $|A|$) no *cache* de um dispositivo d_i , denotado como η_{v_l, d_i} , é um número real com um valor máximo indefinido. Computa-se esse valor multiplicando dois

termos principais, que são denominados *Potencial de Disseminação com Qualidade* (PDQ) e *Potencial de Consumo* (PC):

$$\eta_{v_l, d_i} = \text{PDQ} \times \text{PC}. \quad (4.6)$$

O primeiro termo, PDQ, trata da popularidade de v_l , e em especial da qualidade de v_l em d_i . Denota-se a popularidade de um vídeo v_l para d_i como p_{v_l, d_i} .

Conforme descrito no início desta seção, v_l tem sua representação por meio de *chunks* de uma ou mais qualidades, possuindo com isso uma representação distinta de um arquivo a_j de qualidade única, pertencente ao conjunto A e descrito no MR-PG (Seção 4.1.2). No entanto, o cálculo de popularidade para a_j descrito na Seção 4.2.3 também vale para v_l , ou seja, p_{v_l, d_i} é igual a p_{a_j, d_i} , para todo a_j contido em A que representa a mesma informação de v_l , porém em qualidade única.

Como regra geral, os dispositivos da rede devem compartilhar vídeos populares, completos e de alta qualidade, conseqüentemente o MP-CQ deve computar valores maiores de prioridade para vídeos populares que possuem menos *chunks* faltando, além de ter predominância de *chunks* de maior qualidade. De forma a tornar isso possível, pondera-se a popularidade de v_l com uma métrica local denominada *Score de Qualidade*, aqui denotado como SQ_{v_l, d_i} (a Seção 4.2.2.1 descreve em detalhes o cálculo dessa variável), conforme mostrado na Equação 4.7 para o termo PDQ abaixo:

$$\text{PDQ} = p_{v_l, d_i} \times SQ_{v_l, d_i}. \quad (4.7)$$

Considerando que o usuário pode consumir um vídeo v_l mais de uma vez, o segundo e último termo, PC, estima o quão frequente d_i consome v_l ao longo do tempo, um termo que também define um conceito simples da preferência de d_i por cada vídeo.

O valor de PC é uma EWMA para o número de vezes que d_i consumiu v_l . Denota-se essa EWMA como $\overline{c_{v_l, d_i}}$, conforme mostrado abaixo:

$$\text{PC} = \overline{c_{v_l, d_i}}. \quad (4.8)$$

O dispositivo d_i atualiza a EWMA (isto é, o termo PC) em intervalos de monitoramento fixos. O valor atualizado de $\overline{c_{v_l, d_i}}$ é também uma EWMA, ponderada com β , que depende do seu valor anterior e do número de consumos para v_l no último intervalo de monitoramento, aqui denotado como $c_{\text{last}, v_l, d_i}$:

$$\overline{c_{v_l, d_i}} = \beta \times c_{\text{last}, v_l, d_i} + (1 - \beta) \times \overline{c_{v_l, d_i}}. \quad (4.9)$$

Quanto maior o valor da EWMA, maior será o valor de PC, e mais propenso d_i será para consumir v_l ao longo do tempo. Como consequência, o valor de PC proverá um impacto positivo sobre os resultados de PDQ, que combinados irão prover o valor de prioridade para o vídeo v_l no *cache* de d_i .

A seguir, a Seção 4.2.2.1 descreve o cálculo de SQ_{v_l, d_i} , e a Seção 4.2.2.2 descreve os critérios de seleção de vizinhos quando d_i encontra mais de um vizinho com o vídeo desejado.

4.2.2.1 Calculando o *Score* de Qualidade de um Vídeo no *Cache* do Dispositivo

O *score* de qualidade para um vídeo v_l no *cache* do dispositivo d_i , aqui denotado como SQ_{v_l, d_i} , é uma métrica local que quantifica os níveis de qualidade e completude de v_l no *cache* de d_i , ou seja, o quão alta é a qualidade de v_l e quantos *chunks* estão faltando.

Devido à natureza dinâmica de aplicações de *streaming* de vídeo baseadas em DASH, o servidor de conteúdo pode entregar cada vídeo aos dispositivos em qualidades distintas. Consequentemente, diferentes dispositivos podem possuir o mesmo vídeo com qualidade variada. Além disso, uma vez que os dispositivos podem remover um ou mais *chunks* de um vídeo baseado nos valores de prioridade calculados, o vídeo pode ficar incompleto no *cache* do dispositivo.

Como o MP-CQ visa otimizar o emprego de comunicação D2D oportunística e do *cache*, bem como otimizar a qualidade do vídeo consumido, é necessário priorizar vídeos de maior qualidade e que são mais íntegros (ou seja, menos *chunks* estão faltando).

Assume-se que d_i possui conhecimento prévio da duração total (em segundos) de v_l , aqui denotado como t_{v_l} , e que cada *chunk* tem a duração de um segundo. Computa-se o valor de SQ_{v_l, d_i} como a soma das porcentagens de duração (sobre a duração total de v_l) que d_i possui em seu *cache* para cada qualidade, ponderando com sua respectiva ID de qualidade contida no conjunto Q (veja o MR-CQ na Seção 4.1.3 para maiores detalhes). Quanto maior a qualidade de um *chunk*, maior será a sua respectiva ID de qualidade.

Denota-se uma porcentagem de duração como $0 \leq \theta_{v_l, q_k, d_i} \leq 1$. Para isso, observa-se o número de segundos que representam v_j na qualidade q_k no dispositivo d_i (ou seja, o número de *chunks* para o vídeo v_j na qualidade q_k no dispositivo d_i , aqui denotado como NC_{v_j, q_k, d_i}), conforme descrito na Equação 4.10:

$$\theta_{v_j, q_k, d_i} = \frac{NC_{v_j, q_k, d_i}}{t_{v_l}}. \quad (4.10)$$

A Equação 4.11 mostra o cálculo de SQ_{v_l, d_i} com base nas premissas apresentadas acima:

$$SQ_{v_l, d_i} = \sum_{k=1}^{|Q|} \theta_{v_l, q_k, d_i} \times q_k. \quad (4.11)$$

4.2.2.2 Critérios de Seleção de Vizinhos

Conforme descrito no MR-CQ (Seção 4.1.3), caso d_i encontre um vizinho com o vídeo v_l desejado em maior qualidade, d_i realiza o *download* a partir desse vizinho. Entretanto, caso d_i encontre mais de um vizinho nas condições acima, d_i deverá selecionar apenas um deles.

Para isso, define-se uma métrica local denominada *Score de Vizinho*, aqui denotada como $SV_{v_l, d_i, \kappa}$, onde κ é um vizinho de d_i contido em D que possui o vídeo v_l desejado. Calcula-se o valor de SV multiplicando o valor de SQ para v_l no *cache* de κ (veja a Seção 4.2.2.1 para maiores detalhes) pela duração restante estimada do contato entre d_i e κ (aqui denotado como $t_{\text{left}, d_i, \kappa}$), conforme mostrado abaixo:

$$SV_{v_l, d_i, \kappa} = SQ_{v_l, \kappa} \times t_{\text{left}, d_i, \kappa}. \quad (4.12)$$

Conforme descrito na Seção 4.1.1, d_i monitora a duração do contato com seus vizinhos, e denota-se $t_{\text{curr}, d_i, \kappa}$ como a duração do contato atual entre d_i e κ . Portanto, baseando-se nesse valor e na EWMA para o tempo de contato entre d_i e κ , aqui denotado como $\overline{t_{d_i, \kappa}}$ (Equação 4.14), é possível estimar por quanto tempo d_i irá se manter em contato com κ (Equação 4.13):

$$t_{\text{left}, d_i, \kappa} = \overline{t_{d_i, \kappa}} - t_{\text{curr}, d_i, \kappa}. \quad (4.13)$$

Quanto maior o valor de $t_{\text{left}, d_i, \kappa}$, mais propenso d_i será para receber os *chunks* de v_l armazenados em κ . Com isso, maior deverá ser o peso sobre $SQ_{v_l, \kappa}$, e $SV_{v_l, d_i, \kappa}$ será maior (Equação 4.12). Entre todos os vizinhos com um ou mais *chunks* de v_l armazenados em seu respectivo *cache*, d_i irá selecionar aquele com o maior valor de SV .

O dispositivo d_i atualiza o valor de $\overline{t_{d_i, \kappa}}$ em intervalos de monitoramento fixos. O novo valor de $\overline{t_{d_i, \kappa}}$ é uma EWMA, ponderada com γ , que depende do seu valor anterior e da EWMA para a duração dos contatos com κ no último intervalo de monitoramento (aqui denotado como $t_{\text{last}, d_i, \kappa}$):

$$\overline{t_{d_i, \kappa}} = \gamma \times t_{\text{last}, d_i, \kappa} + (1 - \gamma) \times \overline{t_{d_i, \kappa}}. \quad (4.14)$$

4.2.3 Calculando a Popularidade de um Arquivo em um Dispositivo

A popularidade de um arquivo a_j em um dispositivo d_i , aqui denotada como p_{a_j,d_i} , é um valor real que depende do número de requisições processadas para esse arquivo em toda a rede (ou seja, a popularidade global de a_j , que é uma EWMA e é aqui denotada como $\overline{p_{a_j,global}}$), bem como o número de requisições que d_i e seus vizinhos processaram por meio de comunicação D2D oportunística (ou seja, a popularidade local de a_j , também uma EWMA e aqui denotada como $\overline{p_{a_j,d_i}}$):

$$p_{a_j,d_i} = \overline{p_{a_j,global}} \times \overline{p_{a_j,d_i}}. \quad (4.15)$$

A atualização do valor da EWMA para a popularidade global $\overline{p_{a_j,global}}$ para o arquivo a_j ocorre em intervalos de monitoramento discretos. Esse valor define a porcentagem de dispositivos no conjunto D que requisitaram a_j no último intervalo de monitoramento, aqui denotado como $p_{last,global,a_j}$, ponderado com seu valor anterior (usando o parâmetro δ):

$$\overline{p_{a_j,global}} = \delta \times p_{last,global,a_j} + (1 - \delta) \times \overline{p_{a_j,global}}. \quad (4.16)$$

É importante notar que a popularidade global computada na Equação 4.16 não considera a distribuição geográfica das requisições de arquivo. Apesar de se considerar essencial expressar um monitoramento global que detecte tendências de requisição de arquivo, que por sua vez irão aumentar a sua respectiva prioridade, também é importante considerar o ponto de vista do dispositivo de uma forma isolada, isto é, o quão popular cada arquivo é no raio de comunicação do dispositivo. Dessa forma, evita-se computar valores altos de prioridade para arquivos que não são populares em um ou mais dispositivos, ao mesmo tempo considerando as tendências na rede como um todo.

Para tanto, pondera-se a popularidade global com uma métrica local, que é a EWMA para o número de requisições que o dispositivo d_i e seus vizinhos processaram por meio de comunicação D2D oportunística para o arquivo a_j ; denota-se esse valor como $\overline{p_{a_j,d_i}}$. A atualização do valor de $\overline{p_{a_j,d_i}}$ ocorre em intervalos de monitoramento fixos, ponderando com ϵ o último número de requisições computado para a_j (denota-se esse número como p_{last,a_j,d_i}), e $(1 - \epsilon)$ a EWMA atual para o número de requisições (Equação 4.17):

$$\overline{p_{a_j,d_i}} = \epsilon \times p_{last,a_j,d_i} + (1 - \epsilon) \times \overline{p_{a_j,d_i}}. \quad (4.17)$$

Combinando as duas métricas de popularidade das equações 4.16 e 4.17, obtém-se a popularidade para um arquivo a_j no dispositivo d_i , ou seja, p_{a_j, d_i} (Equação 4.15).

4.3 Conclusão

Este capítulo apresentou os modelos de rede (MR-PG e MR-CQ) e a modelagem matemática dos modelos de prioridade propostos (MP-PG e MP-CQ).

O Capítulo 5 a seguir apresenta a etapa de avaliação, descrevendo o desempenho das soluções propostas sob diferentes cenários.

Capítulo 5

Avaliação dos Modelos de Prioridade

Este capítulo apresenta a avaliação, por meio de simulações, dos modelos de prioridade MP-PG e MP-CQ. Primeiramente, a Seção 5.1 descreve as simulações. Em seguida, a Seção 5.2.1 apresenta os resultados obtidos com o MP-PG e a Seção 5.2.2 apresenta os resultados obtidos com o MP-CQ. Por fim, a Seção 5.3 apresenta as conclusões gerais em vista dos resultados obtidos.

5.1 Preparação do Ambiente de Simulação

Esta seção descreve a preparação do ambiente de simulação. Empregamos o simulador de redes oportunísticas *ONE* [Keränen et al., 2009]. É necessário que avaliemos cenários com uma grande quantidade de dispositivos para fazer observações adequadas sobre o desempenho das soluções propostas. Por conta disso, escolhemos um ambiente de simulação que permitisse que cenários desse tipo fossem avaliados exaustivamente, exigindo assim uma baixa quantidade de recursos de hardware e permitindo determinar a viabilidade e a escalabilidade das soluções propostas.

A seguir, a Seção 5.1.1 explica a definição e geração das requisições de conteúdo, bem como os padrões de mobilidade durante as simulações. Em seguida, a Seção 5.1.2 descreve os *baselines* empregados e a Seção 5.1.3 descreve os parâmetros de simulação. Por fim, a Seção 5.1.4 apresenta as métricas de desempenho empregadas nas avaliações, e a Seção 5.1.5 descreve as mensagens de controle inerentes aos *baselines*, bem como aos modelos de prioridade.

5.1.1 *Datasets* de Conteúdo e Mobilidade

Este trabalho emprega metadados do YouTube¹ para obter características realísticas de conteúdo nas simulações realizadas. O *dataset* padrão contém um registro diário de até 200 vídeos classificados como “trending”, para um conjunto de países entre Novembro de 2017 e Junho de 2018. O *dataset* apresenta cada vídeo *trending* para cada dia e o número de requisições recebidas pelos servidores do YouTube durante aquele dia. As simulações consideram um dia de medidas do *dataset* do YouTube em Novembro de 2017. Consideramos o conjunto de vídeos *trending* como o catálogo de arquivos que os usuários podem enviar requisições. Usamos o número de requisições de cada vídeo para definir o quão provável os dispositivos o requisitam durante as simulações.

É importante observar que a escolha de apenas um dia de medidas do *dataset* se deu apenas para fins de conveniência e simplificação das simulações, que também empregaram um tempo total de simulação de um dia. Entretanto, o emprego de intervalos superiores a um dia poderia ocorrer da mesma forma e com resultados semelhantes, uma vez que todos os vídeos do *dataset* seguem a mesma classificação de *trending*.

O *dataset* padrão descrito acima não provê o tamanho em bytes de cada vídeo. Portanto, associamos o tamanho de cada arquivo com um valor aleatório de acordo com uma distribuição Weibull, baseado em estudos prévios de Pang et al. [2017]. Uma vez que se gera uma requisição, associa-se ela a um dispositivo de acordo com uma distribuição uniforme.

Empregamos dois *traces* de contato coletados a partir de dispositivos reais. O primeiro deles é denominado *Infocom06* e compreende 98 usuários (78 móveis + vinte estáticos) carregando dispositivos ao longo de ambientes de escritório, conferências e cidades [Scott et al., 2009]. O segundo *trace* é denominado *Reality Mining* que coleta, além de uma série de tipos de eventos, informações de proximidade de 97 dispositivos no MIT [Eagle & Pentland, 2005].

Apesar dos *traces* de contato representarem um cenário real observado por dispositivos reais, eles possuem duas deficiências: i) a falta de detalhes acerca do perfil de mobilidade dos usuários, uma informação que ajudaria a compreender o impacto da mobilidade do usuário sobre o desempenho das soluções propostas; e ii) a impossibilidade de analisarmos a escalabilidade das soluções propostas, uma vez que os *traces* de contato possuem um número fixo de usuários.

Assim, a avaliação do MP-PG e do MP-CQ também emprega um modelo de

¹<https://www.kaggle.com/datasnaek/youtube-new>

mobilidade sintético denominado *Shortest Path Map Based Movement*. Neste modelo os usuários iniciam em uma posição aleatória no mapa. Durante a simulação, eles escolhem um destino no mapa e tomam o caminho mais curto para ele, com velocidades aleatórias uniformes bem como tempos de espera aleatórios uniformes entre dois movimentos.

5.1.2 *Baselines*

Os cenários de simulação comparam o MP-PG e o MP-CQ com dois *baselines* distintos: O LRS (*Least Recently Stored*) e o SACC (*Social-Aware Cooperative Caching*) [Wu et al., 2020], ambos descritos na Seção 2.3.

Escolhemos o LRS como *baseline* por se tratar de uma técnica clássica que, apesar de apresentar baixo custo em termos de implementação e execução, não considera atributos inerentes à mobilidade e conteúdo, sobretudo conteúdos com qualidade variável. A comparação do LRS com o MP-PG e o MP-CQ permite-nos avaliar o impacto de se considerar esses atributos no desempenho da rede, em termos do emprego de comunicação D2D oportunística e do *cache*.

Por outro lado, a escolha do SACC como *baseline* se deu pelo fato desse se tratar de uma técnica recente do estado da arte de substituição de conteúdo, e que também considera atributos de mobilidade e conteúdo, porém sob uma abordagem distinta da adotada pelo MP-PG e pelo MP-CQ. Além disso, o SACC tem como limitação o fato de não se considerarem conteúdos com qualidade variada, portanto não focando na melhoria dessa métrica de forma conjunta com o aumento do emprego de comunicação D2D oportunística e do *cache*.

Para o SACC, consideramos que todos os dispositivos sempre têm os mesmos interesses e assim pertencem à mesma comunidade com todos os dispositivos [Wu et al., 2020]. Além disso, o modelo original proposto pelos autores não especifica como computar a popularidade para dispositivos que têm apenas partes de um vídeo. Consideramos a habilidade de um dispositivo de prover pelo menos um *chunk* do respectivo vídeo, computando assim a métrica de popularidade para um vídeo da maneira mais próxima à do modelo original. Seguindo essa premissa, o SACC computará o mesmo valor de popularidade para vídeos que estiverem parcialmente (ou totalmente) presentes em todos os dispositivos. De forma a prover um “desempate” para esses casos, definimos um segundo critério para o SACC, removendo vídeos do maior para o menor, considerando o fato de que vídeos maiores são menos propensos a serem transmitidos por meio de comunicação D2D oportunística.

5.1.3 Parâmetros de Simulação

A Tabela 5.1 lista os parâmetros empregados nas simulações. Explicamos cada um deles a seguir:

Tabela 5.1. Parâmetros de Simulação.

Parâmetro	Variando o Número de Arquivos	Variando o Número de Usuários
<i>Parâmetros Gerais</i>		
Número de Execuções	10, intervalo de confiança de 95% (distribuição t de Student)	
Tempo de Simulação	86.400 segundos (24 horas)	
Número de Arquivos	10,20,50,100,150,200	200
Número de Usuários	98 (<i>Infocom06</i>) 97 (<i>Reality Mining</i>) 100 (Sintético)	10,20,50, 100,200 (Sintético)
Velocidade de Transmissão	5 MB/s	
Capacidade do <i>Cache</i> do Dispositivo	193.277.600 bytes	
Intervalo de Geração de Requisições	1.000-85.400 segundos	
Frequência de Requisição de Arquivos	Uma a cada 8,44 segundos	
Tempo de Busca de um Arquivo via D2D	50 segundos	
Tempo de Consumo de um Arquivo	35,4 segundos	
Qualidades de Vídeo	480p,720p,1080p	
Tamanhos de <i>Chunk</i>	53,42 KB (480p) 217 KB (720p) 326KB (1080p)	
<i>Parâmetros Específicos do Modelo de Mobilidade Sintético</i>		
Dimensões do Mapa	4.500 x 3.400 metros (Mapa da Cidade de Helsinki)	
Velocidade de Movimento dos Dispositivos	0,5-1,5 metros/segundo (valores aleatórios uniformes)	
Tempo de Espera entre Dois Movimentos	0-120 segundos (valores aleatórios uniformes)	
Raio de Comunicação	15 metros	
<i>Parâmetros Específicos do MP-PG e do MP-CQ</i>		
Intervalo de Monitoramento	10.000 segundos	
Pesos das EWMA's (α - β - γ - δ - ϵ)	0,1	

Número de Execuções. Escolhemos o número de execuções com base no tempo necessário para executar simulações com mais usuários, que tende a ser maior e a consumir maior quantidade de recursos de *hardware* em relação a cenários com menos usuários [Keränen et al., 2009]. O maior número considerado foi de 200 usuários. Fixamos o número de execuções em dez com intervalo de confiança de 95%, visando a obtenção de resultados em curto prazo para uma vasta gama de cenários, ao mesmo tempo garantindo alta precisão nos valores médios obtidos.

Tempo de Simulação. Configuramos o tempo de 86.400 segundos (um dia) devido ao uso do *dataset* de vídeos *trending* do YouTube (veja a Seção 5.1.1 para maiores detalhes).

Número de Arquivos. Considerando que o *dataset* de vídeos *trending* do YouTube possui 200 arquivos para cada dia de coleta, para cenários em que o número de arquivos é menor que 200, consideramos apenas as primeiras linhas e calculamos a porcentagem de visualizações apenas considerando esse subconjunto de linhas (veja a Seção 5.1.1 para maiores detalhes). A escolha de números de arquivos menores que 200 se deu de forma a avaliar o impacto do crescimento desse parâmetro no desempenho dos modelos de prioridade e dos *baselines*.

Número de Usuários. Escolhemos o número de usuários de acordo com os *traces* de contato. Uma vez que os *traces Infocom06* e *Reality Mining* compreendem 98 e 97 usuários respectivamente (veja a Seção 5.1.1 para maiores detalhes), fixamos um valor próximo de 100 usuários para o modelo de mobilidade sintético. A principal razão para essa escolha foi que dessa forma seria possível comparar o desempenho entre modelos de mobilidade. Uma vez que os *traces* de contato não permitem variar o número de usuários, nas simulações onde variamos esse parâmetro empregamos apenas o modelo de mobilidade sintético. De forma equivalente às premissas na escolha do número de execuções, escolhemos o número de usuários com base no tempo e recursos de *hardware* exigidos por cenários com mais usuários. Com isso, fixamos o número máximo de usuários em 200.

Velocidade de Transmissão e Raio de Comunicação. Empregamos o Wi-Fi Direct como tecnologia base para definir o desempenho das transmissões via D2D. Ele foi escolhido devido ao amplo suporte oferecido pelos dispositivos na atualidade [Camps-Mur et al., 2013]. No entanto, ressaltamos que outras interfaces de comunicação podem ser empregadas para comunicação D2D, levando-se em

consideração a heterogeneidade dos dispositivos dos usuários. Nas simulações, a definição da velocidade de transmissão e do raio de comunicação segue estudos recentes feitos por Reis et al. [2018], onde os autores avaliaram o desempenho da comunicação D2D por meio de dispositivos reais empregando a tecnologia Wi-Fi Direct. Os autores mediram, dentre outras métricas de desempenho, a vazão média com a transmissão de arquivos de tamanhos distintos. Escolhemos a maior distância considerada pelos autores como o raio de comunicação dos dispositivos (quinze metros) e a vazão média para o maior arquivo como a velocidade de transmissão (5 MB/s). Dessa forma, as simulações consideram um cenário realístico com contatos e desempenho de transmissão limitados. Por fim, é importante ressaltar que o raio de comunicação de quinze metros é válido apenas para o modelo de mobilidade sintético, uma vez que os *traces* de contato já consideram o raio de comunicação variável dos dispositivos reais que fizeram a coleta dos dados. Também é importante notar que a velocidade de transmissão não varia nas simulações devido a limitações inerentes ao cenário de simulação.

Capacidade do Cache do Dispositivo. Para fins de simplificação e comparações justas entre diferentes cenários, definimos o mesmo tamanho de *cache* para todos os dispositivos. Definimos o tamanho do *cache* com base no maior arquivo obtido a partir do *dataset* de vídeos *trending* do YouTube, que nesse caso foi de 184,32 MB (193.277.600 bytes). Dessa forma, garantimos que todos os arquivos caberiam por completo no *cache* dos dispositivos.

Intervalo de Geração de Requisições e Frequência de Requisição de Arquivos. Em um primeiro momento, consideramos um cenário onde cada usuário faz o *download* de uma cópia de todo o conteúdo disponível. Entretanto, durante as simulações variamos o número de arquivos e usuários, comparando o desempenho das soluções propostas e dos *baselines* com o emprego de diferentes combinações de parâmetros. Visando fazer comparações justas, configuramos a mesma frequência de requisição de arquivos para todas as simulações, considerando um cenário com 100 usuários e 100 conteúdos distintos, o que geraria 10.000 requisições de arquivo. Também não consideramos os primeiros e últimos 1.000 segundos do tempo total de simulação para fins de *warmup*. Portanto, para gerar 10.000 requisições de arquivo, as simulações deverão gerar uma requisição a cada 8,44 segundos.

Tempo de Busca de um Arquivo via D2D. A escolha do tempo de busca de 50 segundos para um arquivo via D2D ocorreu por meio de simulações que

compararam o desempenho do MP-PG e dos *baselines* ao variar esse parâmetro. A Seção 5.2.1.2 apresenta e discute os resultados obtidos. Empregamos o mesmo valor no MP-CQ e nos *baselines* para equiparar todos os cenários de simulação.

Tempo de Consumo de um Arquivo. Uma vez que todas as simulações se baseiam em conteúdo do tipo vídeo, escolhemos o tempo de consumo de um arquivo com base no estudo de Krishnan & Sitaraman [2013], que avaliaram o comportamento de 6,7 milhões de espectadores, totalizando 23 milhões de visualizações de vídeos. Os autores mediram um tempo mediano de reprodução de vídeo de 35,4 segundos antes de abandoná-lo. Este é um comportamento que os autores denominam *video surfing*, onde o espectador assiste a uma série de vídeos antes de decidir quais ele irá assistir por completo. Assumimos esse comportamento em todas as simulações.

Qualidades de Vídeo. Consideramos três níveis de qualidade (480p, 720p e 1080p), uma vez que esses são os níveis mais comuns empregados na plataforma YouTube. Além disso, assumimos que a qualidade de conexão entre o usuário e o servidor segue uma distribuição Normal, segundo estudos prévios feitos por Mota et al. [2013]. Assim, o servidor provê *chunks* com qualidade variável de acordo com a mesma distribuição.

Tamanhos de *chunk*. Definimos o tamanho dos *chunks* para cada nível de qualidade com base em estudos prévios feitos por Vaca-Rubio et al. [2019], que caracterizam a distribuição de tamanhos de *chunk* de serviços de *streaming* de vídeo baseados em DASH. Os autores avaliam *chunks* com duração de um e seis segundos. Por fins de conveniência, consideramos a distribuição de tamanhos de *chunk* com duração de um segundo. Uma vez que os autores proveem o tamanho médio de *chunk* para cada qualidade de vídeo em mais de um *bitrate*, escolhemos o tamanho de *chunk* associado ao maior *bitrate* para cada nível de qualidade. Portanto, para a qualidade de 480p definimos o tamanho do *chunk* em 53,42 KB e para as qualidades de 720p e 1080p definimos os tamanhos de *chunk* em 217 KB e 326 KB respectivamente [Vaca-Rubio et al., 2019].

Duração do Vídeo. A duração do vídeo é um parâmetro faltante no *dataset* de vídeos *trending* do YouTube. Nesse caso, assumimos que o tamanho de arquivo associado a cada vídeo do *dataset* (veja a Seção 5.1.1 para mais detalhes) pertence à qualidade de vídeo mais alta, isto é, 1080p. Visando calcular a duração do vídeo, dividimos o tamanho total pelo tamanho de um *chunk*. Por exemplo, se o vídeo possui 3.260 KB, ao considerar o tamanho de 326 KB para um *chunk* de um segundo em qualidade 1080p,

a duração do respectivo vídeo será de dez segundos e com isso o vídeo será composto por dez *chunks*.

Parâmetros Específicos do Modelo de Mobilidade Sintético. Empregamos os parâmetros definidos por padrão pelo modelo *Shortest Path Map Based Movement* para fins de simplificação. O único parâmetro modificado neste conjunto foi o raio de comunicação dos dispositivos, já descrito anteriormente.

Parâmetros Específicos do MP-PG e do MP-CQ. A escolha do intervalo de monitoramento, bem como dos pesos das EWMA's ocorreu por meio de simulações que observaram o desempenho do MP-PG ao variar esses parâmetros. A Seção 5.2.1.1 apresenta e discute os resultados obtidos. O MP-CQ empregou os mesmos valores escolhidos, uma vez que sua modelagem é inspirada no MP-PG.

5.1.4 Métricas de Desempenho

Cada modelo demandou métricas de desempenho específicas. A lista a seguir apresenta as métricas de desempenho consideradas na avaliação do MP-PG:

- *Taxa de acerto global no cache.* A porcentagem de arquivos que os dispositivos requisitaram e receberam por meio de comunicação D2D oportunística ou encontraram em seu respectivo *cache* (arquivos que os dispositivos já haviam baixado anteriormente);
- *Número de arquivos consumidos por minuto* por todos os dispositivos, por meio de cada opção (D2D, Servidor e *Cache*);
- *Sobrecarga de Mensagens de Controle.* A razão entre o número de mensagens de controle enviadas na rede e o número de arquivos consumidos por todos os dispositivos;
- *Sobrecarga de bytes de controle.* A razão entre o número de bytes de mensagens de controle enviadas por todos os dispositivos da rede e o total de bytes transmitidos (controle + arquivos).

Por outro lado, na avaliação do MP-CQ consideramos as seguintes métricas de desempenho:

- *Taxa de acerto global de chunks.* É a porcentagem de *chunks* de todos os arquivos de vídeo que os dispositivos requisitaram e receberam por meio de comunicação D2D oportunística ou encontraram em seu respectivo *cache* (*chunks* que os dispositivos baixaram anteriormente);
- *Taxa de qualidade de chunks.* A porcentagem de *chunks* de cada qualidade consumida por todos os dispositivos por meio de comunicação D2D oportunística e do *cache*, considerando todos os vídeos;
- *Sobrecarga de mensagens de controle.* A razão entre o número de mensagens de controle enviadas na rede e o número de *chunks* consumidos por todos os dispositivos, isto é, quantas mensagens de controle foram necessárias para consumir um *chunk*, independente do canal de transmissão empregado;
- *Sobrecarga de bytes de controle.* A razão entre o número de bytes de mensagens de controle enviadas por todos os dispositivos na rede e o total de bytes transmitidos (controle + *chunks*).

5.1.5 Mensagens de Controle

Esta seção descreve as mensagens de controle enviadas pelos os *baselines* LRS e SACC, bem como pelos modelos de prioridade MP-PG e MP-CQ. É com base nas mensagens especificadas nesta seção que computamos a sobrecarga de mensagens e de bytes de controle especificadas na Seção 5.1.4.

Em todos os cenários, consideramos um servidor de monitoramento (SM) centralizado e localizado na operadora, que coleta e computa informações de monitoramento que são de interesse de um ou mais dispositivos da rede (como por exemplo, a popularidade global descrita na Seção 4.2.3). É importante observar que as tarefas desempenhadas pelo SM podem ocorrer em locais distintos, de acordo com a infraestrutura de rede onde se emprega a solução de gerenciamento do *cache*.

Para fins de simplificação das avaliações inerentes à sobrecarga de controle, consideramos apenas o conteúdo útil (*payload*) das mensagens, ignorando portanto informações ligadas ao cabeçalho. Também consideramos que nenhuma das mensagens é reenviada em razão de perdas ou falhas na transmissão. Por fim, é importante observar que as mensagens consideradas nas avaliações são inerentes apenas à camada de aplicação. Com isso, desconsideramos mensagens geradas em camadas inferiores, ligadas a protocolos de roteamento ou de acesso ao meio, por exemplo.

O *baseline* LRS, por não depender de operações de monitoramento como o SACC, o MP-PG e o MP-CQ, terá sua sobrecarga de controle mensurada apenas em termos das requisições de arquivo (via D2D ou a partir do servidor) enviadas por ele, compostas pelos campos a seguir:

- ID do arquivo requisitado (inteiro) - 4 bytes
- Segundo a partir do qual o *download* deve ocorrer (inteiro) - 4 bytes (empregado apenas em *downloads* ocorridos a partir do servidor onde o conteúdo é dividido em *chunks*)

As seções a seguir apresentam uma descrição detalhada de cada uma das mensagens de controle inerentes ao SACC, bem como aos modelos de prioridade MP-PG e MP-CQ.

5.1.5.1 Mensagens Inerentes ao SACC

Para o SACC, com base no cálculo de popularidade em duas etapas descrito na Seção 2.3, consideramos que as informações necessárias para esse cálculo são reunidas no SM. Por conta disso, a cada operação de substituição de conteúdo no *cache* realizada por um dispositivo, o mesmo irá requisitar ao SM o processamento da popularidade dos arquivos que ele possui.

Com base nas afirmações acima, consideramos as mensagens de controle a seguir. Para fins de referência a cada tipo de mensagem, identificamos cada uma com uma sigla no formato “SA-X”, sendo “X” um número entre 1 e 5.

- **SA-1:** Requisição de um arquivo para um vizinho via D2D ou para o servidor de conteúdo
 - ID do arquivo requisitado (inteiro) - 4 bytes
 - Segundo a partir do qual o *download* deve ocorrer (inteiro) - 4 bytes (empregado apenas em *downloads* ocorridos a partir do servidor onde o conteúdo é dividido em *chunks*)
- **SA-2:** Requisição ao SM da popularidade de cada arquivo contido no seu *cache*:
 - Uma lista com as IDs (valores do tipo inteiro) dos arquivos contidos no *cache* - 4 bytes para cada ID

- Uma lista com o valor resultante da habilidade de compartilhamento para cada um dos vizinhos com quem o dispositivo transmissor teve contato:
 - * ID do vizinho (inteiro) - 4 bytes
 - * Valor resultante da habilidade de compartilhamento (*double*) - 8 bytes

- **SA-3:** Requisição *broadcast* do SM a todos os dispositivos da rede (com exceção daquele que está realizando a substituição de conteúdo), solicitando os valores de habilidade de compartilhamento deles com cada um de seus vizinhos, bem como a lista de conteúdos contidos em seu respectivo *cache*:
 - ID do servidor de monitoramento (inteiro) - 4 bytes

- **SA-4:** Resposta de um dispositivo ao SM com os valores de habilidade de compartilhamento calculados, bem como a lista de arquivos contidos em seu *cache*:
 - Uma lista com o valor resultante da habilidade de compartilhamento para cada um dos vizinhos com quem o dispositivo transmissor teve contato:
 - * ID do vizinho (inteiro) - 4 bytes
 - * Valor resultante da habilidade de compartilhamento (*double*) - 8 bytes
 - Uma lista com as IDs (valores do tipo inteiro) dos arquivos contidos no *cache* - 4 bytes para cada ID

- **SA-5:** Resposta do SM ao dispositivo que está realizando a substituição de conteúdo, com os valores de popularidade calculados:
 - Uma lista com o valor resultante da popularidade calculada para cada arquivo que o dispositivo receptor possui:
 - * ID do arquivo (inteiro) - 4 bytes
 - * Valor resultante da popularidade (*double*) - 8 bytes

A seguir, definimos as mensagens de controle enviadas para cada evento de transmissão ou monitoramento, bem como a ordem e quantidade de mensagens enviadas em cada um.

- **Evento 1:** *Download* de um arquivo a partir do servidor (arquivo não dividido em *chunks*):
 1. Uma mensagem SA-1 (dispositivo consumidor)
 2. *Download do arquivo requisitado*

- **Evento 2:** *Download* de um arquivo a partir do servidor (arquivo dividido em *chunks*):
 1. Uma mensagem SA-1 (dispositivo consumidor)
 2. *Download do arquivo a partir do ponto requisitado*

- **Evento 3:** *Download* de um arquivo através de comunicação D2D (arquivo não dividido em *chunks*):
 1. Uma mensagem SA-1 (dispositivo consumidor)
 2. *Download do arquivo requisitado*

- **Evento 4:** *Download* de um arquivo através de comunicação D2D (arquivo dividido em *chunks*):
 1. Uma mensagem SA-1 (dispositivo consumidor)
 2. *Download do arquivo requisitado*
 3. *Se um arquivo incompleto foi recebido via D2D: Download da parte faltante a partir do servidor (veja Evento 2)*

- **Evento 5:** Após o *download* completo de um arquivo via D2D ou a partir do servidor, caso o *cache* esteja sobrecarregado:
 1. Uma mensagem SA-2 (dispositivo realizando substituição de conteúdo)
 2. Uma mensagem SA-3 (SM)
 3. $(|D| - 1)$ mensagens SA-4 (todos os dispositivos exceto aquele que está realizando a substituição de conteúdo)
 4. Uma mensagem SA-5 (SM)

5.1.5.2 Mensagens Inerentes ao MP-PG e ao MP-CQ

Para o MP-PG e o MP-CQ, as mensagens de controle consideradas são apresentadas abaixo. De forma análoga ao SACC, identificamos cada mensagem com uma sigla no formato “MP-X”, sendo “X” um valor inteiro entre 1 e 7.

- **MP-1:** Requisição de um arquivo para um vizinho via D2D ou para o servidor de conteúdo
 - ID do arquivo requisitado (inteiro) - 4 bytes
 - Segundo a partir do qual o *download* deve ocorrer (inteiro) - 4 bytes (empregado apenas em *downloads* ocorridos a partir do servidor onde o conteúdo é dividido em *chunks*)
- **MP-2:** Notificação *unicast* de um dispositivo ao SM sobre um arquivo recebido via comunicação D2D:
 - ID do arquivo (inteiro) - 4 bytes
- **MP-3:** Notificação *broadcast* via comunicação D2D de um dispositivo aos seus vizinhos sobre um arquivo transmitido/recebido por ele pela mesma interface de comunicação:
 - ID do arquivo (inteiro) - 4 bytes
- **MP-4:** Notificação *broadcast* via comunicação D2D de um dispositivo aos seus vizinhos sobre um arquivo recebido por ele a partir do servidor de conteúdo:
 - ID do arquivo (inteiro) - 4 bytes
- **MP-5:** Requisição ao SM da popularidade global de cada arquivo contido no seu *cache*:
 - Uma lista com as IDs (valores do tipo inteiro) dos arquivos contidos no *cache* - 4 bytes para cada ID
- **MP-6:** Resposta do SM com a popularidade global de cada arquivo contido no *cache* do dispositivo que fez a requisição:
 - Uma lista com os valores de popularidade global computados:
 - * ID do arquivo (inteiro) - 4 bytes
 - * Valor da popularidade global (*double*) - 8 bytes

- **MP-7:** Notificação de contato de um dispositivo com um vizinho via D2D
 - ID do dispositivo transmissor (inteiro) - 4 bytes

A seguir, definimos as mensagens de controle enviadas para cada evento de transmissão ou monitoramento, bem como a ordem e quantidade de mensagens enviadas em cada um.

- **Evento 1:** *Download* de um arquivo a partir do servidor (MP-PG apenas):
 1. Uma mensagem MP-1 (dispositivo consumidor)
 2. *Download do arquivo requisitado*
 3. Uma mensagem MP-4, (dispositivo consumidor)
- **Evento 2:** *Download* de um arquivo a partir do servidor (MP-CQ apenas):
 1. Uma mensagem MP-1 (dispositivo consumidor)
 2. *Download do arquivo a partir do ponto requisitado*
 3. *Se o arquivo completo foi recebido a partir do servidor:* Uma mensagem MP-4, (dispositivo consumidor)
- **Evento 3:** *Download* de um arquivo através de comunicação D2D (MP-PG apenas):
 1. Uma mensagem MP-1 (dispositivo consumidor)
 2. *Download do arquivo requisitado*
 3. Uma mensagem MP-3, (dispositivo consumidor)
 4. Uma mensagem MP-3, (dispositivo compartilhador)
 5. Uma mensagem MP-2, (dispositivo consumidor)
- **Evento 4:** *Download* de um arquivo através de comunicação D2D (MP-CQ apenas):
 1. Uma mensagem MP-1 (dispositivo consumidor)
 2. *Download do arquivo requisitado*
 3. Uma mensagem MP-3, (dispositivo consumidor)
 4. Uma mensagem MP-3, (dispositivo compartilhador)

5. *Se um arquivo completo foi recebido via D2D: Uma mensagem MP-2, (dispositivo consumidor)*
 6. *Se um arquivo incompleto foi recebido via D2D: Download da parte faltante a partir do servidor (veja Evento 2)*
- **Evento 5:** Após o *download* completo de um arquivo via D2D ou a partir do servidor, caso o *cache* esteja sobrecarregado:
 1. Uma mensagem MP-5 (dispositivo consumidor)
 2. Uma mensagem MP-6 (SM)
 - **Evento 6:** Quando um evento de contato ocorre entre dois dispositivos:
 1. Uma mensagem MP-7 (dispositivo 1 envolvido no evento)
 2. Uma mensagem MP-7 (dispositivo 2 envolvido no evento)

5.2 Resultados

Esta seção apresenta os resultados obtidos com os modelos de prioridade MP-PG (Seção 5.2.1) e MP-CQ (Seção 5.2.2), com base nas simulações descritas na Seção 5.1.

5.2.1 MP-PG

Esta seção apresenta os resultados de simulação obtidos com o emprego do *Modelo de Prioridade de Propósito Geral* (MP-PG). Dividimos as simulações para esse modelo em três partes: (i) Ajuste de parâmetros importantes para o MP-PG (Seção 5.2.1.1); (ii) Comparação do MP-PG com os *baselines* LRS e SACC (Seção 5.2.1.2). A comparação se dá inicialmente em termos da *taxa de acerto global no cache*, variando-se o *tempo de busca de um arquivo via D2D*, bem como o número de arquivos e usuários. Finalizamos esta parte das simulações com a avaliação da sobrecarga de controle, também por meio da variação do número de arquivos e usuários; (iii) Investigação sobre a influência de cada componente do MP-PG na taxa de acerto global no *cache*, visando mostrar a importância de um modelo com todos os atributos descritos na Seção 4.2.1 (Seção 5.2.1.3).

5.2.1.1 Ajuste de Parâmetros

Esta parte das simulações avalia o MP-PG segundo os seguintes parâmetros de simulação: o intervalo de monitoramento e os pesos das EWMA. Consideramos um

cenário com 100 usuários, 100 arquivos distintos e a aplicação do modelo de mobilidade sintético.

Intervalo de Monitoramento

O intervalo de monitoramento indica a frequência de atualização das métricas de monitoramento global e local para cada arquivo (veja a Seção 4.2.1 para mais detalhes). Consideramos intervalos de monitoramento entre 5.000 segundos (aproximadamente 1,39 horas) e 50.000 segundos (aproximadamente 13,89 horas) com passo de 5.000 segundos, levando-se em conta o tempo total de simulação de 86.400 segundos (24 horas).

De acordo com os resultados apresentados na Figura 5.1, a taxa de acerto global no *cache* atinge um pico com um intervalo de monitoramento de 2,78 horas, decrescendo gradualmente para valores maiores. Esse é um comportamento esperado, uma vez que o emprego de intervalos mais longos tende a causar uma redução na precisão dos dados coletados, o que por sua vez pode prejudicar o desempenho dos modelos de prioridade. Ao mesmo tempo, o emprego de intervalos muito curtos faz com que as EWMA's sejam atualizadas com uma frequência desnecessária em alguns momentos, o que também prejudica a precisão das métricas de monitoramento e explica o desempenho inferior com um intervalo de 1,39 horas. Com base nesse comportamento, escolhemos o intervalo de 2,78 horas como o intervalo de monitoramento para todas as simulações.

Ainda que a maior taxa de acerto obtida (para um intervalo de monitoramento de 2,78 horas) tenha sido 3,77% superior ao menor valor obtido (para um intervalo de monitoramento de 13,89 horas), é necessário considerar que todas as simulações são limitadas à duração total de um dia, bem como *datasets* fixos de conteúdo e mobilidade. Além disso, os resultados apresentados na Figura 5.1 mostram que, para cada cenário, o intervalo de monitoramento da popularidade apresenta um respectivo valor ótimo, portanto demandando um ajuste dinâmico desse atributo. Propomos soluções dinâmicas como trabalhos futuros.

Pesos das EWMA's

Os pesos das EWMA's definem a relevância dos valores médios atuais em comparação com os últimos valores observados. Conforme descrito na Seção 4.2, empregamos EWMA's no monitoramento das métricas ligadas ao conteúdo e à

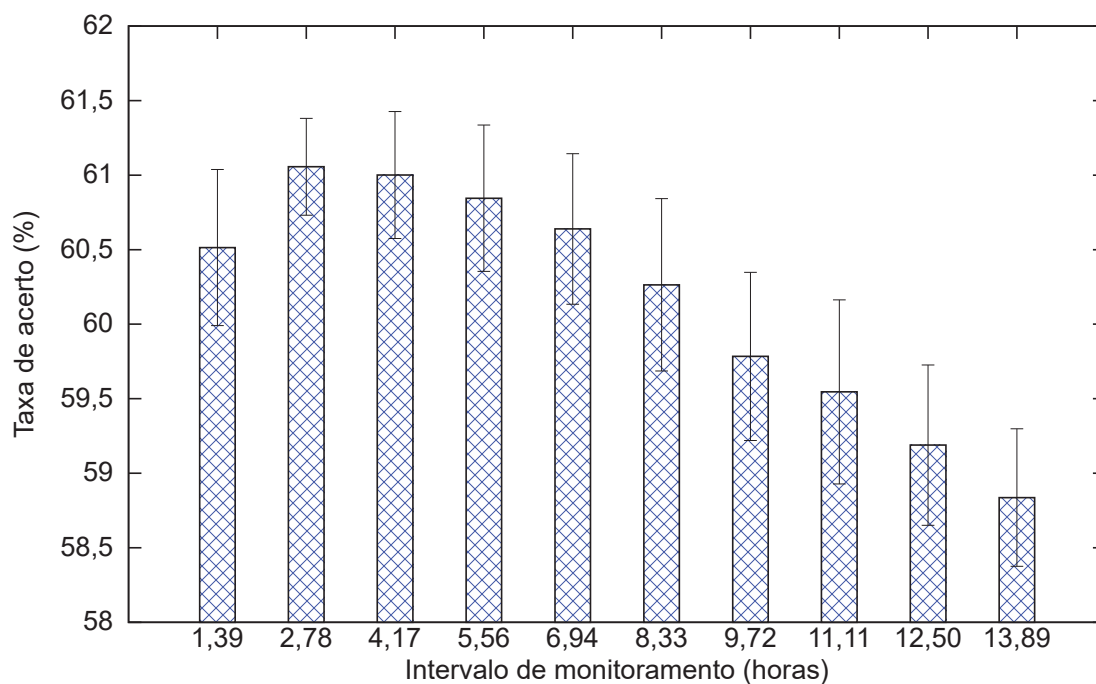


Figura 5.1. Variando o Intervalo de Monitoramento.

mobilidade dos dispositivos. Para fins de conveniência e simplificação, consideramos o mesmo valor para todos os pesos das EWMA's.

Também para fins de conveniência e simplificação, consideramos valores entre 0,1 e 0,9 com passo de 0,1. Conforme mostrado na Figura 5.2, a taxa de acerto no *cache* tem um acréscimo gradual com o decréscimo dos pesos. Com base nos resultados obtidos e em todas as outras simulações, escolhemos o valor que obteve a maior taxa de acerto no *cache*, que foi 0,1 nesse caso.

Ainda que haja uma tendência em se obter taxas de acerto maiores com o emprego de pesos menores que 0,1, devemos levar em consideração a melhoria obtida com esses valores. Como mostrado na Figura 5.2, a diferença obtida na taxa de acerto com o emprego de pesos entre 0,1 e 0,9 foi de menos de 10%, o que nos leva à conclusão de que melhorias mais significativas não serão obtidas com o emprego de pesos menores e de maior precisão.

5.2.1.2 Comparação com os *Baselines*

Esta parte avalia o MP-PG em comparação com dois *baselines* distintos, LRS e SACC, conforme descrito na Seção 5.1.2. Avaliamos as soluções em termos da taxa de acerto global no *cache*, bem como a sobrecarga de dados de controle em cenários distintos.

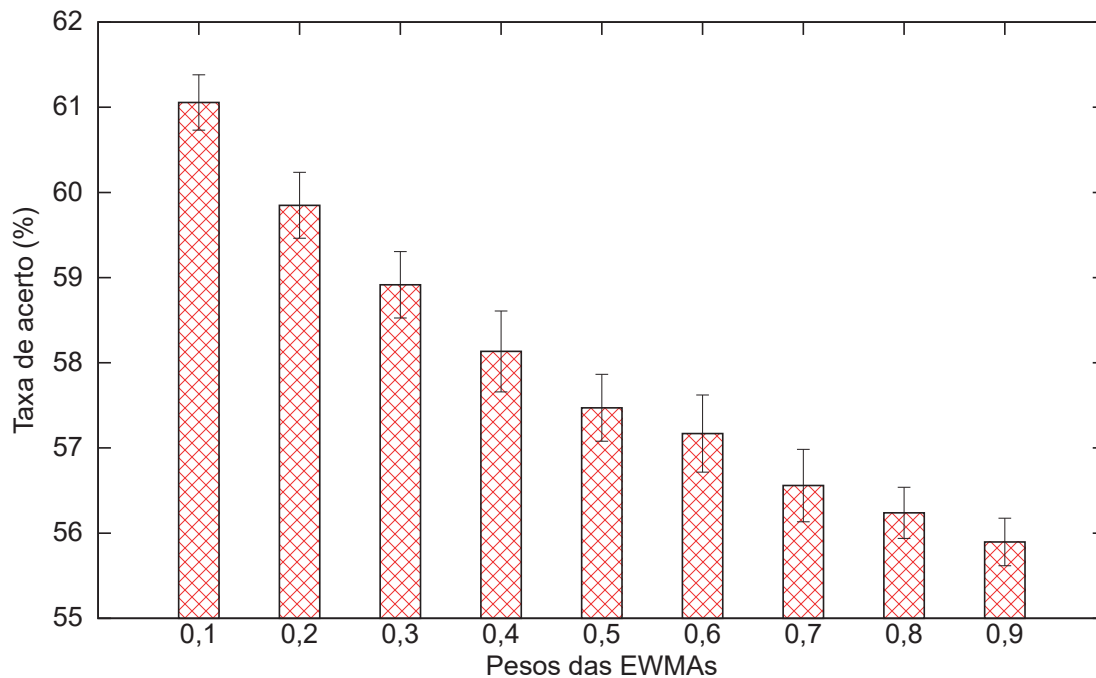


Figura 5.2. Variando os Pesos das EWMA's.

Taxa de Acerto Global no *Cache*: Variando o Tempo de Busca via D2D

O tempo de busca via D2D indica por quanto tempo o dispositivo busca pelo arquivo desejado na vizinhança D2D antes de encaminhar a requisição para o servidor de conteúdo. Segundo Krishnan & Sitaraman [2013], os usuários levam até 50 segundos para desistirem de fazer o *download* de um vídeo. Com base nesse limite, avaliamos o tempo de busca medindo a taxa de acerto global no *cache*, considerando tempos de busca entre 0,5 e 60 segundos, com passo de 0,5 segundos. Aplicamos os *traces* de contato *Infocom06* e *Reality Mining*, bem como o modelo de mobilidade sintético com 100 usuários, em um cenário com 100 arquivos distintos.

De acordo com os resultados apresentados nas figuras de 5.3 a 5.5, usuários com tolerância menor e maior (ou seja, tempos de busca menor e maior respectivamente) possuem desempenho similar, com uma diferença na taxa de acerto de menos de 1%. A baixa variabilidade da taxa de acerto se deve à dinâmica dos contatos entre os dispositivos, que impacta diretamente na probabilidade de se encontrar, durante o intervalo de busca, um vizinho D2D com o arquivo desejado.

Nos cenários avaliados, verificamos que os tempos de busca considerados não exercem impacto significativo na taxa de acerto. Ainda assim, observando-se os valores

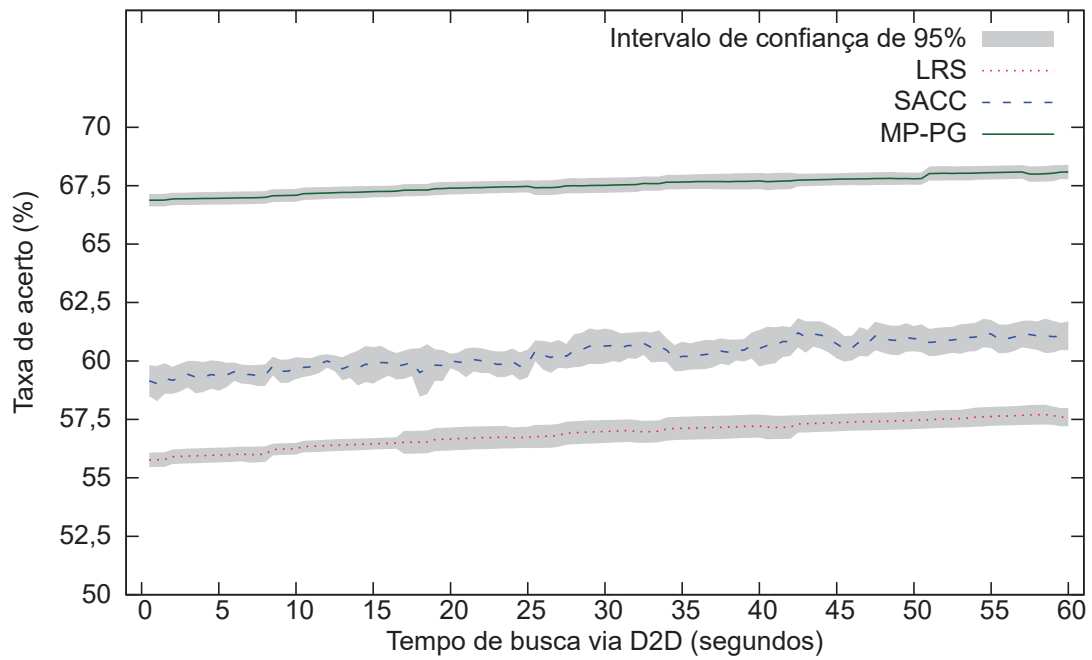


Figura 5.3. Variando o Tempo de Busca via D2D para o *Trace* de Contato *Infocom06*.

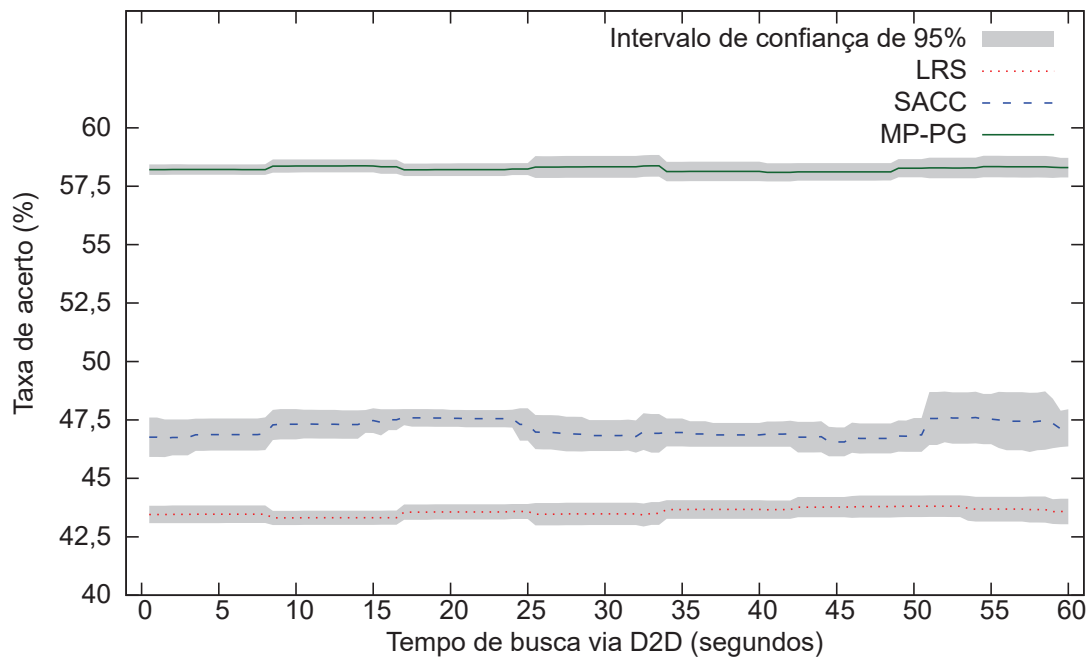


Figura 5.4. Variando o Tempo de Busca via D2D para o *Trace* de Contato *Reality Mining*.

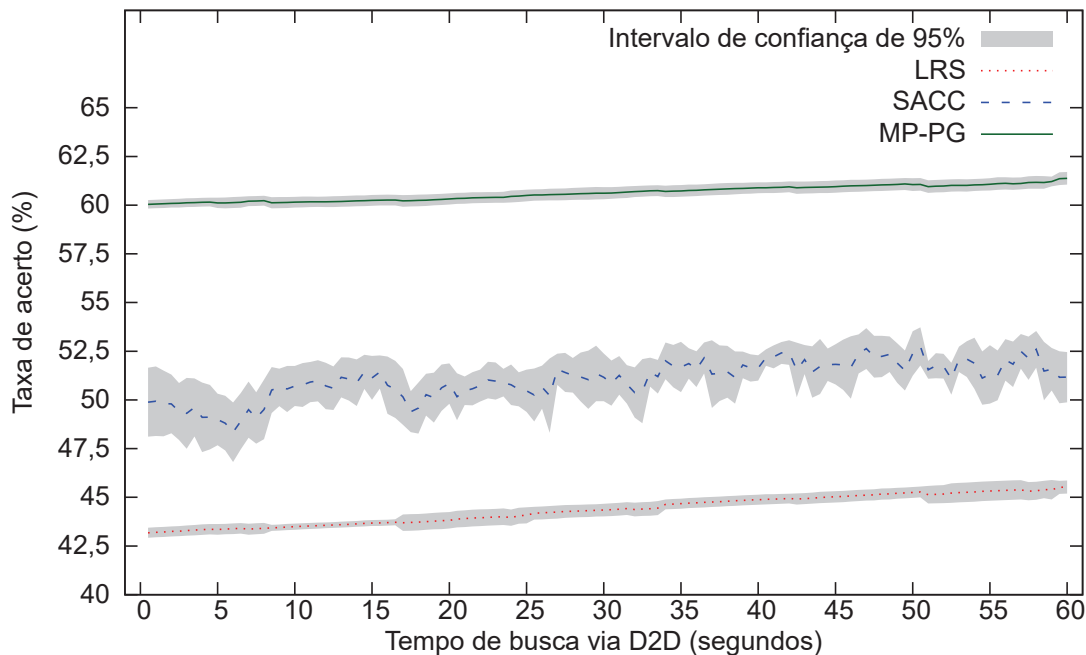


Figura 5.5. Variando o Tempo de Busca via D2D para o Modelo de Mobilidade Sintético.

absolutos obtidos por cada solução em cada cenário, verificamos taxas de acerto mais altas com tempos de busca longos. Esse é um comportamento esperado, uma vez que usuários mais pacientes tendem a encontrar o arquivo desejado na vizinhança D2D com maior frequência. Outro comportamento interessante a se apontar é o fato de que, para todos os cenários, o MP-PG supera os *baselines*, além de apresentar um comportamento mais previsível com intervalos de confiança mais curtos.

Normalizamos todos os usuários com um perfil *paciente* com um tempo de busca de 50 segundos. Tal fato permite observar o desempenho máximo das soluções propostas nos cenários avaliados e também permite comparações justas e mais detalhadas com os *baselines* em diferentes cenários.

Taxa de Acerto Global no *Cache*: Variando o Número de Arquivos e Usuários

As figuras de 5.6 a 5.8 mostram a taxa de acerto global no *cache* quando variamos o número de arquivos. Em todas as figuras e para todas as soluções, observamos um decréscimo gradual na taxa de acerto global no *cache* quando o número de arquivos na rede aumenta. Este é um comportamento esperado, uma vez que muitos arquivos distintos na rede reduzem a probabilidade de se encontrar um arquivo específico no *cache* do dispositivo ou na vizinhança D2D. Nessas simulações, o número médio de

vizinhos foi de 1,63, 0,13 e 0,07, para os *traces* de contato *Infocom06*, *Reality Mining* e o modelo de mobilidade sintético respectivamente.

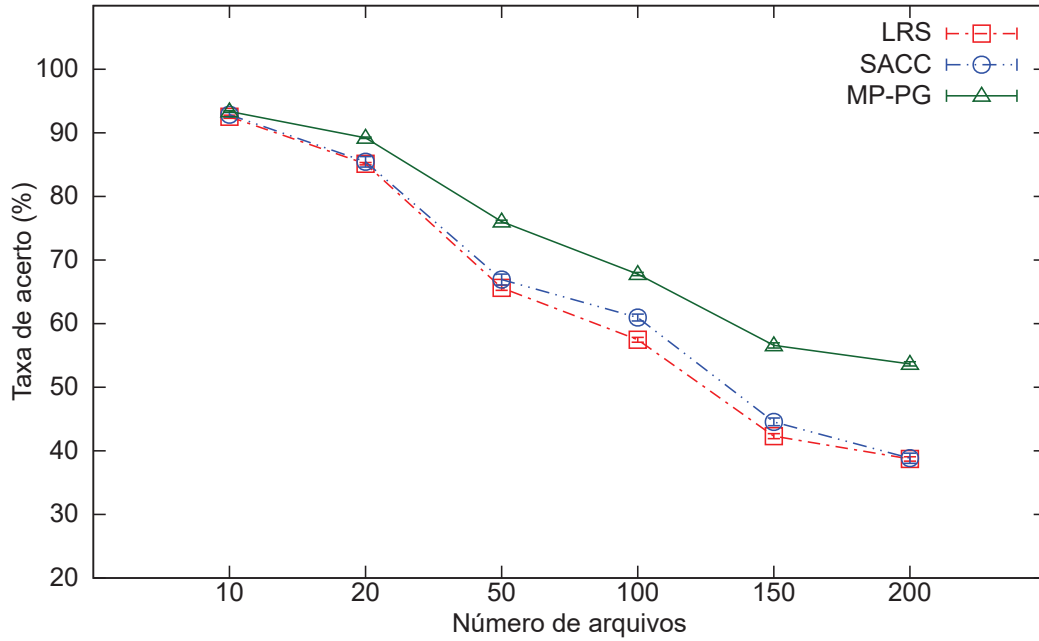


Figura 5.6. Variando o Número de Arquivos para o *Trace* de Contato *Infocom06*.

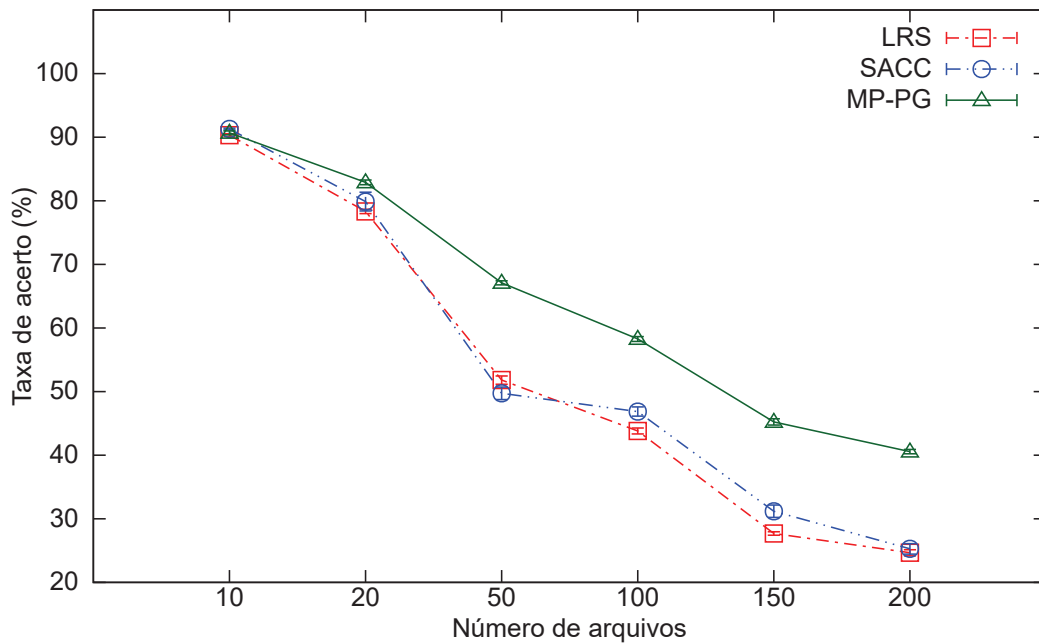


Figura 5.7. Variando o Número de Arquivos para o *Trace* de Contato *Reality Mining*.

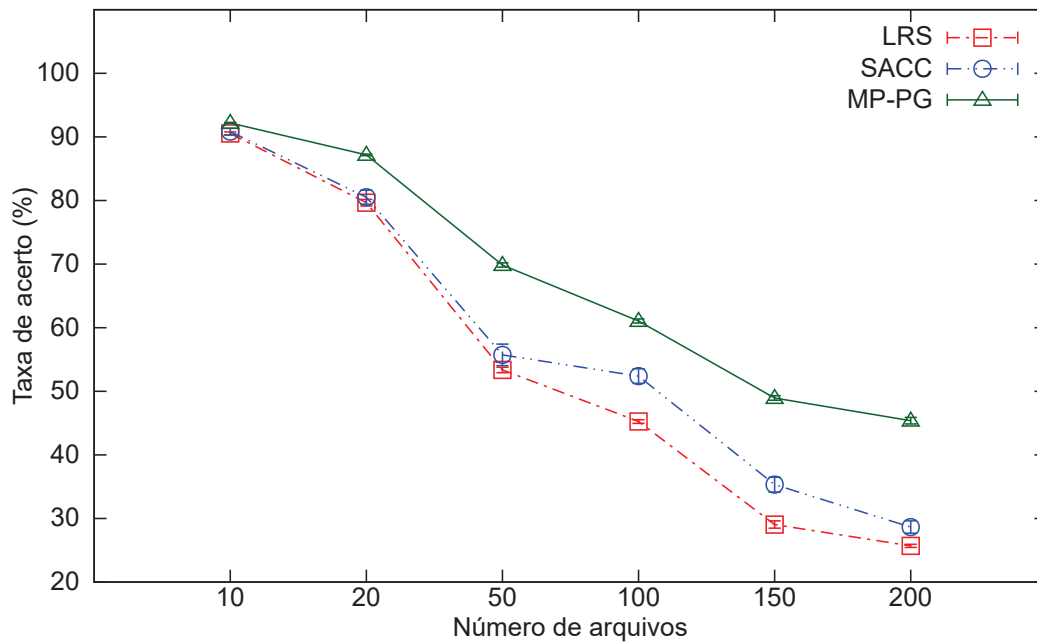


Figura 5.8. Variando o Número de Arquivos para o Modelo de Mobilidade Sintético.

Ainda no que concerne o decréscimo na taxa de acerto no *cache*, quando o número de arquivos varia de 150 a 200, o decréscimo na taxa de acerto é menor do que quando variamos de 100 a 150 em todas as soluções. Isso sugere que cenários com mais arquivos tendem a causar taxas de acerto menores na rede e o desempenho para todas as soluções tende a se estabilizar, não oferecendo melhorias mais significantes nessa métrica.

As maiores diferenças entre o MP-PG e o LRS ocorreram nos cenários com 200 arquivos. Nesses casos, o MP-PG supera o LRS em aproximadamente 38,6%, 64,48% e 76,56%, ao aplicarmos os *traces* de contato *Infocom06* e *Reality Mining* e o modelo de mobilidade sintético respectivamente. Ao compararmos o MP-PG com o SACC, existe uma melhoria de 38,2%, 60,46% e 58,33% para os mesmos respectivos cenários. A superioridade do MP-PG indica a importância de computarmos valores de prioridade para substituição de arquivos em cenários com muitos arquivos.

Todas as soluções proveem uma taxa de acerto similar em cenários com dez arquivos. Nesses cenários, no início da simulação, todos os dispositivos fazem o *download* da maior parte dos arquivos por meio de comunicação D2D ou a partir do servidor; em seguida, eles baixam a maior parte dos arquivos diretamente do *cache*. Tal comportamento justifica as taxas de acerto superiores a 90%, obtidas para todas as soluções nos cenários com dez arquivos.

A Figura 5.9 mostra o número de arquivos baixados em intervalos de 30 minutos para cada opção de *download* (por meio de comunicação D2D oportunística, do servidor ou do *cache*). Esses valores refletem uma única execução de 24 horas empregando-se o MP-PG, com um cenário com 100 usuários e aplicando o modelo de mobilidade sintético. Na Figura 5.9, após cinco horas de tempo de execução, todos os dispositivos começam a consumir a maior parte dos arquivos diretamente do seu respectivo *cache*. Omitimos os resultados com o emprego do LRS e do SACC para fins de simplicidade, uma vez que eles apresentaram o mesmo comportamento de *download* de arquivos.

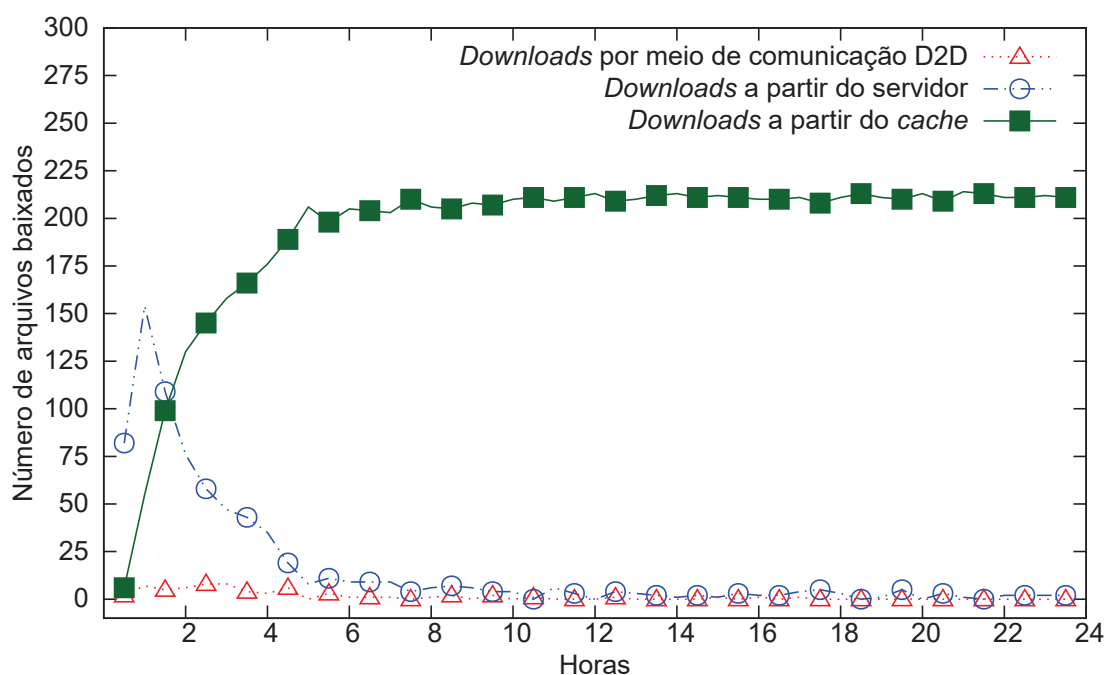


Figura 5.9. Número de Arquivos Baixados por Meio de Comunicação D2D, do Servidor e do *Cache* durante uma Execução do MP-PG.

Conforme descrito na Seção 5.1, também avaliamos os efeitos da variação do número de usuários na taxa de acerto global no *cache*. Fixamos o número de arquivos em 200 e aplicamos apenas o modelo de mobilidade sintético, devido à impossibilidade de fazermos o mesmo com a aplicação dos *traces* de contato. Nessas simulações, o número médio de vizinhos foi de 0,006, 0,015, 0,034, 0,07, e 0,139 para dez, vinte, 50, 100 e 200 usuários respectivamente.

A Figura 5.10 apresenta a taxa de acerto global no *cache*. Observamos um decréscimo na taxa de acerto com o emprego do MP-PG quando o número de usuários aumenta na rede. O mesmo comportamento também ocorre com o SACC, apesar de ter ocorrido com menor intensidade e não tão previsível quanto o MP-PG,

devido à alta variação observada por meio do intervalo de confiança. Isso também é um comportamento esperado, uma vez que os usuários têm um raio de comunicação limitado e estão restritos a se moverem nas ruas do mapa especificado nas simulações, um fato que ajuda a restringir o número médio de vizinhos na rede. Também é importante notar que cada dispositivo pode se comunicar por meio de comunicação D2D oportunística com apenas um dispositivo por vez. Um número limitado de vizinhos reduz o número de contatos e transmissões D2D em potencial na rede, que por sua vez causa uma “disputa” pelo uso de comunicação D2D oportunística. Tal fenômeno reduz a probabilidade de um dispositivo encontrar um vizinho D2D disponível com o arquivo desejado. Como consequência, a substituição de arquivos no *cache* com o MP-PG e o SACC provê melhorias menos significativas em cenários com mais usuários, uma vez que parte dos dispositivos passa a enviar requisições somente ao servidor.

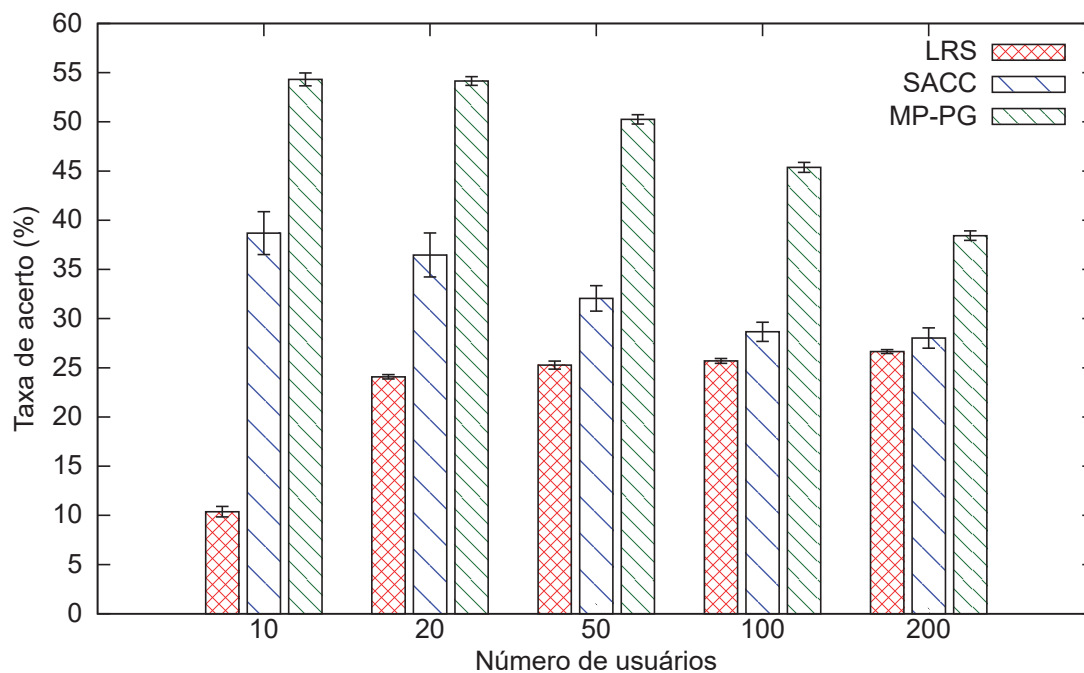


Figura 5.10. Variando o Número de Usuários para o Modelo de Mobilidade Sintético.

De forma contrária, observamos que o LRS teve um leve aumento na taxa de acerto global no *cache* com o aumento no número de usuários. Entretanto, a taxa de acerto começa a se estabilizar entre cenários com 50 e 200 usuários. Isso indica um potencial decréscimo para cenários com ainda mais usuários, dessa forma seguindo o mesmo comportamento já observado com o emprego do MP-PG e do SACC.

Outro ponto interessante é como o MP-PG supera o LRS e o SACC para cenários com dez usuários. Nesses cenários, a taxa de acerto no *cache* do MP-PG superou o LRS em 423,63% e o SACC em 40,41%, mostrando a importância de se ter uma técnica de substituição de conteúdo em *cache* em cenários com poucos usuários e poucos contatos entre eles.

É possível também notar que a superioridade do MP-PG diminui à medida em que o número de usuários aumenta, atingindo 44,22% sobre o LRS e 37,1% sobre o SACC no cenário com 200 usuários. É possível explicar esse fenômeno pelo fato de que cada dispositivo substitui conteúdo com base em sua respectiva observação local; portanto, deverão haver diferenças de conteúdo no *cache* de dispositivos distintos. Se forem considerados cenários realísticos com muitos usuários, onde um dispositivo tende a ter mais vizinhos, além de um número fixo de arquivos oferecidos na rede, a melhoria obtida com a substituição de arquivos no *cache* é limitada. Isso se deve ao fato de que haverá uma maior probabilidade de um dispositivo encontrar o que ele precisa em sua vizinhança.

Não é possível fazer a mesma afirmação acima para cenários realísticos onde o número de usuários e arquivos varia ao mesmo tempo. Ainda que tenhamos mostrado nesta seção a eficácia do MP-PG em cenários com muitos arquivos, é importante avaliar a relação entre esse parâmetro e o número de usuários na rede, de forma a determinar em quais casos o MP-PG provê melhor desempenho.

O emprego do MP-PG nos instantes de tempo em que ele provê melhorias significativas no uso de D2D e do *cache* também ajudaria a reduzir a sobrecarga de controle na rede. Além disso, isso contribuiria para a redução no consumo de energia de cada dispositivo, uma vez que o modelo deixaria de executar em determinados casos sem sucesso. Como trabalho futuro, propomos estudos mais aprofundados visando definir o que é aqui chamado de “regiões de operação”, ou seja, em quais momentos e como devemos aplicar o MP-PG.

Sobrecarga de Controle: Variando o Número de Arquivos e Usuários

Esta parte discute a sobrecarga de mensagens de controle e a sobrecarga de bytes de controle do MP-PG, em comparação com os *baselines*.

Conforme descrito na Seção 5.1.5, o LRS depende do registro do tempo de armazenamento de cada arquivo consumido por um dispositivo. Portanto, não há mensagens de controle para operações de substituição de conteúdo em *cache*. Como resultado, a sobrecarga de controle do LRS compõe-se apenas de requisições de arquivo. Por outro lado, o SACC precisa trocar mensagens de controle com todos os

dispositivos e o SM, de forma a descobrir quais dispositivos têm cada arquivo e calcular sua popularidade. Além disso, o SACC requer mensagens de controle para atualizar as métricas sociais entre dispositivos. De forma similar, o MP-PG realiza a troca de mensagens de controle para monitorar a popularidade global e local dos arquivos, bem como para monitorar o tempo de contato entre os usuários.

A Figura 5.11 mostra o comportamento da sobrecarga de mensagens de controle ao variar o número de arquivos, para cenários com 100 usuários e aplicando o modelo de mobilidade sintético. Omitimos os resultados para os *traces* de contato *Infocom06* e *Reality Mining* para fins de simplificação, uma vez que esses proveem comportamento similar.

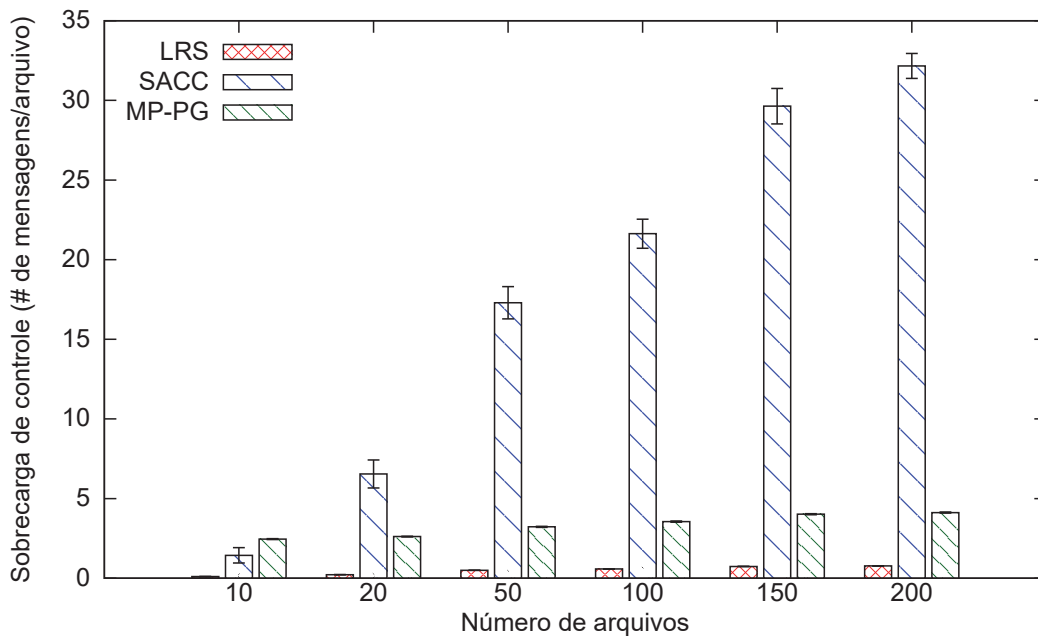


Figura 5.11. Sobrecarga de Mensagens de Controle com a Variação do Número de Arquivos.

O LRS proveu a menor sobrecarga de mensagens de controle entre todas as soluções, computando uma média de menos de uma mensagem de controle para cada arquivo consumido. Mesmo com a ausência de níveis significantes de sobrecarga de mensagens, o LRS apresenta baixo desempenho em termos da taxa de acerto global no *cache*, em cenários com poucos usuários (Figura 5.10). Além disso, em termos dessa mesma métrica, o MP-PG supera o LRS em pelo menos 38,6% em cenários com muitos arquivos (figuras de 5.6 a 5.8).

No caso do SACC, a sobrecarga de mensagens de controle tem um crescimento linear quando o número de arquivos aumenta na rede. Isso se deve ao fato de que

uma grande variedade de arquivos reduz a chance de um dispositivo encontrar o arquivo desejado na vizinhança D2D ou em seu respectivo *cache*. Nas figuras de 5.6 a 5.8 também observamos esse mesmo comportamento, com uma queda na taxa de acerto global no *cache* quando o número de arquivos na rede aumenta. Encontrar menos arquivos no *cache* implica que novos arquivos chegam no dispositivo com maior frequência, sobrecarregando o *cache* mais vezes e, portanto, demandando mais operações de substituição de conteúdo no *cache*. Como descrito anteriormente, para cada operação de substituição com o emprego do SACC, o SM precisa trocar uma série de mensagens de controle com os dispositivos para atualizar as métricas sociais entre eles, bem como para coletar informações sobre os arquivos armazenados em cada *cache*.

Ao contrário do SACC, o MP-PG apresenta baixa variação quando o número de arquivos aumenta. Isso acontece por duas razões, sendo que a primeira delas é referente aos custos de monitoramento constantes. Para cada *download* de arquivo por meio de comunicação D2D oportunística ou a partir do servidor, o número de trocas de mensagens de controle é o mesmo, de forma a atualizar a popularidade global e local conforme descrito anteriormente. O mesmo acontece quando uma substituição de conteúdo em *cache* ocorre, onde um dispositivo atualiza a popularidade global de seus arquivos com apenas uma troca de mensagens de controle com o servidor. Como o número de usuários e seu padrão de mobilidade é o mesmo, a sobrecarga em razão do monitoramento do tempo de contato é constante para todos os cenários apresentados na Figura 5.11.

A segunda razão para uma sobrecarga de controle quase constante com o emprego do MP-PG refere-se ao número de requisições de arquivo geradas na rede. Vale notar que a frequência de requisição de arquivos foi a mesma para todas as simulações, conforme descrito na Seção 5.1. Se for considerado um cenário hipotético onde o número de requisições é maior, a sobrecarga de controle do MP-PG apresentará um crescimento linear. Isso se deve ao fato de que os dispositivos sempre enviarão um número fixo de mensagens de controle na rede para cada consumo de arquivo, visando atualizar as métricas de monitoramento.

Apesar de um aumento no número de arquivos causar uma queda na taxa de acerto no *cache*, como também observado nas figuras de 5.6 a 5.8, o emprego do MP-PG permitiu a otimização do conjunto de arquivos armazenado no *cache*. Isso contribuiu para a redução no número de operações de substituição de conteúdo no *cache*, que por sua vez contribuiu para a redução na sobrecarga de mensagens de controle em comparação com o SACC.

No cenário com 200 arquivos, o MP-PG proveu uma sobrecarga de mensagens de

controle 87,19% inferior ao valor obtido com o SACC. No mesmo cenário, o MP-PG foi 58,33% superior ao SACC em termos da taxa de acerto global no *cache* (Figura 5.8). Em uma rede densa com vários dispositivos e vários eventos de contato entre eles, muitas operações de substituição de conteúdo em *cache* tendem a ocorrer. Com isso, a atualização da popularidade de cada arquivo precisará ocorrer com maior frequência. Conseqüentemente, muitas trocas de mensagens de controle entre os dispositivos e o SM ocorrerão no SACC. Além disso, várias das mensagens de controle tendem a ser longas, devido ao extenso histórico de contato e compartilhamento de arquivos de vários dispositivos.

Variamos o número de usuários para compreender o seu impacto na sobrecarga de mensagens de controle. Mantivemos o número de arquivos fixo em 200 e aplicamos o modelo de mobilidade sintético. A Figura 5.12 mostra os resultados.

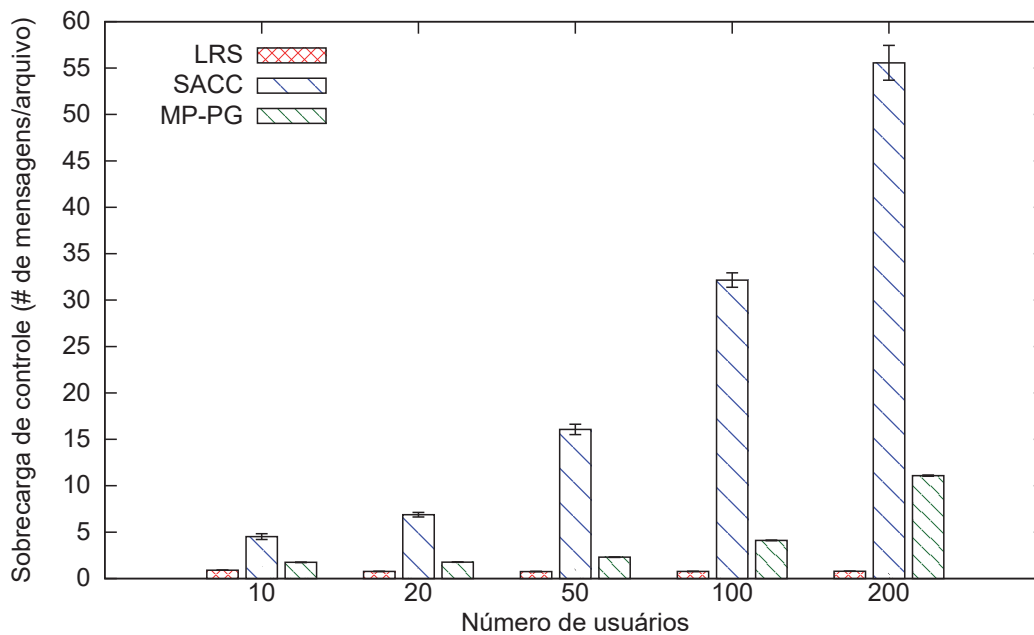


Figura 5.12. Sobrecarga de Mensagens de Controle com a Variação do Número de Usuários.

O LRS apresentou valores constantes para a sobrecarga de mensagens de controle, o que é esperado uma vez que não há trocas de mensagens de controle visando substituir conteúdo no *cache*. Apesar da sobrecarga de controle baixa e constante, é importante lembrar que o LRS atinge uma taxa de acerto no *cache* de cerca de 10% no cenário com dez usuários, como mostrado na Figura 5.10. Nesse mesmo cenário, o MP-PG demandou menos de duas mensagens de controle para cada arquivo (como mostrado na Figura 5.12) e proveu uma taxa de acerto no *cache* cerca de quatro vezes maior

que a obtida pelo *LRS* (como mostrado na Figura 5.10). Já o *SACC*, apesar da taxa de acerto no *cache* ser quase três vezes maior em comparação com o *LRS*, possui o custo de cerca de quatro mensagens de controle para cada arquivo baixado, o dobro em comparação com o *MP-PG*.

Tanto o *SACC* quanto o *MP-PG* apresentaram um crescimento linear em cenários com mais de vinte usuários. No caso do *SACC*, esse crescimento é devido ao aumento constante nos custos de monitoramento, o que demanda a coleta de dados de monitoramento de todos os dispositivos durante cada operação de substituição de conteúdo no *cache*. Portanto, com o aumento no número de usuários, será necessário o envio de mais mensagens de controle entre eles e o *SM*.

No caso do *MP-PG*, o aumento no número de contatos entre dispositivos explica o aumento na sobrecarga de mensagens de controle. O *MP-PG* demanda a troca de mensagens de controle para monitorar o tempo de contato. Apesar do crescimento no número de contatos na rede, a sobrecarga de mensagens de controle do *MP-PG* foi cerca de 80% inferior em comparação com o *SACC* no cenário com 200 usuários, ao mesmo tempo provendo uma superioridade em termos da taxa de acerto no *cache* de mais de 37% em comparação com o *LRS* e o *SACC* (Figura 5.10).

A Figura 5.13 mostra a sobrecarga de bytes de controle quando variamos o número de arquivos. O *LRS* apresentou uma sobrecarga de bytes constante, composta apenas por requisições de arquivo como mencionado anteriormente.

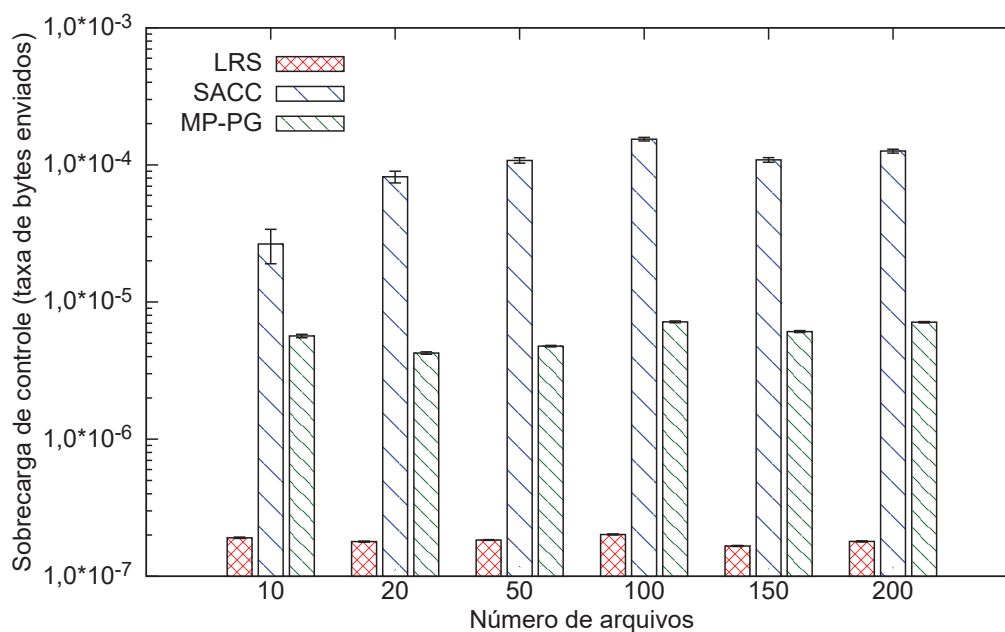


Figura 5.13. Sobrecarga de Bytes de Controle com a Variação do Número de Arquivos.

Comparando o MP-PG com o SACC, a sobrecarga de bytes de controle foi quase 96% menor. Isso se deve aos custos menores de monitoramento com o emprego do MP-PG, além de serem constantes em sua maior parte, como descrito anteriormente.

Outro ponto interessante a observarmos é que todas as soluções mantiveram quase a mesma sobrecarga de bytes de controle para todos os cenários. É possível explicar esse comportamento por meio da queda na taxa de acerto global no *cache*, com o aumento no número de arquivos para todas as soluções, conforme mostrado nas figuras de 5.6 a 5.8. Essa queda causa um aumento no número de bytes enviados entre os dispositivos e o servidor, um fato que ajuda a estabilizar a sobrecarga de bytes de controle entre todos os cenários. Também é importante notar que o emprego de uma solução de substituição de conteúdo em *cache* pode não reduzir de forma significativa o número de arquivos requisitados ao servidor. Esse fenômeno foi mostrado na Figura 5.10, em cenários com mais usuários. Além disso, há também o custo do monitoramento, um fato que aumenta o número de bytes de controle enviados na rede.

A Figura 5.14 mostra a sobrecarga de bytes de controle em função do aumento no número de usuários. O LRS foi a única solução que apresentou um comportamento constante.

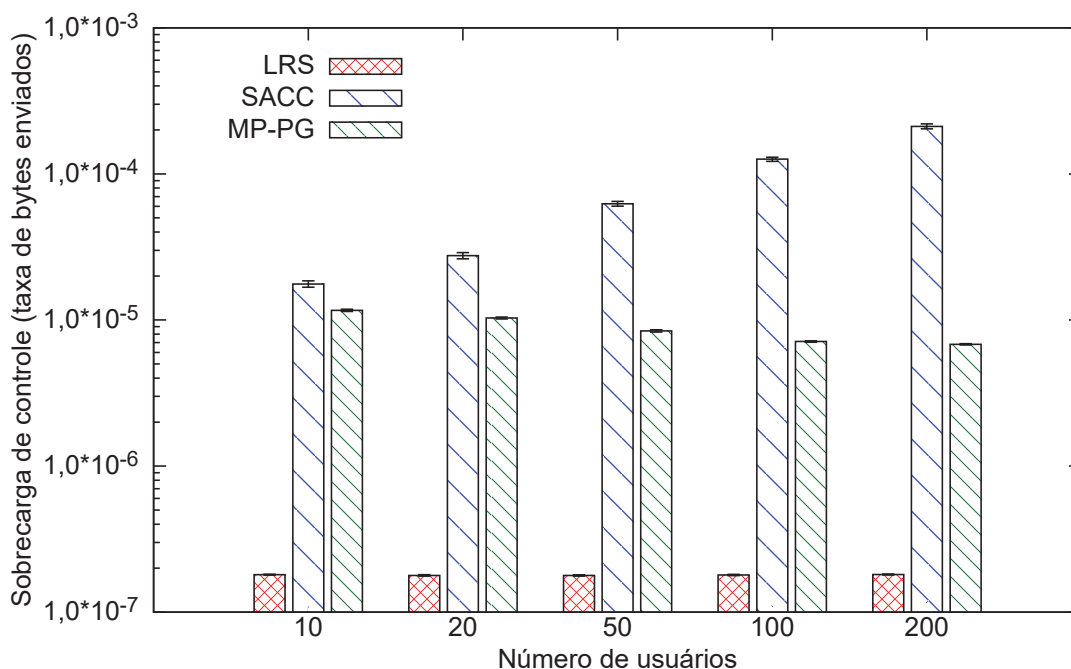


Figura 5.14. Sobrecarga de Bytes de Controle com a Variação do Número de Usuários.

Além disso, como esperado, o SACC apresentou um crescimento linear. Isso

quer dizer que um aumento no número de usuários significa que mais conexões tendem a ocorrer entre usuários, um fato que aumenta os custos de monitoramento do SACC, demandando a transmissão de uma quantidade de dados de monitoramento mais substancial para o SM.

Em comparação com o SACC, o MP-PG foi mais vantajoso em cenários com mais usuários. No cenário com dez usuários, a sobrecarga de bytes de controle foi 34,04% menor, enquanto que no cenário com 200 usuários a sobrecarga foi 96,78% menor. De forma similar ao que observamos na Figura 5.12, é importante ressaltar que no cenário com dez usuários, além de prover uma sobrecarga de bytes de controle menor, o MP-PG superou o LRS e o SACC em 423,63% e 40,41% respectivamente, em termos da taxa de acerto global no *cache*.

Outro comportamento interessante a ser observado na Figura 5.14 é como a sobrecarga de bytes de controle diminui ao empregarmos o MP-PG com o aumento no número de usuários. Entre os cenários com dez e 200 usuários, a queda foi de 41,47%. É possível explicar esse comportamento pelo crescimento quase constante na quantidade de dados de monitoramento trocada entre os dispositivos e o servidor, com o aumento do número de usuários na rede. Esse fenômeno se alia ao crescimento mais acelerado no número de *downloads* do servidor em cenários com mais usuários, conforme mostrado na Figura 5.10, com a queda na taxa de acerto global no *cache*. Um número maior de *downloads* do servidor significa que os dispositivos trocam menos mensagens de controle entre si por meio de comunicação D2D, reduzindo com isso a sobrecarga de bytes de controle.

5.2.1.3 Influência de Cada Componente do MP-PG

Esta parte das simulações avalia a taxa de acerto global no *cache* alcançável com variantes do MP-PG. As avaliações visam verificar a importância de cada componente do modelo no cálculo da prioridade para cada arquivo. Executamos simulações variando o número de arquivos e usuários para cada variante do MP-PG, comparando em seguida o desempenho de cada uma.

Nas simulações, variamos o MP-PG de cinco formas distintas, conforme descrito a seguir:

1. **MP-PG completo**, conforme descrito na Seção 4.2.1 e formalizado na Equação 4.1.
2. **Potencial de Disseminação (PD) apenas (Equação 4.3)**: Elimina o termo Potencial de Envio (PE) (Equação 4.2), que depende de um limiar de tamanho

de arquivo, conforme descrito na Seção 4.2.1. Com isso, removemos o aspecto da mobilidade, focando apenas na popularidade e no tamanho do arquivo (PD).

3. **Potencial de Envio (PE) apenas (Equação 4.2):** Elimina o termo Potencial de Disseminação (PD) (Equação 4.3), que avalia a popularidade alinhada ao tamanho do arquivo, conforme descrito na Seção 4.2.1. Assim, o dispositivo considera apenas o aspecto da mobilidade.
4. **Popularidade do arquivo apenas (Equação 4.15):** Este modelo considera apenas a variável de popularidade descrita na Seção 4.2.3, desconsiderando com isso os fatores de mobilidade e tamanho do arquivo. Quanto maior a popularidade, maior será a prioridade de um arquivo.
5. **Tamanho do arquivo apenas (s_j):** Consideramos esse modelo como o mais simples de todas as variantes do MP-PG, dando maior prioridade para arquivos menores, dessa forma preservando mais arquivos no *cache*.

As figuras de 5.15 a 5.17 mostram a taxa de acerto global no *cache* para cada modelo descrito acima, para cada número de arquivos entre dez e 200, com a aplicação de cada *trace* de contato e o modelo de mobilidade sintético. Ao aplicarmos o *trace Infocom06* e o modelo de mobilidade sintético (figuras 5.15 e 5.17 respectivamente), o MP-PG completo superou todos os outros para números de arquivos superiores a 50. Isso confirma as observações feitas na Seção 5.2.1.2, onde mostramos a importância do MP-PG completo em cenários com muitos arquivos. Ao mesmo tempo, o desempenho de todos os modelos foi similar em cenários com menos de 50 arquivos. Esse é outro comportamento que observamos nos resultados da Seção 5.2.1.2, onde um pequeno número de arquivos aumenta a probabilidade de um dispositivo encontrar tudo que ele deseja em sua vizinhança D2D ou em seu respectivo *cache*, reduzindo dessa forma os efeitos de uma técnica de substituição de conteúdo em *cache*.

Ainda considerando o *trace* de contato *Infocom06* e o modelo de mobilidade sintético, ao compararmos os modelos *PD apenas* e *PE apenas*, o modelo *PE apenas* leva a uma taxa de acerto no *cache* menor, principalmente para números de arquivos menores que 150. Ambos os termos dependem do tamanho do arquivo, levando-nos à conclusão de que a métrica de popularidade (parte do termo PD) provê efeitos positivos maiores, em comparação com o aspecto da mobilidade (parte do termo PE). Tal fenômeno ocorre uma vez que a maior parte das variações nos valores de prioridade vêm da popularidade e do tamanho dos arquivos, bem como o fato de que o MP-PG completo prioriza arquivos que são pequenos e populares ao mesmo tempo.

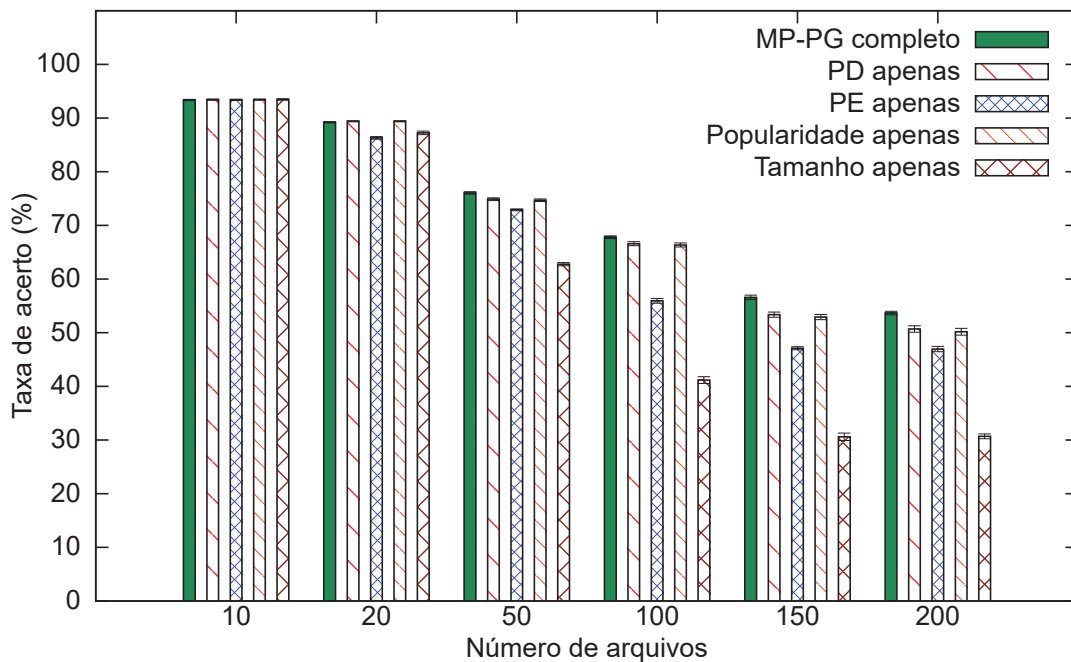


Figura 5.15. Variando o Modelo e o Número de Arquivos para o *Trace* de Contato *Infocom06*.

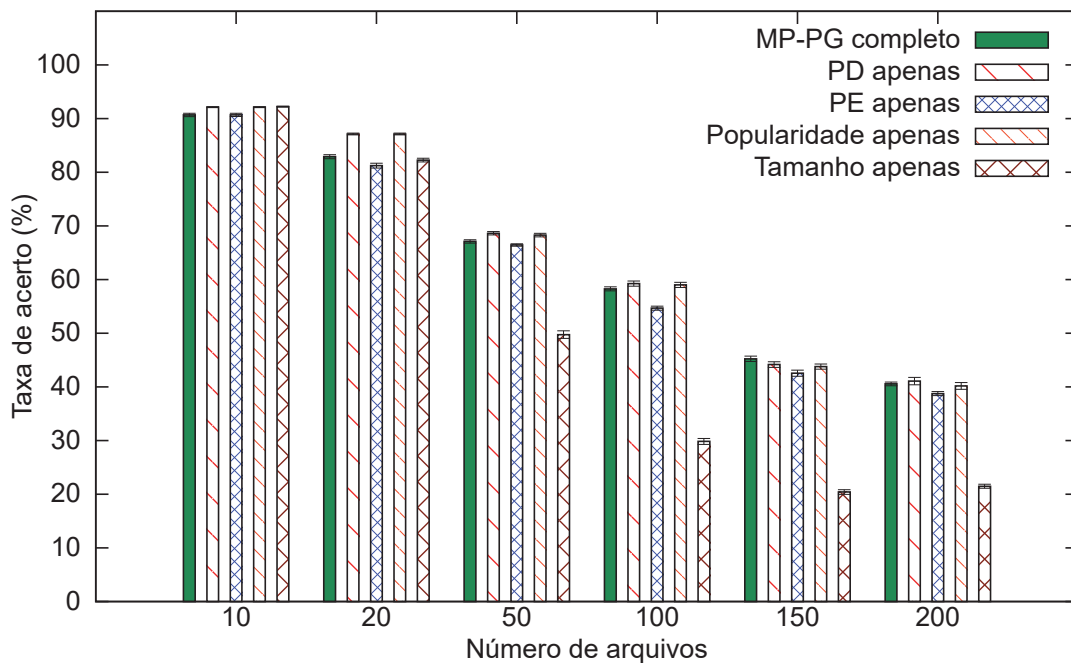


Figura 5.16. Variando o Modelo e o Número de Arquivos para o *Trace* de Contato *Reality Mining*.

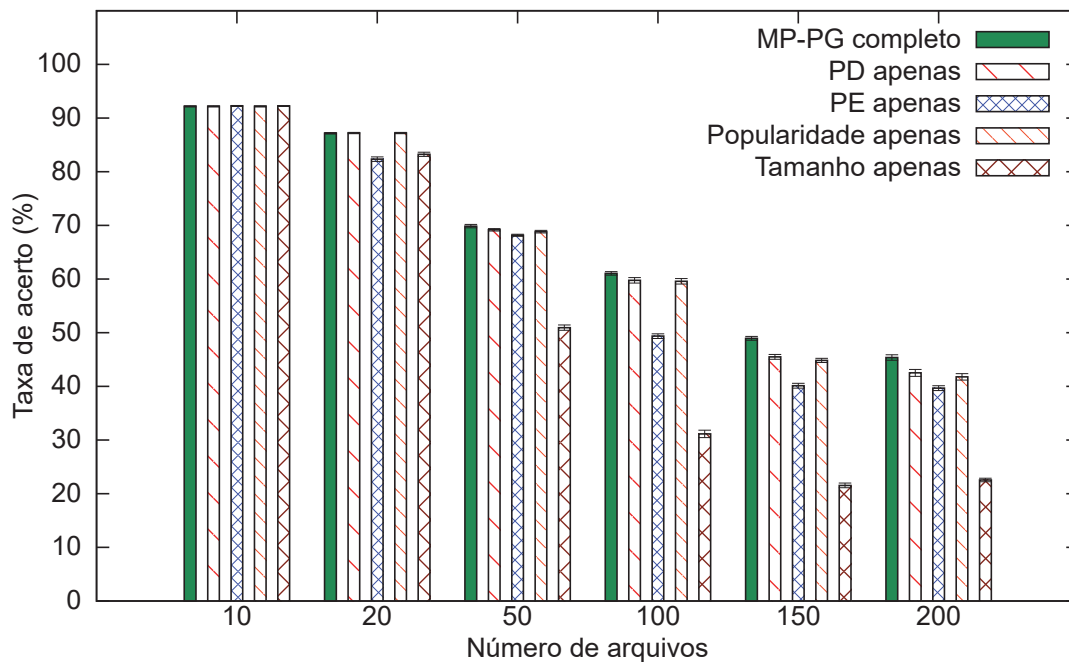


Figura 5.17. Variando o Modelo e o Número de Arquivos para o Modelo de Mobilidade Sintético.

Além disso, ambos os termos PE e PD dependem de tendências expressadas por EWMA's (conforme visto na Seção 4.2.1), ao passo que as tendências para a mobilidade se baseiam principalmente em um grande número de eventos de contato ao longo do tempo. Esse não é o caso da popularidade, que se baseia em um conjunto limitado de conteúdos, fato que contribui para uma maior precisão das tendências, tendo portanto maior impacto na prioridade.

Reforçamos a importância da popularidade do arquivo com o modelo *Popularidade apenas*, que obteve uma taxa de acerto no *cache* similar à do modelo *PD apenas* para cenários com muitos arquivos. Ainda que similar, a taxa de acerto foi levemente menor (uma diferença de menos de 1% em todos os casos), dessa forma mostrando a necessidade de mantermos o atributo do tamanho do arquivo juntamente com a popularidade no termo PD.

Também é importante notar que computar a prioridade, com base apenas no termo PE, não impede o dispositivo de armazenar um arquivo com tamanho acima do limiar (veja a Seção 4.2.1.1 para mais detalhes), caso os outros arquivos se encaixem em uma situação similar. Portanto, o risco de não ser capaz de transmitir arquivos por meio de comunicação D2D oportunística é maior se não considerarmos outros aspectos (isto é, o termo PD), ao computarmos a prioridade. A indicação desse fenômeno é

possível ao observarmos o modelo *PE apenas* em comparação com o MP-PG completo, sendo esse último superior em quase todos os cenários.

Considerando somente o modelo *Tamanho apenas*, obtivemos a menor taxa de acerto no *cache* na maior parte dos cenários, em comparação com os outros quatro modelos. O emprego desse modelo aumenta a probabilidade de armazenarmos um maior número de arquivos no *cache* do dispositivo. Essa operação ajuda a resolver, pelo menos de forma parcial, o problema de armazenarmos apenas arquivos que o dispositivo pode transmitir durante os contatos D2D.

Entretanto, o fato de o dispositivo não considerar a popularidade de cada arquivo o impede de armazenar alguns arquivos que seus vizinhos podem requisitar no futuro. Além disso, a baixa taxa de acerto no *cache* obtida pelo modelo *Tamanho apenas* também mostra a necessidade de uma avaliação justa dos tamanhos dos arquivos a partir de outro ponto de vista, uma questão que solucionamos por meio do modelo *PE apenas*, que define um limiar de tamanho para comparar com o tamanho de cada arquivo. O modelo resultante supera o modelo *Tamanho apenas* em até 108,48% em termos da taxa de acerto no *cache*, um valor de pico que obtivemos com a aplicação do *trace* de contato *Reality Mining*. Por fim, as figuras de 5.15 a 5.17 mostram a importância crucial de aplicarmos o atributo do tamanho do arquivo em ambos os termos PE e PD, considerando que o modelo *Tamanho apenas* atingiu taxas de acerto no *cache* superiores a 40% das taxas obtidas pelos outros modelos.

No sentido oposto ao do *trace* de contato *Infocom06* e do modelo de mobilidade sintético, com a aplicação do *trace* de contato *Reality Mining* (Figura 5.16), obtivemos as maiores taxas de acerto no *cache* com variantes do MP-PG completo, sendo esse último superior apenas no cenário com 150 arquivos. Isso complementa as observações feitas sobre os resultados da Seção 5.2.1.2, onde esperamos cenários onde o MP-PG completo não provocará impactos significativos na taxa de acerto no *cache*.

Os resultados apresentados na Figura 5.16 mostram que ainda é possível melhorar a taxa de acerto no *cache* desses cenários com variantes do modelo proposto. Conforme apresentado na Seção 5.2.1.2, como trabalho futuro, propomos a definição de um módulo para emprego do modelo mais adequado, para cada cenário.

Apesar de todas as observações acima, é importante ressaltar que o número de usuários não varia entre os cenários acima. Modificar esse parâmetro implica que a densidade da rede também será modificada, uma vez que a área de movimentação dos usuários é limitada nas simulações. Uma densidade de rede maior impacta diretamente na forma como o modelo funciona, como pode ser visto na Figura 5.18, que apresenta a taxa de acerto global no *cache* com o emprego do MP-PG completo e de suas variantes, variando-se o número de usuários na rede.

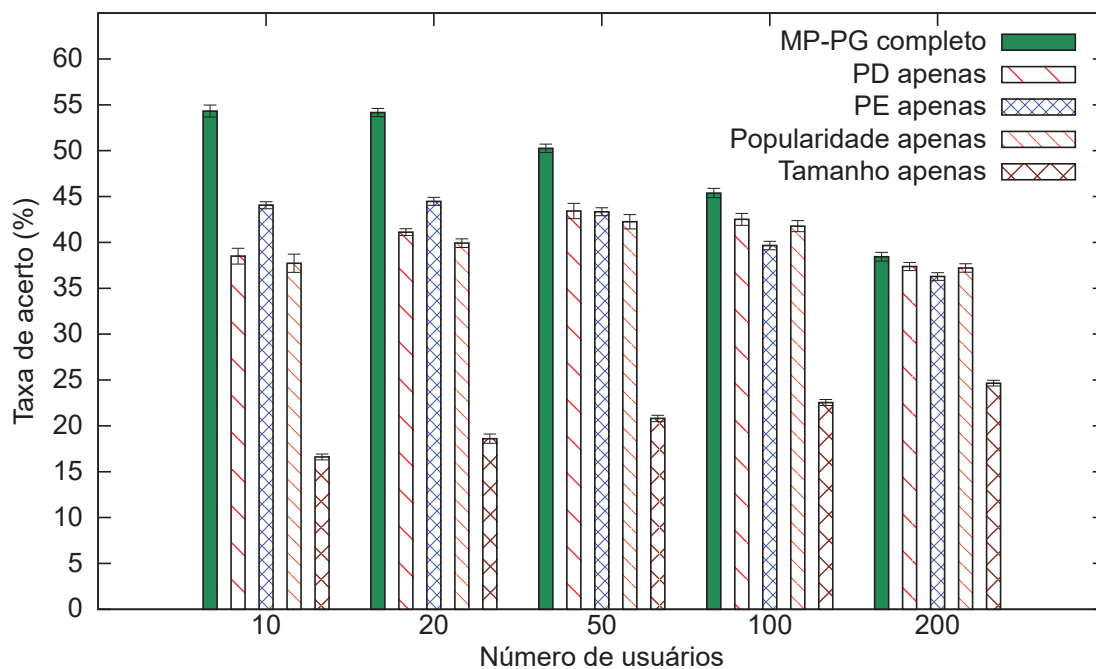


Figura 5.18. Variando o Modelo e o Número de Usuários para o Modelo de Mobilidade Sintético.

Como podemos observar na Figura 5.18, o MP-PG completo superou todos os outros modelos, em concordância com os resultados da Seção 5.2.1.2. Observamos os maiores efeitos positivos para cenários com dez, vinte e 50 arquivos, o que fez o MP-PG ser 23,29%, 21,76% e 15,74% superior à segunda melhor opção respectivamente.

Outro ponto interessante a observarmos na Figura 5.18 é o fato de que o modelo *PE apenas* foi superior aos modelos *PD apenas* e *Popularidade apenas*, para números de usuários inferiores a 50, o que mostra a maior importância do aspecto da mobilidade nesses cenários. É possível explicar tal comportamento ao considerarmos que em cenários com poucos usuários, há uma redução no número de contatos entre eles. Nesses casos, sua capacidade de transmissão por meio de comunicação D2D oportunística tem maior importância, com isso arquivos pequenos recebem valores de prioridade maiores do que aqueles recebidos por arquivos grandes.

Isso não significa que o tamanho do arquivo é suficiente para obtermos uma substituição de conteúdo no *cache* adequada, como pode ser visto ao empregarmos o modelo *Tamanho apenas*, que atingiu a menor taxa de acerto no *cache* em comparação com o modelo *PE apenas*. Mesmo se combinarmos os atributos de mobilidade com o tamanho do arquivo (que foi o caso do modelo *PE apenas*), ainda atingimos taxas de acerto no *cache* menores em comparação com o MP-PG completo, um fato que indica

a importância de combinarmos os atributos de conteúdo e mobilidade em um único modelo.

Considerando o modelo *Tamanho apenas*, sua respectiva taxa de acerto no *cache* se aproximou da taxa obtida para os outros quatro modelos no cenário com 200 usuários. É possível explicar esse comportamento ao lembrar que o ambiente de simulação considera uma área limitada para movimentos, como mencionado anteriormente. Portanto, quando variamos o número de usuários, variamos ao mesmo tempo a densidade da rede.

Se for considerado um cenário com um número fixo de arquivos (que foi o caso para os resultados da Figura 5.18), uma maior densidade implica em uma maior probabilidade de encontrar o que cada dispositivo deseja em sua respectiva vizinhança. Consequentemente, os efeitos positivos ao aplicarmos um modelo para substituição de conteúdo em *cache* caem. Complementamos as observações feitas aqui com a Figura 5.10, onde apresentamos o mesmo comportamento com o emprego do MP-PG completo, juntamente com os *baselines*. Portanto, os resultados apresentados nas figuras 5.18 e 5.10 mostram que um monitoramento da densidade da rede tem um papel chave ao decidirmos o momento certo para empregar um modelo de substituição de conteúdo em *cache*. Como trabalho futuro, propomos estudos mais aprofundados que permitem expressar a relação conjunta entre o MP-PG e a densidade da rede.

5.2.2 MP-CQ

Esta seção apresenta os resultados das simulações com o emprego do MP-CQ. Dividimos as simulações em três partes, cada uma avaliando uma métrica de desempenho específica, conforme descrito na Seção 5.1.4: **(i)** A taxa de acerto global de *chunks* (Seção 5.2.2.1); **(ii)** A taxa de qualidade de *chunks* (Seção 5.2.2.2) e **(iii)** A sobrecarga de controle (Seção 5.2.2.3). Em cada seção, comparamos o MP-CQ com os *baselines* LRS e SACC, conforme descrito na Seção 5.1.2.

5.2.2.1 Taxa de Acerto Global de *Chunks*

As figuras de 5.19 a 5.21 mostram a taxa de acerto global de *chunks* para os *baselines* LRS, SACC e o MP-CQ ao variar o número de arquivos de vídeo, com a aplicação do *trace* de contato *Reality Mining*, o modelo de mobilidade sintético e o *trace* de contato *Infocom06* respectivamente.

O MP-CQ superou os *baselines* em todos os cenários ao aplicarmos o *trace* de contato *Reality Mining* e o modelo de mobilidade sintético (figuras 5.19 e 5.20

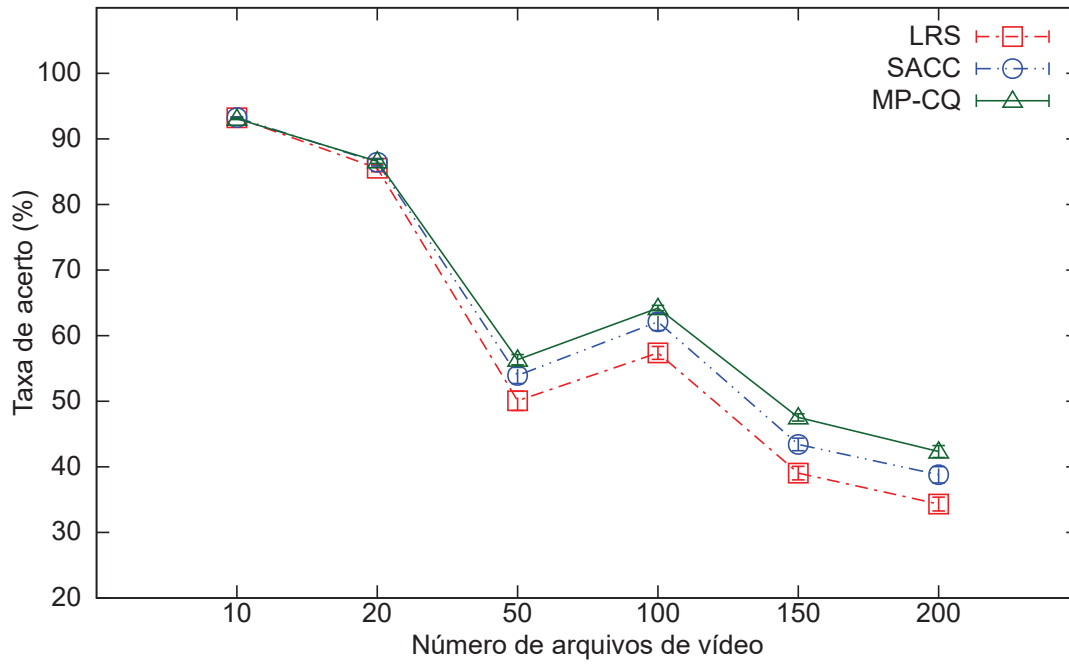


Figura 5.19. Taxa de Acerto Global de *Chunks* com a Variação do Número de Arquivos de Vídeo para o *Trace* de Contato *Reality Mining*.

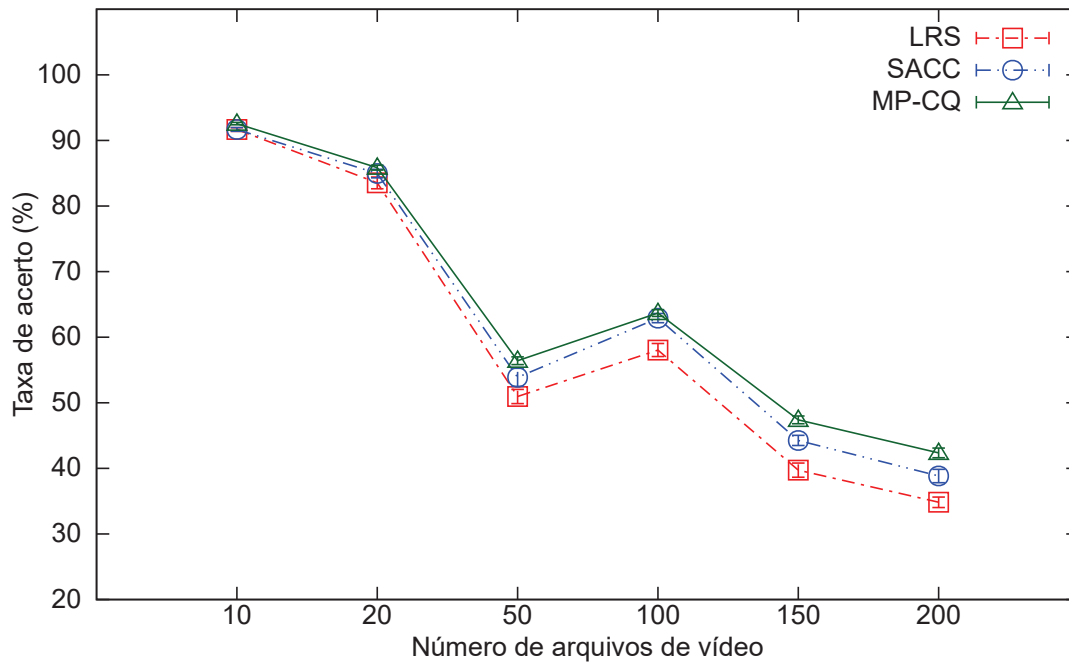


Figura 5.20. Taxa de Acerto Global de *Chunks* com a Variação do Número de Arquivos de Vídeo para o Modelo de Mobilidade Sintético.

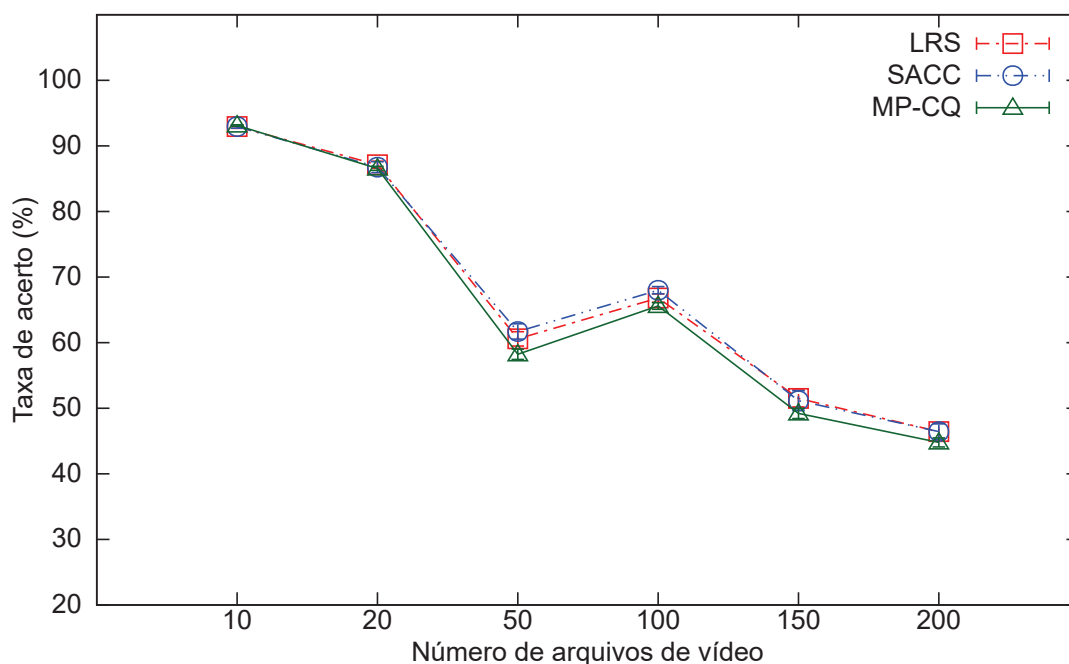


Figura 5.21. Taxa de Acerto Global de *Chunks* com a Variação do Número de Arquivos de Vídeo para o *Trace* de Contato *Infocom06*.

respectivamente). Em ambas as figuras, a superioridade do MP-CQ aumenta à medida em que o número de arquivos de vídeo na rede também aumenta. No cenário com 200 arquivos de vídeo, o MP-CQ superou o LRS em 23,34% e 21,61% e o SACC em 9,11% e 9,07% ao aplicarmos os *traces* de contato *Reality Mining* e o modelo de mobilidade sintético, respectivamente.

Além disso, observamos que todas as soluções proveem uma taxa de acerto de *chunks* superior a 90% para cenários com poucos arquivos de vídeo. Podemos justificar esse fato pela habilidade do dispositivo em armazenar a maior parte dos arquivos de vídeo no *cache* nas primeiras horas da simulação, dispensando a necessidade de novos *downloads* a partir do servidor.

À medida em que o número de arquivos de vídeo aumenta, a taxa de acerto global de *chunks* diminui para todas as soluções. Esse também é um comportamento esperado, uma vez que cada dispositivo requisita uma maior variedade de arquivos, o que reduz a probabilidade de se encontrar um arquivo de vídeo na vizinhança D2D ou no *cache* do dispositivo. Além disso, mais operações de substituição de conteúdo em *cache* tendem a ocorrer em todos os dispositivos.

Ainda com relação ao decréscimo na taxa de acerto global de *chunks*, para todas as soluções, verificamos que o decréscimo observado entre os cenários

com 150 e 200 arquivos é menor do que o observado entre os cenários com 100 e 150 arquivos. Isso sugere que melhorias mais significativas não serão observadas em cenários com mais arquivos, estabilizando dessa forma o desempenho para todas as soluções.

Por fim, o MP-CQ e os *baselines* apresentaram desempenho similar com a aplicação do *trace* de contato *Infocom06* (Figura 5.21). Tal fato indica a influência do modelo de mobilidade na capacidade de uma solução de substituição de conteúdo em melhorar o emprego de comunicação D2D e do *cache*. Ainda que neste trabalho tenhamos avaliado o desempenho do MP-CQ com diferentes modelos de mobilidade, não há um foco em quantificar e parametrizar o impacto desses modelos. Como trabalho futuro, propomos estudos mais aprofundados sobre a influência do modelo de mobilidade, bem como a representação desse modelo na forma de um ou mais atributos no procedimento de substituição de conteúdo em *cache*.

A Figura 5.22 mostra a taxa de acerto global de *chunks* para os *baselines* e o MP-CQ quando variamos o número de usuários. À medida em que o número de usuários diminui, a superioridade do MP-CQ e do SACC em relação ao LRS aumenta, o que mostra a importância de se considerar a relação conjunta entre atributos de mobilidade e conteúdo nesses cenários. No cenário com dez usuários, o SACC e o MP-CQ superaram o LRS em 158,35% e 159,17% respectivamente. A diferença cai para 7,2% e 5,94% no cenário com 200 usuários.

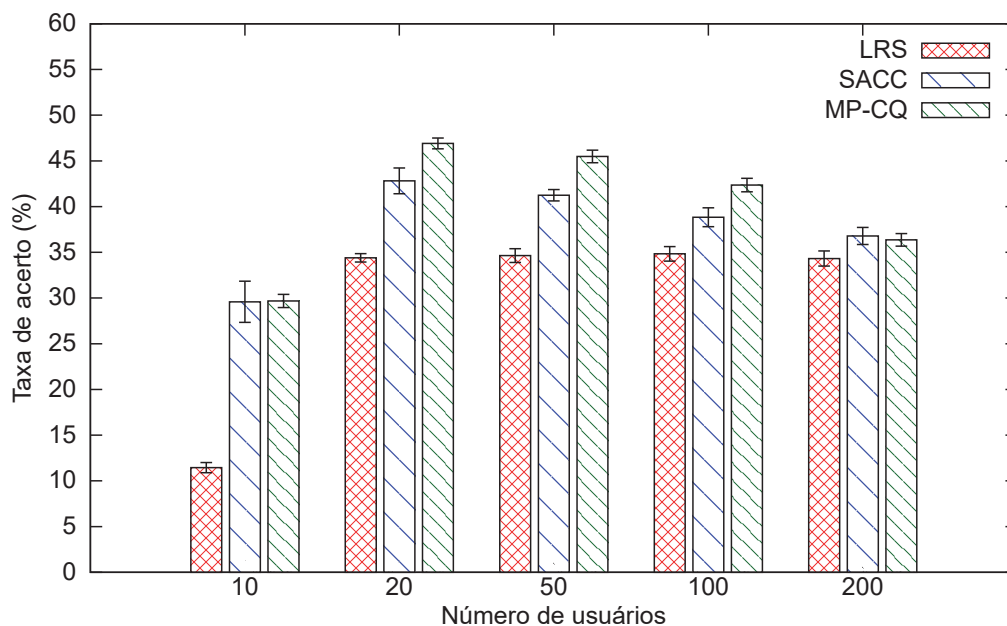


Figura 5.22. Taxa de Acerto Global de *Chunks* com a Variação do Número de Usuários para o Modelo de Mobilidade Sintético.

No cenário com dez usuários, apesar do desempenho do SACC e do MP-CQ terem sido estatisticamente similares, o comportamento do MP-CQ se mostrou mais previsível, um fato que observamos com base nos intervalos de confiança mais curtos. Além disso, é possível explicar o aumento repentino da taxa de acerto de *chunks*, para todas as soluções entre os cenários com dez e vinte usuários, por meio do posicionamento dos usuários pelo simulador no início de cada execução. Alguns usuários podem ser posicionados em locais que reduzem sua probabilidade de encontro com outros usuários, o que pode causar um impacto negativo na taxa de acerto de *chunks*. Mesmo assim, à medida em que o número de usuários aumenta, esse efeito reduz drasticamente e todas as soluções passam a apresentar um comportamento mais regular, com a queda na taxa de acerto de *chunks*.

A queda na taxa de acerto de *chunks* com o aumento no número de usuários é outro comportamento interessante a observarmos. É possível explicar esse fenômeno por dois motivos. O primeiro deles é o fato de que os usuários possuem um raio de comunicação limitado, e o segundo motivo é a possibilidade de se compartilharem *chunks* via D2D com apenas um vizinho por vez. Conseqüentemente, em cenários com um maior número de usuários há uma “disputa” pelo canal D2D. Isso reduz o número de vizinhos disponíveis e conseqüentemente a taxa de acerto de *chunks*. Esse fenômeno também leva a uma queda nos efeitos de uma solução de substituição de conteúdo em *cache*, o que também pôde ser observado no cenário com 200 usuários, onde o desempenho do SACC e do MP-CQ foram estatisticamente equivalentes.

A mesma afirmação acima não vale para um cenário onde a área de movimentação dos usuários varia ao longo do tempo, uma vez que a probabilidade de encontro de dispositivos e a densidade da rede irão mudar simultaneamente. Como trabalho futuro, propomos um estudo mais aprofundado a respeito dos efeitos da área de movimentação sobre a taxa de acerto global de *chunks*.

5.2.2.2 Taxa de Qualidade de *Chunks*

As figuras de 5.23 a 5.25 apresentam a taxa de qualidade de *chunks* ao empregarmos o LRS, o SACC e o MP-CQ com a variação do número de arquivos de vídeo, aplicando os *traces Infocom06* e *Reality Mining* e o modelo de mobilidade sintético respectivamente.

Na maior parte dos cenários, houve um aumento na porcentagem de *chunks* de maior qualidade com o emprego do MP-CQ, em comparação com os *baselines*. Esse aumento é ainda mais evidente com a aplicação do *trace* de contato *Infocom06*, onde a taxa de qualidade para a resolução de 1080p foi 110,22% e 109,53% superior às taxas observadas com o emprego do LRS e do SACC respectivamente, no cenário com dez

usuários.

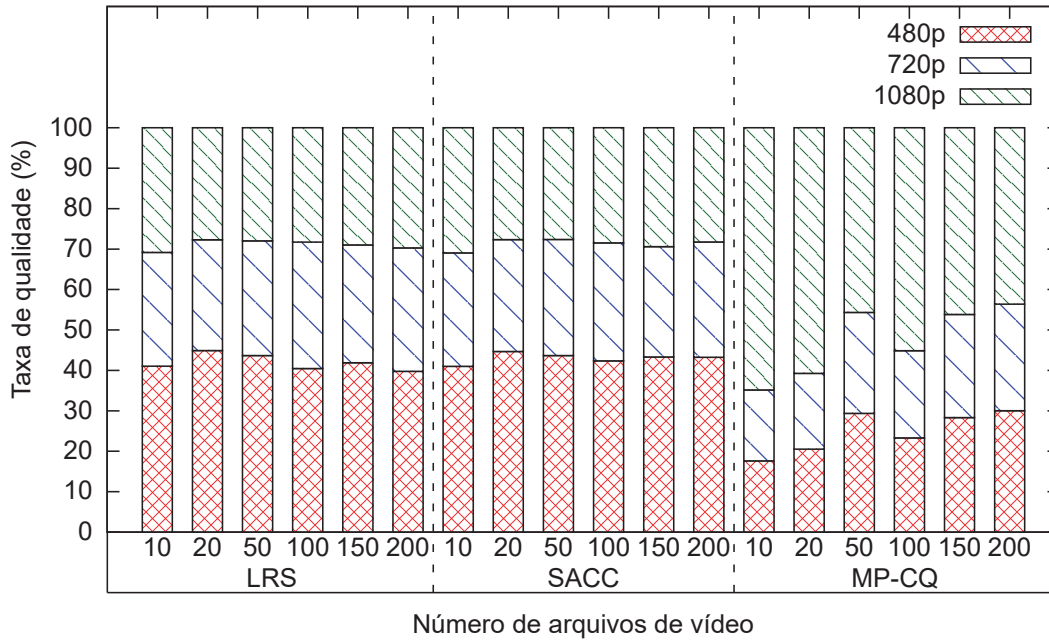


Figura 5.23. Taxa de Qualidade de *Chunks* com a Variação do Número de Arquivos de Vídeo para o *Trace* de Contato *Infocom06*.

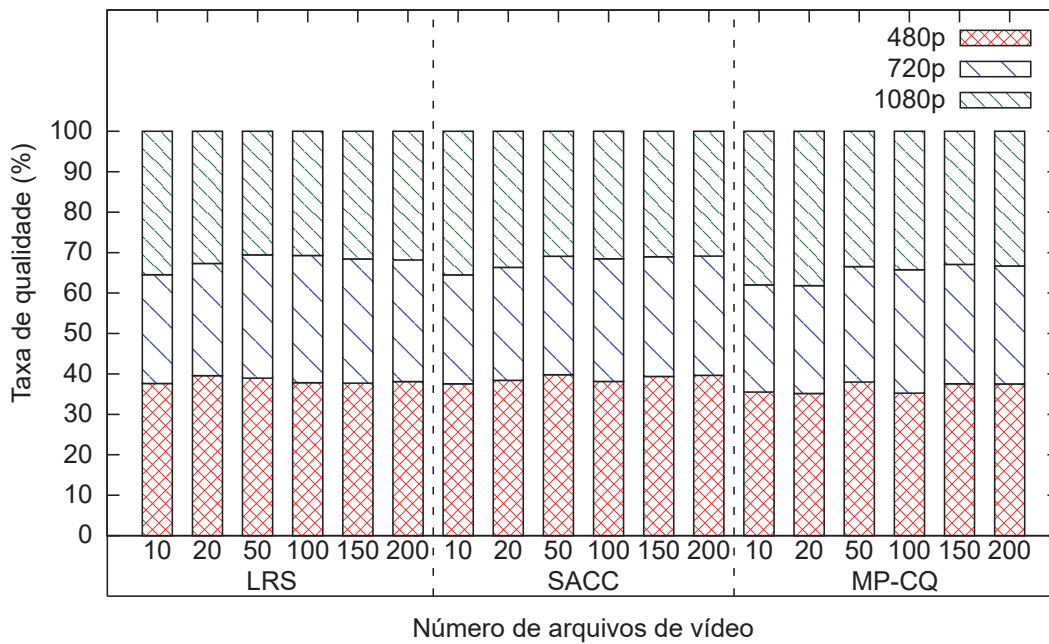


Figura 5.24. Taxa de Qualidade de *Chunks* com a Variação do Número de Arquivos de Vídeo para o *Trace* de Contato *Reality Mining*.

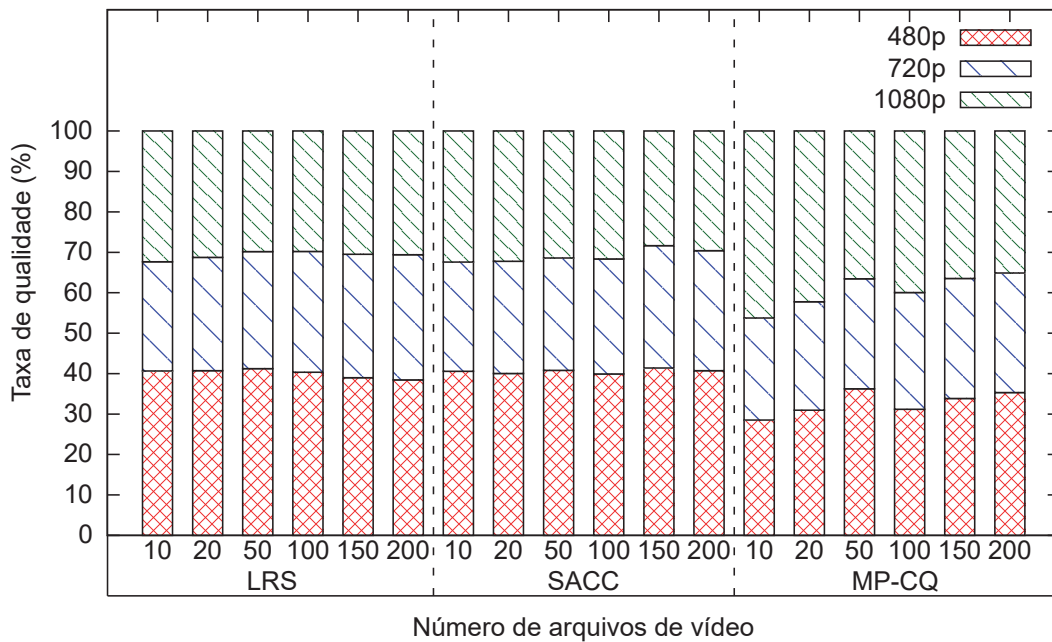


Figura 5.25. Taxa de Qualidade de *Chunks* com a Variação do Número de Arquivos de Vídeo para o Modelo de Mobilidade Sintético.

Além disso, como esperado, há uma tendência de queda na qualidade para todas as soluções, à medida em que o número de arquivos na rede aumenta. A principal razão para esse fenômeno é que uma maior variedade de arquivos na rede aumenta a disputa por espaço no *cache* dos dispositivos. Com isso, visando preservar a maior taxa de acerto no *cache* possível, mais *chunks* com menor qualidade tendem a se manter no *cache*.

A Figura 5.26 apresenta a taxa de qualidade no *cache* com a variação do número de usuários. O MP-CQ atingiu as maiores taxas de qualidade em cenários com poucos usuários, de forma similar ao ocorrido com a taxa de acerto global de *chunks* com a variação do número de usuários (Figura 5.22). As maiores taxas de qualidade observadas nesses cenários mostram a importância de aplicarmos um modelo que considera atributos de mobilidade e conteúdo, de forma a melhorar o emprego de comunicação D2D oportunística e do *cache*, além da qualidade do vídeo consumido. O MP-CQ superou o LRS e o SACC em 54,07% e 52,07% respectivamente, em cenários com dez usuários em termos da taxa de qualidade para a resolução de 1080p.

Ao empregarmos o LRS e o SACC, não observamos variações significativas na taxa de qualidade de *chunks* com o aumento do número de usuários. Com o emprego do MP-CQ, houve uma queda gradual na taxa de qualidade de *chunks* para a resolução de 1080p e um aumento gradual para a resolução de 480p. Ao mesmo tempo, houve

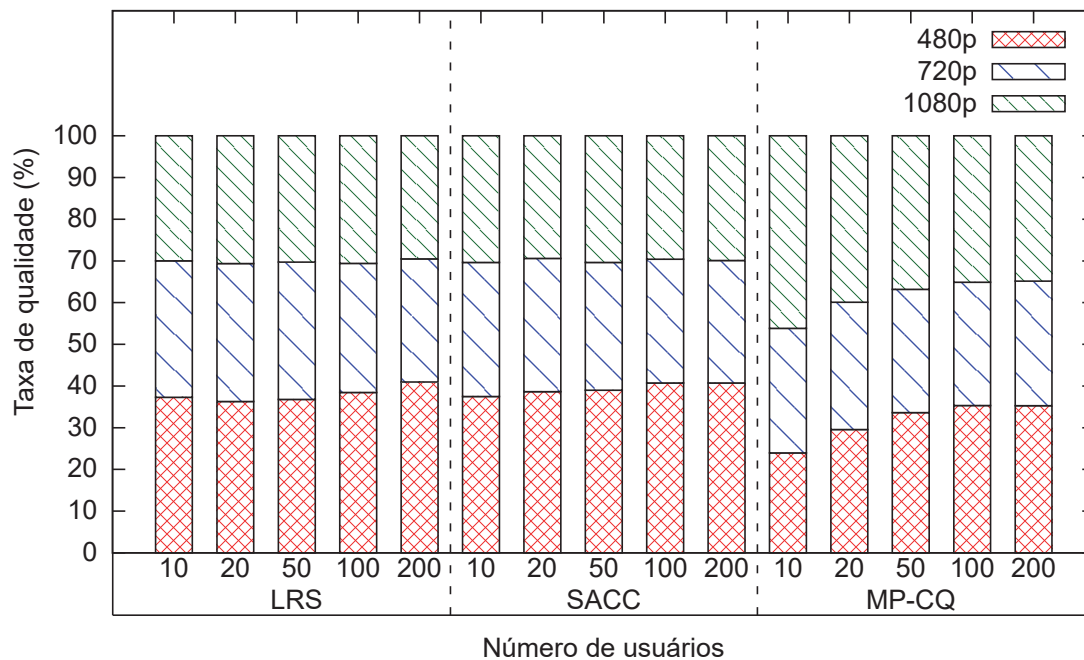


Figura 5.26. Taxa de Qualidade de *Chunks* com a Variação do Número de Usuários para o Modelo de Mobilidade Sintético.

uma leve estabilização para todas as taxas de qualidade. Tal comportamento indica uma tendência a se obter um desempenho similar em cenários com mais de 200 usuários, mas ao mesmo tempo mantendo a superioridade do MP-CQ em comparação com os *baselines*. É importante notar que essa superioridade sofre uma queda no cenário com 200 usuários para a resolução de 1080p (18,15% e 16,63% sobre o LRS e o SACC respectivamente) em comparação com o que observamos no cenário com dez usuários para a mesma resolução. Podemos mais uma vez explicar esse fato pela limitação de movimento dos usuários em uma área de simulação fixa. Também observamos esse comportamento nos resultados para a taxa de acerto global de *chunks* (Seção 5.2.2.1).

5.2.2.3 Sobrecarga de Controle

Esta seção avalia o MP-CQ e os *baselines* de acordo com a sobrecarga de mensagens de controle e a sobrecarga de bytes de controle, conforme o comportamento de envio descrito na Seção 5.1.5. Limitamos a apresentação dos resultados ao modelo de mobilidade sintético para fins de simplificação.

A Figura 5.27 apresenta a sobrecarga de mensagens de controle com o emprego do MP-CQ e dos *baselines* com a variação do número de arquivos de vídeo. O LRS obteve a menor sobrecarga de mensagens de controle entre todas as soluções. Isso é

devido ao fato de que ele dispensa a necessidade de mensagens de monitoramento, o que leva a uma sobrecarga de mensagens de controle baixa e quase constante, composta apenas por requisições de conteúdo.

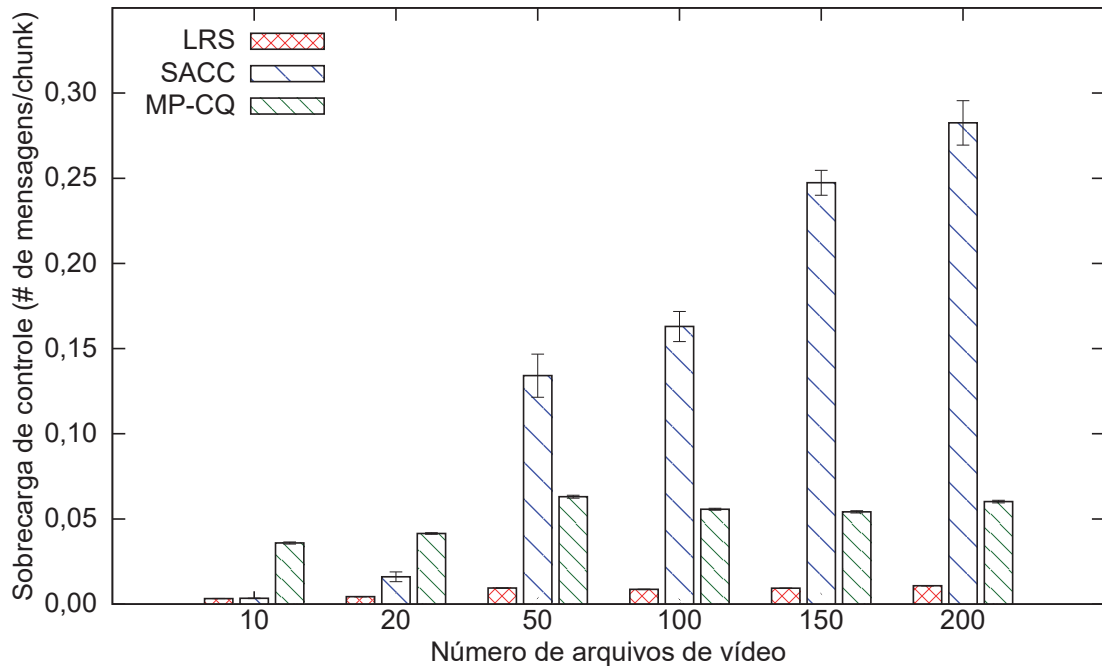


Figura 5.27. Sobrecarga de Mensagens de Controle com a Variação do Número de Arquivos de Vídeo para o Modelo de Mobilidade Sintético.

Por outro lado, o SACC apresentou um crescimento linear na sobrecarga de mensagens de controle, com o aumento no número de arquivos de vídeo. Em cenários com um número fixo de usuários, que foi o caso nessa parte das simulações, há uma queda na probabilidade de se encontrar o arquivo desejado na vizinhança D2D ou no *cache*, dessa forma aumentando o número de requisições de conteúdo enviadas para o servidor. Observamos esse comportamento na Seção 5.2.2.1, onde a taxa de acerto global de *chunks* sofre uma queda com o aumento do número de arquivos de vídeo.

Além disso, uma maior variedade de arquivos de vídeo na rede significa que mais operações de substituição de conteúdo no *cache* tendem a ocorrer. Tal fato demanda a transmissão de mais mensagens de monitoramento na rede, o que também contribui para aumentar a sobrecarga de mensagens de controle.

Por fim, o MP-CQ também apresentou uma tendência de crescimento com o aumento no número de arquivos de vídeo. Entretanto, esse crescimento foi consideravelmente inferior ao que observamos com o emprego do SACC. No cenário

com 200 arquivos de vídeo, o MP-CQ demandou em torno de 78,69% menos mensagens de controle de forma a consumir um *chunk*. Isso reflete o fato de que o MP-CQ sempre possui custos de monitoramento constantes, para cada *download* via D2D ou do servidor, bem como para cada operação de substituição de conteúdo no *cache*.

A Figura 5.28 apresenta a sobrecarga de mensagens de controle ao empregarmos o MP-CQ e os *baselines* com a variação do número de usuários. De forma similar à Figura 5.27 onde o variamos o número de arquivos de vídeo, o MP-CQ e o SACC apresentaram um comportamento de crescimento linear com o aumento do número de usuários. Esse é um comportamento esperado, uma vez que um maior número de usuários implica no envio de mais requisições de conteúdo e mensagens de monitoramento na rede.

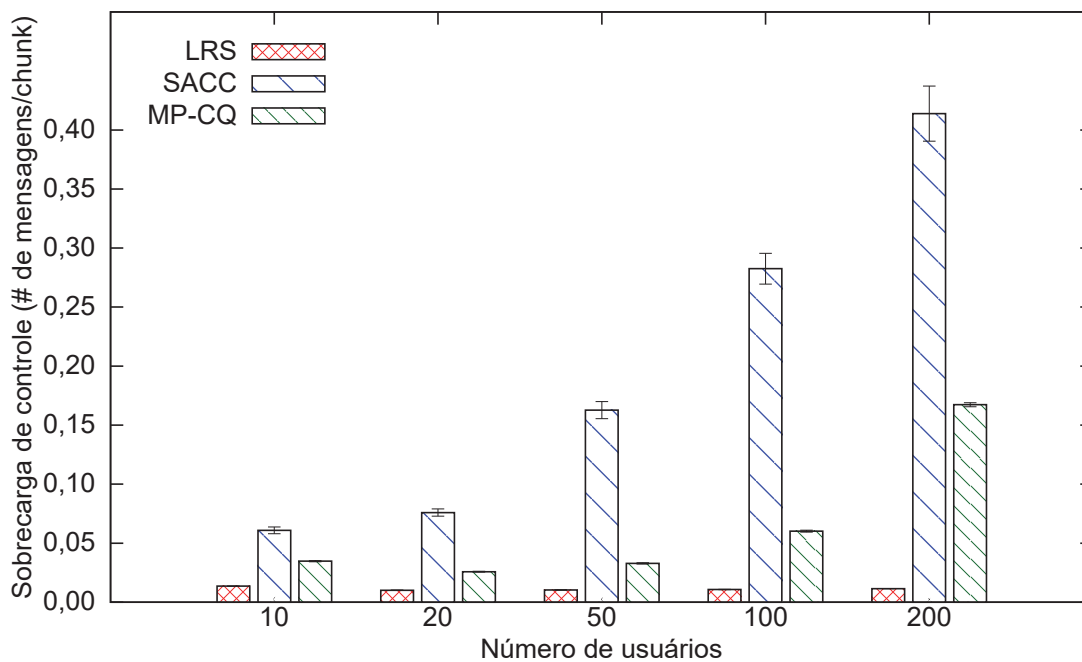


Figura 5.28. Sobrecarga de Mensagens de Controle com a Variação do Número de Usuários para o Modelo de Mobilidade Sintético.

Além disso, o fato de se ter um maior número de usuários com raio de comunicação D2D limitado, bem como a limitação de uma transmissão D2D para cada par de usuários, significa que mais “disputas” pelo canal D2D tendem a ocorrer, o que pode forçar alguns usuários a fazerem *downloads* a partir do servidor. Mesmo assim, no cenário com 200 usuários, o MP-CQ demandou cerca de 59,57% menos mensagens de controle em comparação com o SACC. Isso mostra a eficiência do

procedimento de substituição de conteúdo em diminuir a quantidade de requisições ao servidor, ao mesmo tempo com a transmissão de um volume menor de mensagens de monitoramento.

A Figura 5.29 mostra a sobrecarga de bytes de controle com o emprego do MP-CQ e dos *baselines* quando variamos o número de arquivos de vídeo. Todas as soluções apresentaram um comportamento de estabilização na maior parte dos cenários. É possível explicar esse fenômeno pelo fato de que à medida em que o número de arquivos de vídeo aumenta, há também uma queda na probabilidade de se encontrar o arquivo de vídeo desejado na vizinhança D2D ou no *cache*, o que causa um aumento no número de requisições de conteúdo no servidor. Além disso, um maior número de arquivos tende a aumentar o número de operações de substituição de conteúdo no *cache*, uma vez que o usuário tende a requisitar uma maior variedade de arquivos de vídeo, o que significa que mais mensagens de monitoramento serão enviadas. Todos os fatores acima contribuem para uma estabilização no número de bytes de controle enviados na rede.

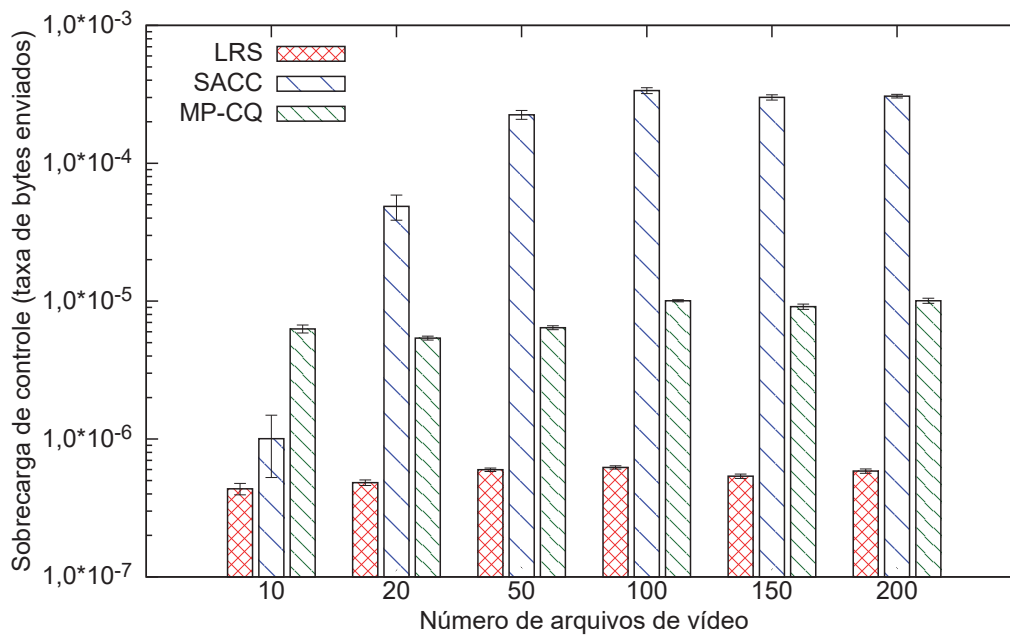


Figura 5.29. Sobrecarga de Bytes de Controle com a Variação do Número de Arquivos de Vídeo para o Modelo de Mobilidade Sintético.

Mesmo apresentando um comportamento de estabilização, o MP-CQ reduziu a sobrecarga de mensagens de controle em torno de 96,72% no cenário com 200 arquivos de vídeo. Isso mais uma vez mostra a eficácia do MP-CQ em preservar arquivos de vídeo de maior qualidade, com a troca de mensagens de controle em menor volume na rede.

A Figura 5.30 apresenta a sobrecarga de bytes de controle com o emprego do MP-CQ e dos *baselines* ao variarmos o número de usuários. Como podemos observar, o LRS apresenta um comportamento quase constante, composto apenas por requisições de conteúdo, conforme descrito na Seção 5.1.5.

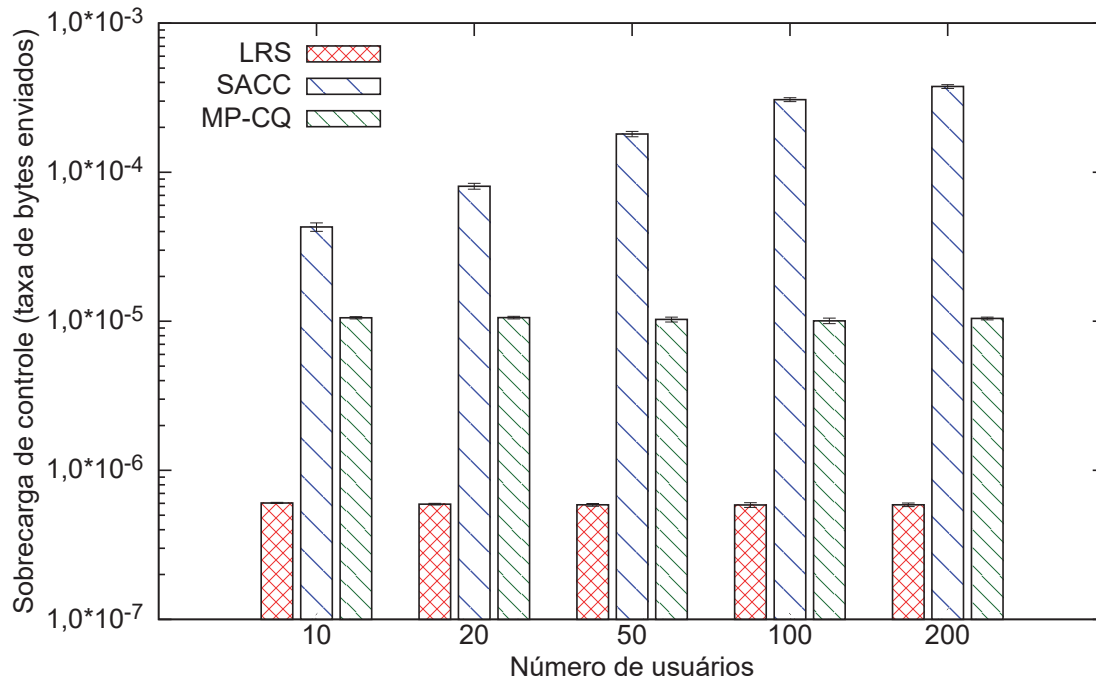


Figura 5.30. Sobrecarga de Bytes de Controle com a Variação do Número de Usuários para o Modelo de Mobilidade Sintético.

Por outro lado, o SACC apresentou um crescimento linear. Isso se deve ao fato de que um crescimento no número de usuários implica em mais conexões entre eles. Tal fator aumenta os custos de monitoramento do SACC, demandando a transmissão de uma quantidade mais substancial de dados de monitoramento para o servidor centralizado.

Em comparação com o SACC, o MP-CQ obteve sobrecargas de bytes de controle menores com o aumento do número de usuários. A queda foi de 75,4% no cenário com dez usuários, ao passo que com 200 usuários a queda foi de 97,22%.

5.3 Conclusão

Este capítulo apresentou a avaliação do *Modelo de Prioridade de Propósito Geral* (MP-PG) e do *Modelo de Prioridade Ciente de Qualidade* (MP-CQ), por meio de

simulações. Comparamos os modelos com uma solução clássica, denominada *Least Recently Stored* (LRS), que não considera a relação conjunta entre atributos de mobilidade e conteúdo. Também foram feitas comparações com uma solução do estado da arte, denominada *Social-Aware Cooperative Caching* (SACC), que considera a relação conjunta entre esses atributos sob uma abordagem distinta. Ambas as soluções também não consideram o fator de qualidade, o que ocorre no presente trabalho por meio do MP-CQ.

Os resultados mostraram que o MP-PG obteve uma taxa de acerto global no *cache* quase 80% superior em cenários com 200 arquivos e em torno de 420% superior para cenários com dez usuários, em comparação com o LRS e o SACC. Comparado com o SACC, o MP-PG reduziu em até 80% a sobrecarga de mensagens de controle e em até 90% a sobrecarga de bytes de controle.

Focando também em melhorar a qualidade do conteúdo, o MP-CQ proveu uma taxa de acerto de *chunks* em torno de 23% superior em cenários com 200 arquivos de vídeo e cerca de 160% superior em cenários com dez usuários. Além disso, o MP-CQ obteve uma taxa de qualidade de *chunks* quase 110% superior, em comparação com o LRS e o SACC. Comparado com o SACC, o MP-CQ reduziu em até cerca de 80% a sobrecarga de mensagens de controle e em até cerca de 97% a sobrecarga de bytes de controle.

Capítulo 6

Conclusões

Este trabalho teve como foco principal o problema de gerenciamento do *cache* dos dispositivos, visando definir o melhor conteúdo a se manter em *cache*. Dessa forma, aumenta-se o emprego de comunicação D2D oportunística e do *cache* dos dispositivos, bem como a qualidade do conteúdo recebido.

Foram propostos dois modelos de prioridade, que os dispositivos empregam de forma independente para realização de substituição de conteúdo no *cache* quando este está sobrecarregado. O primeiro modelo, denominado *Modelo de Prioridade de Propósito Geral* (MP-PG), considera conteúdos genéricos não segmentados e de qualidade fixa. Resultados de simulação mostraram que o MP-PG foi capaz de prover uma taxa de acerto global no *cache* quase 80% superior em cenários com muitos conteúdos distintos, e em torno de 420% superior para cenários com poucos usuários em comparação com os *baselines*. Comparado com a solução do estado da arte, o MP-PG reduziu em até 80% a sobrecarga de mensagens de controle, e em até 90% a sobrecarga de bytes de controle. Os resultados deste trabalho foram publicados no *Journal of Network and Systems Management* [Silva et al., 2021].

Já o segundo modelo, aqui denominado *Modelo de Prioridade Ciente de Qualidade* (MP-CQ), foca em cenários em que o conteúdo se divide em segmentos (*chunks*), que o servidor de conteúdo pode prover em qualidade variável ao longo do tempo, de acordo com as condições da rede. De forma similar ao MP-PG, o MP-CQ combina atributos de mobilidade e conteúdo, em especial a qualidade desse último, visando não somente aumentar o emprego da comunicação D2D oportunística e do *cache*, como também aumentar a qualidade do conteúdo fornecido.

Resultados de simulação, focados em aplicações de *streaming* de vídeo adaptativo, mostraram que o MP-CQ proveu uma taxa de acerto de *chunks* em torno de 23% superior em cenários com vários arquivos de vídeo, e cerca de 160% superior em cenários

com poucos usuários. Além disso, o MP-CQ foi capaz de prover uma qualidade de *chunks* quase 110% superior em comparação com os *baselines*. Comparado à solução do estado da arte, o MP-CQ reduziu em até cerca de 80% a sobrecarga de mensagens de controle e em até cerca de 97% a sobrecarga de bytes de controle.

6.1 Trabalhos Futuros

Com relação ao MP-PG, as simulações que ajustam o intervalo de monitoramento do modelo mostraram que, para cada cenário, deverá haver um respectivo intervalo ótimo, demandando com isso um ajuste dinâmico desse parâmetro. É necessário que estudos mais aprofundados sejam feitos visando desenvolver um módulo de ajuste dinâmico do intervalo de monitoramento.

Ainda com relação ao MP-PG, em alguns cenários, variantes do modelo ofereceram taxas de acerto superiores às do modelo completo. Por outro lado, em determinados cenários o modelo completo e suas variantes não ofereceram melhorias significativas na taxa de acerto. Portanto, é preciso que seja desenvolvido um módulo de monitoramento que empregue o MP-PG em cada dispositivo de usuário no momento adequado, bem como trocar para variantes desse modelo.

Já com relação ao MP-CQ, alguns resultados obtidos sugerem que há uma influência direta da mobilidade dos usuários e seu comportamento de consumo na sua habilidade em encontrar o conteúdo desejado na vizinhança D2D ou em seu respectivo *cache*, bem como na qualidade do conteúdo recebido por eles. Como trabalho futuro, propõe-se uma avaliação mais aprofundada sobre os efeitos do MP-CQ, considerando outros comportamentos de consumo e mobilidade.

Por fim, com relação a ambos os modelos de prioridade desenvolvidos, são necessários estudos futuros focados em cenários onde o número de arquivos e usuários (inclusive a densidade desses na região monitorada) se altera de forma dinâmica e simultânea. Além disso, também são necessários estudos variando-se o comportamento de busca e *download* de conteúdo dos usuários. Tais estudos têm como objetivo estabelecer de forma mais clara a relação entre esses parâmetros no desempenho dos modelos propostos.

Agradecimentos

Este trabalho foi desenvolvido por meio de recursos provenientes das agências de fomento CAPES, CNPq, Fapemig, e o projeto FUTEBOL. O projeto FUTEBOL

recebeu recursos do programa Horizon 2020 da União Europeia, via contrato número 688941 (FUTEBOL), bem como recursos do Ministério da Ciência, Tecnologia e Inovações (MCTI) via RNP e CTIC.

Referências Bibliográficas

- 3GPP (2015). Overview of 3GPP Release 12. Technical report (tr), 3rd Generation Partnership Project (3GPP). Version 0.2.0.
- Ali, A. S.; Mahmoud, K. R. & Naguib, K. M. (2020). Optimal Caching Policy for Wireless Content Delivery in D2D Networks. *Journal of Network and Computer Applications*, 150:102467. ISSN 1084-8045.
- Andreev, S.; Galinina, O.; Pyattaev, A.; Johnsson, K. & Koucheryavy, Y. (2015). Analyzing Assisted Offloading of Cellular User Sessions onto D2D Links in Unlicensed Bands. *IEEE Journal on Selected Areas in Communications*, 33(1):67–80. ISSN 0733-8716.
- Andreev, S.; Pyattaev, A.; Johnsson, K.; Galinina, O. & Koucheryavy, Y. (2014). Cellular Traffic Offloading onto Network-Assisted Device-To-Device Connections. *IEEE Communications Magazine*, 52(4):20–31. ISSN 0163-6804.
- Asadi, A.; Wang, Q. & Mancuso, V. (2014). A Survey on Device-to-Device Communication in Cellular Networks. *IEEE Communications Surveys and Tutorials*, 16(4):1801–1819. ISSN 1553-877X.
- Baccour, E.; Erbad, A.; Bilal, K.; Mohamed, A. & Guizani, M. (2020). PCCP: Proactive Video Chunks Caching and Processing in Edge Networks. *Future Generation Computer Systems*, 105:44 – 60. ISSN 0167-739X.
- Bao, X.; Lin, Y.; Lee, U.; Rimal, I. & Choudhury, R. R. (2013). Dataspotting: Exploiting Naturally Clustered Mobile Devices to Offload Cellular Traffic. In *2013 Proceedings IEEE INFOCOM*, pp. 420–424. IEEE.
- Bouazizi, I.; Jarvinen, K. & Hannuksela, M. (2010). 3GPP Mobile Multimedia Services [Standards in a Nutshell]. *IEEE Signal Processing Magazine*, 27(5):125–130.

- Burleigh, S.; Hooke, A.; Torgerson, L.; Fall, K.; Cerf, V.; Durst, B.; Scott, K. & Weiss, H. (2003). Delay-Tolerant Networking: An Approach to Interplanetary Internet. *IEEE Communications Magazine*, 41(6):128–136.
- Camps-Mur, D.; Garcia-Saavedra, A. & Serrano, P. (2013). Device-To-Device Communications with Wi-Fi Direct: Overview and Experimentation. *IEEE Wireless Communications*, 20(3):96–104.
- Chandrasekaran, G.; Wang, N.; Hassanpour, M.; Xu, M. & Tafazolli, R. (2018). Mobility as a Service (MaaS): A D2D-Based Information Centric Network Architecture for Edge-Controlled Content Distribution. *IEEE Access*, 6:2110–2129.
- Cisco (2020). White Paper: Cisco Annual Internet Report (2018-2023). Relatório técnico, Cisco.
- Doppler, K.; Rinne, M.; Wijting, C.; Ribeiro, C. & Hugl, K. (2009). Device-To-Device Communication as an Underlay to LTE-Advanced Networks. *IEEE Communications Magazine*, 47(12):42–49.
- Duanmu, Z.; Zeng, K.; Ma, K.; Rehman, A. & Wang, Z. (2017). A Quality-of-Experience Index for Streaming Video. *IEEE Journal of Selected Topics in Signal Processing*, 11(1):154–166.
- Eagle, N. & Pentland, A. S. (2005). CRAWDDAD dataset mit/reality (v. 2005-07-01). Downloaded from <https://crawdad.org/mit/reality/20050701>.
- Ge, C.; Wang, N.; Skillman, S.; Foster, G. & Cao, Y. (2016). QoE-Driven DASH Video Caching and Adaptation at 5G Mobile Edge. In *Proceedings of the 3rd ACM Conference on Information-Centric Networking, ACM-ICN '16*, p. 237–242, New York, NY, USA. Association for Computing Machinery.
- Gupta, A. & Jha, R. K. (2015). A Survey of 5G Network: Architecture and Emerging Technologies. *IEEE Access*, 3:1206–1232.
- Haldar, K. L.; Li, H. & Agrawal, D. P. (2013). A Cluster-Aware Soft Frequency Reuse Scheme for Inter-Cell Interference Mitigation in LTE Based Femtocell Networks. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6. IEEE.
- Hosny, S.; Eryilmaz, A. & El Gamal, H. (2016). Impact of User Mobility on D2D Caching Networks. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6. ISSN .

- Hui, P.; Chaintreau, A.; Scott, J.; Gass, R.; Crowcroft, J. & Diot, C. (2005). Pocket Switched Networks and Human Mobility in Conference Environments. In *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking, WDTN '05*, p. 244–251, New York, NY, USA. Association for Computing Machinery.
- Keränen, A.; Ott, J. & Kärkkäinen, T. (2009). The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA. ICST.
- Krishnan, S. S. & Sitaraman, R. K. (2013). Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs. *IEEE/ACM Transactions on Networking*, 21(6):2001–2014. ISSN 1558-2566.
- Kua, J.; Armitage, G. & Branch, P. (2017). A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP. *IEEE Communications Surveys and Tutorials*, 19(3):1842–1866.
- Li, C.; Gong, S.; Ning, Z.; Zhao, L.; Jiang, Q. & Zhou, P. (2018). Robust Caching Control in Crowdsourced Content-Centric Mobile Networking. *IEEE Access*, 6:59811–59821.
- Lin, X.; Andrews, J. G.; Ghosh, A. & Ratasuk, R. (2014). An Overview of 3GPP Device-To-Device Proximity Services. *IEEE Communications Magazine*, 52(4):40–48. ISSN 0163-6804.
- Liu, J.; Kato, N.; Ma, J. & Kadowaki, N. (2015). Device-To-Device Communication in LTE-Advanced Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 17(4):1923–1940.
- Mehr, S. K. & Medhi, D. (2019). QoE Performance for DASH Videos in a Smart Cache Environment. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 388–394.
- Mehrabi, A.; Siekkinen, M.; Illahi, G. & Ylä-Jääski, A. (2019a). D2D-Enabled Collaborative Edge Caching and Processing with Adaptive Mobile Video Streaming. In *2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 1–10.
- Mehrabi, A.; Siekkinen, M. & Ylä-Jääski, A. (2019b). Green Edge Adaptive Mobile Video Streaming with Device-to-Device Communication. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*

- 2019, pp. 1--7, United States. IEEE Communications Society. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC ; Conference date: 08-09-2019 Through 11-09-2019.
- Mota, V. F.; Cunha, F. D.; Macedo, D. F.; Nogueira, J. M. & Loureiro, A. A. (2014). Protocols, Mobility Models and Tools in Opportunistic Networks: A Survey. *Computer Communications*, 48:5 – 19. ISSN 0140-3664. Opportunistic Networks.
- Mota, V. F. S.; Macedo, D. F.; Ghamri-Doudane, Y. & Nogueira, J. M. S. (2013). On the Feasibility of WiFi Offloading in Urban Areas: The Paris Case Study. In *IFIP Wireless Days conference*.
- Oyman, O. & Singh, S. (2012). Quality of Experience for HTTP Adaptive Streaming Services. *IEEE Communications Magazine*, 50(4):20–27.
- Pang, Z.; Sun, L.; Wang, Z.; Xie, Y. & Yang, S. (2017). Understanding Performance of Edge Prefetching. In *MultiMedia Modeling*, pp. 527--539, Cham. Springer International Publishing.
- Peterson, L. L. & Davie, B. S. (2007). *Computer Networks: A Systems Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 4th edição. ISBN 0123705487.
- Pyattaev, A.; Hosek, J.; Johnsson, K.; Krkos, R.; Gerasimenko, M.; Masek, P.; Ometov, A.; Andreev, S.; Sedy, J.; Novotny, V. & Koucheryavy, Y. (2015). 3GPP LTE-Assisted Wi-Fi-Direct: Trial Implementation of Live D2D Technology. *ETRI Journal*, 37(5).
- Pyattaev, A.; Johnsson, K.; Andreev, S. & Koucheryavy, Y. (2013a). 3GPP LTE Traffic Offloading onto WiFi Direct. In *2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 135–140.
- Pyattaev, A.; Johnsson, K.; Andreev, S. & Koucheryavy, Y. (2013b). Proximity-Based Data Offloading via Network Assisted Device-to-Device Communications. In *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*, pp. 1–5. ISSN 1550-2252.
- Pyattaev, A.; Johnsson, K.; Surak, A.; Florea, R.; Andreev, S. & Koucheryavy, Y. (2014). Network-assisted D2D Communications: Implementing a Technology Prototype for Cellular Traffic Offloading. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 3266–3271. ISSN 1525-3511.

- Rebecchi, F.; de Amorim, M. D.; Conan, V.; Passarella, A.; Bruno, R. & Conti, M. (2015). Data Offloading Techniques in Cellular Networks: A Survey. *IEEE Communications Surveys Tutorials*, 17(2):580–603. ISSN 1553-877X.
- Reis, D. M.; Lins, T. S.; Nogueira, J. M. S. & Mota, V. F. S. (2018). ShareFile: Sharing Content Through Device-to-Device Communication. In *2018 IEEE Symposium on Computers and Communications, ISCC 2018, Natal, Brazil, June 25-28, 2018*, pp. 736–741. IEEE.
- Schwarz, S. & Rupp, M. (2016). Society in Motion: Challenges for LTE and Beyond Mobile Communications. *IEEE Communications Magazine*, 54(5):76–83.
- Scott, J.; Gass, R.; Crowcroft, J.; Hui, P.; Diot, C. & Chaintreau, A. (2009). CRAWDAD dataset cambridge/haggle (v. 2009-05-29). Downloaded from <https://crawdad.org/cambridge/haggle/20090529>.
- Shah, S. T.; Hasan, S. F.; Seet, B.-C.; Chong, P. H. J. & Chung, M. Y. (2018). Device-To-Device Communications: A Contemporary Survey. *Wireless Personal Communications*, 98(1):1247–1284. ISSN 1572-834X.
- Shan, G. & Zhu, Q. (2019). Sociality and Mobility-Based Caching Strategy for Device-to-Device Communications Underlying Heterogeneous Networks. *IEEE Access*, 7:53777–53791. ISSN .
- Silva, V.; Mota, V.; Macedo, D. & Dias De Amorim, M. (2021). Joint Content-Mobility Priority Modeling for Cached Content Selection in D2D Networks. *Journal of Network and Systems Management*, 29(1):1–37.
- Sodagar, I. (2011). The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia*, 18(4):62–67.
- Song, J. & Choi, W. (2019). Mobility-Aware Content Placement for Device-to-Device Caching Systems. *IEEE Transactions on Wireless Communications*, 18(7):3658–3668. ISSN .
- Stockhammer, T. (2011). Dynamic Adaptive Streaming over HTTP: Standards and Design Principles. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, pp. 133–144. ACM.
- Tanenbaum, A. S. & Wetherall, D. J. (2010). *Computer Networks*. Prentice Hall Press, Upper Saddle River, NJ, USA, 5th edição. ISBN 0132126958, 9780132126953.

- Vaca-Rubio, C.; Gómez, G.; Aguayo-Torres, M. & López Martínez, F. (2019). Statistical Characterization of the Chunk Size Distribution in DASH. In *XXXIV Simposio Nacional de la Unión Científica Internacional de Radio (URSI)*.
- Vasilakos, X.; Siris, V. A. & Polyzos, G. C. (2016). Addressing Niche Demand Based on Joint Mobility Prediction and Content Popularity Caching. *Computer Networks*, 110:306 – 323. ISSN 1389-1286.
- Wang, M.; Xu, C.; Chen, X.; Hao, H.; Zhong, L. & Yu, S. (2019). Differential Privacy Oriented Distributed Online Learning for Mobile Social Video Prefetching. *IEEE Transactions on Multimedia*, 21(3):636–651.
- Wang, S.; Zhang, X.; Zhang, Y.; Wang, L.; Yang, J. & Wang, W. (2017). A Survey on Mobile Edge Networks: Convergence of Computing, Caching and Communications. *IEEE Access*, 5:6757–6779. ISSN .
- Wu, D.; Liu, B.; Yang, Q. & Wang, R. (2020). Social-Aware Cooperative Caching Mechanism in Mobile Social Networks. *Journal of Network and Computer Applications*, 149:102457. ISSN 1084-8045.
- Ye, Z.; De Pellegrini, F.; El-Azouzi, R.; Maggi, L. & Jimenez, T. (2017). Quality-Aware DASH Video Caching Schemes at Mobile Edge. In *2017 29th International Teletraffic Congress (ITC 29)*, volume 1, pp. 205–213.
- Yuan, Z.; Zhuang, W.; Wei, X. & Zhou, L. (2019). Joint Social-Aware and Mobility-Aware Caching in Cooperative D2D. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, pp. 656–661. ISSN .
- Zhang, W.; Wu, D.; Yang, W. & Cai, Y. (2019). Caching on the Move: A User Interest-Driven Caching Strategy for D2D Content Sharing. *IEEE Transactions on Vehicular Technology*, 68(3):2958–2971. ISSN .
- Zheng, K.; Zhang, X.; Zheng, Q.; Xiang, W. & Hanzo, L. (2015). Quality-of-Experience Assessment and its Application to Video Services in LTE Networks. *IEEE Wireless Communications*, 22(1):70–78.

Apêndice A

Lista de Siglas

- 3GPP** - *3rd Generation Partnership Project*
- CUE** - *Cellular UE*
- DASH** - *Dynamic Adaptive Streaming over HTTP*
- D2D** - *Device-to-Device*
- DTN** - *Delay and Disruption Tolerant Network*
- EWMA** - *Exponentially Weighted Moving Average*
- FIFO** - *First In First Out*
- HTTP** - *Hypertext Transfer Protocol*
- IEEE** - *Institute of Electrical and Electronics Engineers*
- ISM** - *Industrial, Scientific and Medical*
- ISO** - *International Organization for Standardization*
- IUE** - *Isolated UE*
- LFU** - *Least Frequently Used*
- LRS** - *Least Recently Stored*
- LRU** - *Least Recently Used*
- LTE** - *Long Term Evolution*
- LTE-A** - *Long Term Evolution Advanced*
- MEC** - *Mobile Edge Computing*
- MFQ** - *Most Frequent Quality*
- MIT** - *Massachusetts Institute of Technology*
- MP-CQ** - *Modelo de Prioridade Ciente de Qualidade*
- MP-PG** - *Modelo de Prioridade de Propósito Geral*
- MPEG** - *Moving Picture Experts Group*
- MR-CQ** - *Modelo de Rede Ciente de Qualidade*
- MR-PG** - *Modelo de Rede de Propósito Geral*

MRQ - *Most Recent Quality*
NAT - *Network Address Translation*
ONE - *Opportunistic Network Environment*
ProSe - *Proximity Services*
PS-D2D - *Public Safety D2D*
PSN - *Pocket Switched Network*
QoE - *Quality of Experience*
RUE - *Relay UE*
SACC - *Social-Aware Cooperative Caching*
SC - *Servidor de Conteúdo*
SM - *Servidor de Monitoramento*
SVC - *Scalable Video Coding*
UE - *User Equipment*

Apêndice B

Tabela de Variáveis Empregadas nos Modelos de Rede e de Prioridade

Variável	Descrição
D	Conjunto de dispositivos contidos na rede
d_i	i -ésimo dispositivo contido em D
C_{\max}	Capacidade do <i>cache</i> de um dispositivo em bytes
A	Conjunto de arquivos oferecidos pelo Servidor de Conteúdo
a_j	j -ésimo arquivo contido em A
s_j	Tamanho do arquivo a_j em bytes
Q	Conjunto de qualidades disponíveis para cada arquivo em A
q_k	k -ésima ID de qualidade contida em Q
ζ_{a_j, d_i}	Prioridade para o arquivo a_j no <i>cache</i> de d_i
PE	Potencial de Envio
PD	Potencial de Disseminação
λ_{d_i}	Máximo de bytes que d_i pode transmitir via D2D
p_{a_j, d_i}	Popularidade do arquivo a_j em d_i
$\overline{p_{a_j, \text{global}}}$	Popularidade global do arquivo a_j
$p_{\text{last, global, } a_j}$	Popularidade global do arquivo a_j no últ. interv. de monit.
$\overline{p_{a_j, d_i}}$	Popularidade local do arquivo a_j em d_i
$p_{\text{last, } a_j, d_i}$	Popularidade local do arquivo a_j em d_i no últ. interv. de monit.
$\overline{t_{d_i}}$	Duração do contato de d_j com seus vizinhos
ε	Vazão alcançável dos dispositivos por meio de comunicação D2D

Variável	Descrição
t_{last,d_i}	Duração do último contato de d_i com um de seus vizinhos
v_l	l -ésimo vídeo contido no <i>cache</i> de um dispositivo
η_{v_l,d_i}	Prioridade para o vídeo v_l no <i>cache</i> de d_i
PDQ	Potencial de Disseminação com Qualidade
PC	Potencial de Consumo
p_{v_l,d_i}	Popularidade do vídeo v_l em d_i
SQ_{v_l,d_i}	<i>Score</i> de Qualidade do vídeo v_l em d_i
$\overline{c_{v_l,d_i}}$	Número de vezes que d_i consumiu o vídeo v_l
c_{last,v_l,d_i}	Número de vezes que o vídeo v_l foi consumido por d_i no último intervalo de monitoramento
t_{v_l}	Duração do vídeo v_l em sua versão completa
θ_{v_l,q_k,d_i}	Porcentagem de <i>chunks</i> do vídeo v_l na qualidade q_k em d_i (em relação à duração total do vídeo)
NC_{v_l,q_k,d_i}	Quantidade de <i>chunks</i> do vídeo v_l , na qualidade q_k , em d_i
$SV_{v_l,d_i,\kappa}$	<i>Score</i> do vizinho κ de d_i , que possui o vídeo v_l desejado
κ	Vizinho de um dispositivo contido em D
$t_{left,d_i,\kappa}$	Duração restante estimada do contato entre d_i e κ
$t_{curr,d_i,\kappa}$	Duração do contato atual entre d_i e κ
$\overline{t_{d_i,\kappa}}$	Duração do contato entre d_i e κ
$t_{last,d_i,\kappa}$	Duração do contato entre d_i e κ no último intervalo de monitoramento
$\alpha,\beta,\gamma,\delta,\epsilon$	Pesos das EWMA's