Gerenciamento do Cache em Gateways IoT com atualização oportunista de Dados e Seleção Dinâmica de Grupos

Edvar Afonso L. Filho[1] **e Miguel Elias M. Campista**[1] *

[1]Universidade Federal do Rio de Janeiro
GTA/PEE-COPPE/DEL-Poli

{edvar,miguel}@gta.ufrj.br

Resumo. *O mercado de desenvolvimento de aplicações no contexto IoT estão em um momento de enorme efervescência. Com isso, o controle do tráfego de comunicação entre os dispositivos IoT é uma solução para o gerenciamento do consumo de energia desses dispositivos. Neste trabalho é proposto um mecanismo para coleta inteligente de dados em dispositivos IoT com múltiplos sensores. O trabalho propõe o uso do armazenamento de dados em* cache*, combinado com coleta oportunista de dados com uso de Agentes Móveis. A seleção dinâmica de grupos é usada como base para construção dos itinerários e a proposta é avaliada num cenário com fusão de dados. As simulações demonstram que o mecanismo proposto reduz o número de envios do agente em até 37% e uma redução no consumo de energia correspondente em até 25% em comparação ao uso tradicional de Agentes Móveis.*

1. Introdução

Os desafios relacionados ao paradigma da Internet das Coisas (*Internet of Things* – IoT) vêm impulsionando pesquisas que cobrem desde a análise e filtragem de grandes massas de dados até a comunicação eficiente dos dispositivos em rede. Tais redes, formadas entre dispositivos IoT, são compostas por elementos heterogêneos que tipicamente possuem restrições de banda passante e de energia. Em termos de processamento, há atualmente um entendimento que os dispositivos IoT podem ser mais "inteligentes" que os sensores tradicionais, aqueles usados em redes para sensoriamento especializado, como definido há mais de uma década [Liu e Baiocchi, 2016].

A maioria dos dispositivos IoT conta com interfaces de comunicação sem fio. Caso estas sejam usadas, elas se tornam as principais responsáveis pelo consumo de energia, uma vez que a recepção e a transmissão de dados são tarefas energeticamente custosas nesse cenário. Na arquitetura de comunicação em múltiplos saltos, a preocupação com a energia é ainda maior, já que o tráfego gerado por qualquer um dos dispositivos pode exigir encaminhamento até o destino final remoto, muitas vezes o *gateway* da rede. Como o encaminhamento nesse caso é inevitável para a manutenção da conectividade da rede, alternativas como a introdução de inteligência no *gateway* ou agregação de dados vêm sendo investigadas. A inclusão de inteligência no *gateway* visa tornar a coleta de dados mais eficiente com o suporte da computação em

*Trabalho realizado com apoio do CNPq; da FAPERJ; da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES), Código de Financiamento 001; e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processos n^o 15/24494-8 e 15/24490-2.

névoa [Bonomi et al., 2012, Rahmani et al., 2015] ou com o uso de técnicas de *cache* otimizadas [Mišić e Mišić, 2018]. Já as técnicas de agregação de dados [Xu et al., 2012] buscam aumentar a eficiência do protocolo de comunicação através da redução da quantidade de dados enviados em direção ao *gateway*. Nesse sentido, o uso de Agentes Móveis (AMs) surge como uma possibilidade eficiente para a coleta de dados em redes IoT.

Os AMs são códigos de computador enviados na rede com tarefas previamente programadas, que devem ser executadas em dispositivos IoTs pré-determinados. Quando as tarefas correspondem a uma coleta de dados, os dispositivos visitados são chamados de *nós fontes*. As tarefas a serem executadas pelos AMs podem ser, por exemplo, o pré-processamento dos dados em cada dispositivo (usando técnicas de compressão de dados, fusão ou agregação) para redução da quantidade de dados produzidos e, consequentemente, encaminhados pela rede. Os AMs são despachados na rede de múltiplos saltos através de um itinerário que pode ou não ser pré-determinado pelo *gateway*. Na coleta tradicional, os agentes percorrem comumente nós intermediários para seguir o itinerário programado entre dois nós fontes consecutivos. As propostas tradicionais, no entanto, ignoram a coleta de dados nos dispositivos intermediários, usados apenas para encaminhamento. Estes dispositivos, porém, podem conter dados atualizados e relevantes aos interesses dos usuários da rede, que poderiam ser coletados de forma proativa. Apesar do conceito de AM ser da década de 90, o seu emprego vem sendo rediscutido atualmente como uma forma de maximizar a quantidade de dados coletados em redes IoT [Gavalas et al., 2017, Lu et al., 2019].

Este trabalho propõe o uso de AMs para coleta de dados em redes IoT orquestradas pela presença de um *gateway* "inteligente". O *gateway* é responsável pela definição do tipo de dado a ser coletado e, consequentemente, dos nós fontes e do itinerário percorrido pelo AM. Além do tipo de dado e da sequência de nós fontes, o AM também é programado para coletar oportunisticamente dados de dispositivos intermediários, ou seja, dados provenientes de nós usados para interconectar nós fontes consecutivos. Esse tipo de coleta é chamada oportunista, uma vez que o conteúdo é coletado pró-ativamente, mesmo que não haja uma solicitação explícita por parte dos usuários externos à rede IoT. A ideia é aproveitar o itinerário para coletar dados com potencial de serem solicitados em breve, e assim, economizar energia com o despacho de um novo agente.

O tipo de dado a ser coletado oportunisticamente e as respectivas fontes dependem da política de atualização de cache do *gateway*, que leva ainda em conta a possibilidade da fusão de dados do mesmo tipo e falhas de transmissão. A fusão é importante para conter o aumento do tamanho do AM, algo que não seria mais complicado caso uma política de concatenação de dados fosse empregada. Já a introdução de falhas é importante para tornar o cenário mais realista, uma vez que redes IoT geralmente são compostas por dispositivos sem fio cuja comunicação pode estar sujeita à interferência e perdas por obstáculo. Este trabalho modela o problema de seleção dos nós fontes como um problema de cobertura do tipo *Weigthed Set Cover* e o de seleção dos tipos de conteúdo oportunistas, como um problema *knapsack*. Uma versão preliminar da coleta oportunista com AM foi publicada anteriormente pelos autores [Filho e Campista, 2019]. Na versão preliminar, porém, um cenário bem mais simplificado foi considerado, sem fusão de dados e sem cenários com perdas. Além disso, critérios diferentes para a coleta dos dados provenientes dos nós fontes e dos nós oportunistas foram adotados. A proposta atual considera adicionalmente

a presença de dispositivos mais complexos, equipados com sensores de tipos diferentes. Os resultados obtidos via simulação mostram que com o mecanismo proposto é possível obter redução de até 37% no número de envios do AM para realizar a coleta de dados na rede, além de ser possível uma redução entre 14% e 25% no consumo médio de energia nos dispositivos da rede em comparação com a coleta tradicional com AMs.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve o problema atacado e introduz a arquitetura da rede considerada neste trabalho. O mecanismo de coleta proposto é apresentado na Seção 4. A Seção 5 descreve o ambiente de simulação, enquanto a Seção 6 mostra os resultados obtidos. Finalmente, a Seção 7 conclui este trabalho e lista os trabalhos futuros.

2. Trabalhos Relacionados

A proposta deste trabalho está relacionada a duas áreas de pesquisa em redes IoT: seleção de nós fontes e uso dos AMs. Parte dos nós da rede IoT são selecionados para realização de tarefas específicas, tais como a coleta de dados pelos dispositivos. Já o uso de AMs está relacionado com a coleta de dados desde as redes de sensores tradicionais e os respectivos mecanismos desenvolvidos para a redução do consumo de energia da rede. Por exemplo, considera-se como forma de economia de energia a determinação do itinerário e a fusão dos dados nos próprios dispositivos da rede. Dessa forma, esta seção descreve trabalhos que pertençam a uma das duas áreas de pesquisa destacadas.

A seleção de grupos de nós fonte para realização de tarefas tem como objetivo principal a redução do tráfego de informações redundantes coletadas pelos dispositivos. A ideia central é usar a sobreposição de cobertura dos dados sensoriados pelos dispositivos distribuídos em uma mesma região para selecionar com mais critério quais dispositivos devem estar ativos num determinado intervalo de tempo. Assim, evita-se que informações redundantes sejam enviadas na rede e, consequentemente, permite que haja uma redução do consumo de energia dos dispositivos sem fio. Algumas propostas foram desenvolvidas buscando selecionar de forma inteligente qual grupo de dispositivos é mais adequado para realizar as tarefas desejadas. A maioria dos trabalhos consideram parâmetros individuais dos dispositivos para realizar a seleção, tais como localização, alcance de sensoriamento (*sensing range*), custo, energia remanescente do nó, confiabilidade e quantidade de falhas. Além de parâmetros individuais, há propostas mais recentes [Azzam et al., 2016, Alagha et al., 2019] que consideram parâmetros de grupo para a seleção do conjunto de dispositivos mais adequados.

Azzam et al. tratam da seleção de dispositivos ativos aplicada a *Mobile Crowd Sensing* [Azzam et al., 2016]. Os algoritmos de seleção avaliam o uso de incentivos aos participantes da rede de modo a compensar o gasto energético com as tarefas atribuídas. Os autores defendem que a utilização de parâmetros de grupo para a seleção é mais eficiente e permite uma Qualidade de Informação melhor que o uso de parâmetros individuais. Alagha et al. propõem o uso de dois tipos de parâmetros na seleção dos nós [Alagha et al., 2019]. O objetivo é aumentar a eficiência do processo de localização de nós fontes de radiação com a utilização de sensores distribuídos em uma área de interesse. O processo de localização é realizado em várias rodadas, nas quais as informações dos sensores são enviadas para um elemento central que avalia a qualidade dos dados recebidos e parâmetros individuais dos sensores. Dessa forma, o nó central se torna capaz

de selecionar o grupo de dispositivos mais adequado para a rodada seguinte de coleta. Na inicialização do sistema, os primeiros dispositivos são selecionados com o uso de parâmetros de grupo: cobertura, utilização, confiabilidade, custo e energia residual do grupo. Um algoritmo genético é empregado para a seleção do grupo inicial. O trabalho aqui apresentado utiliza parâmetro individual para seleção do grupo de nós. Esse tipo de seleção é escolhido porque permite o uso de algoritmos menos complexos. Como a seleção de grupos de nós fontes proposta é feita a cada rodada de coleta, é importante que a seleção desses grupos de dispositivos seja um processo rápido e pouco custoso computacionalmente.

A utilização de AMs para coleta de dados é um assunto popular em redes de sensores sem fio e que recentemente vem sendo revisitado em pesquisas relacionadas a redes IoT e veiculares [Lu et al., 2019, Urra e Ilarri, 2019]. A maioria dos trabalhos focam em estratégias para determinar itinerários otimizados para encaminhamento dos AMs, proporcionando eficiência energética. Gavalas et al. propõem um mecanismo de cálculo de itinerários para múltiplos AMs, considerando principalmente a eficiência no consumo de energia influenciada pelo aumento do tamanho dos AMs [Gavalas et al., 2017]. Os AMs aumentam de tamanho à medida que avançam no itinerário e agregam informações dos nós fontes visitados. Urra e Ilarri apresentam uma abordagem inovadora com o uso de AMs em redes veiculares [Urra e Ilarri, 2019]. Nela, os veículos são usados como elementos nos quais os AMs realizam a filtragem, fusão e coleta de dados, além de servir como nós intermediários para o encaminhamento dos AMs. O trabalho também apresenta um mecanismo oportunista para aplicação do AM, relacionado com o encaminhamento dos agentes. Esse oportunismo refere-se ao encaminhamento do AM, que só pode ser feito quando o veículo estiver conectado em uma área de cobertura de comunicação. Diferentemente da proposta apresentada neste artigo, ambos trabalhos não consideram que dados relevantes também possam ser coletados oportunisticamente nos dispositivos intermediários, enquanto o AM é encaminhado ao longo do seu itinerário entre nós fonte consecutivos. Até onde se sabe não existem outros trabalhos na literatura que considerem o mecanismo de coleta oportunista de dados ou a relevância dos conteúdos para aumento da eficiência da coleta de dados com AMs.

3. Descrição do Problema e Modelo da Rede

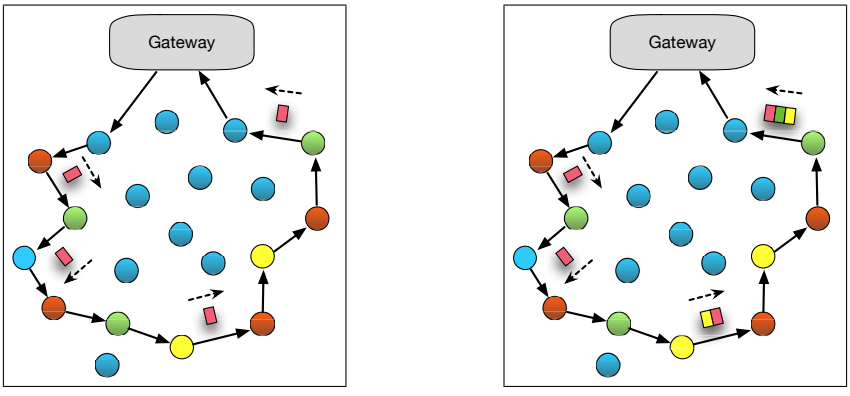
Esta seção descreve o problema atacado e o modelo de rede considerado.

3.1. Descrição do problema

A aplicação de AMs em redes de sensores, ou mais recentemente, em redes IoT pode ocorrer com o uso de Planejamento Estático de Itinerários ou Planejamento Dinâmico de Itinerários. O Planejamento Estático existe quando o planejamento do itinerário é todo feito pelo *gateway* e registrado no código do AM antes que este seja despachado pela rede. O Planejamento Estático possui a vantagem de transferir o processamento mais complexo para o elemento que possui o maior poder de processamento da rede. Por outro lado, exige que o *gateway* periodicamente se atualize sobre o estado dos dispositivos e da topologia para calcular os itinerários utilizando enlaces ainda ativos. Já no Planejamento Dinâmico, o AM tem autonomia para encontrar os melhores caminhos para percorrer a rede entre os nós fontes definidos. Essa abordagem exige um gasto maior de energia com processamento nos dispositivos IoT, restritos em energia, mas elimina a

necessidade de atualizações periódicas do *gateway*. Tanto no caso Estático quanto no Dinâmico, porém, a grande maioria das propostas é focada na construção de itinerários com o objetivo de reduzir o consumo de energia da rede. Nas propostas mais recentes, os itinerários são calculados considerando o crescimento do AM à medida que este coleta dados dos nós fontes. Este trabalho considera apenas o Planejamento Estático de Itinerários por questões de simplicidade.

A Figura 1(a) ilustra uma aplicação típica de coleta com Planejamento Estático de Itinerários. A figura exemplifica uma configuração tradicional, com o uso de um AM em uma rede IoT. Nesse exemplo, o agente é programado pelo *gateway* para realizar a coleta em quatro nós fontes (em vermelho na figura). Como pode haver encaminhamento por múltiplos saltos entre dois nós fontes consecutivos, surge uma oportunidade de coleta também nos nós intermediários (em azul, amarelo e verde na figura). Essa é a ideia capturada pela proposta deste trabalho, que aproveita os itinerários programados para coleta de dados também dos nós intermediários. A Figura 1 apresenta um exemplo da coleta proposta. Neste exemplo, uma fusão com taxa de compressão 1 é utilizada e, por isso, o AM não aumenta de tamanho ao coletar conteúdos do mesmo tipo. Considerando que a fusão não é feita com tipos diferentes de conteúdo, o AM aumenta seu tamanho apenas se conteúdos distintos forem concatenados em seu *payload*. Na Figura, além dos dados provenientes dos nós fontes em vermelho, considera-se também os conteúdos existentes em um subconjunto de nós intermediários, indicados pelas cores amarelo e verde. Os nós intermediários, provedores dos dados não solicitados, são selecionados baseado em um problema de otimização para o aumento da eficiência da coleta. O objetivo com a coleta oportunista é reduzir o número de vezes que um agente móvel precisa ser despachado na rede, uma vez que os dados coletados são atualizados com maior frequência. O mecanismo proposto é descrito a seguir em maiores detalhes.



(a) (b)

Figura 1. (a) Coleta tradicional. (b) Coleta oportunista proposta.

3.2. Modelo da rede

A rede considerada neste trabalho é composta por um *gateway* e dispositivos IoT fixos que monitoram uma Área de Interesse. O gerenciamento da coleta de dados é realizado exclusivamente pelo *gateway*. Portanto, este é o elemento com a função de coordenar a recepção de requisições e envio das respostas para os clientes externos à rede. Além disso, o *gateway* gerencia a seleção dinâmica do grupo de dispositivos mais adequado

para a coleta dos dados (nós fontes) e coordena a atualização das informações em *cache*. Também é função do *gateway*, a cada rodada de coleta, realizar o cálculo do itinerário que deve ser percorrido pelo AM.

No modelo considerado, cada rodada de coleta é iniciada após o recebimento de uma requisição pelo *gateway*. Cada requisição contém a descrição de uma tarefa de coleta que deve ser executada na respectiva rodada. As requisições são relacionadas à coleta de um tipo de conteúdo principal. As descrições das tarefas devem conter, no mínimo, as seguintes informações: tipo de conteúdo principal a ser coletado (p.ex., temperatura, nível de ruído, umidade) – chamado, a partir de agora, de conteúdo principal; tipo de agregação desejada na coleta (p.ex., valor médio, mínimo ou máximo); e Área de Interesse do dado. Alguns exemplos de tarefas recebidas são: coletar a temperatura média na área de interesse; coletar o nível máximo de poluição referente a um determinado gás em determinada sub-região da área de interesse; e assim por diante. As respostas enviadas pelo *gateway* para os clientes são geradas a partir da agregação de dados coletados nos dispositivos. Essas respostas podem ser uma agregação dos dados recém coletados da rede com dados previamente armazenados em *cache*, desde que estes últimos ainda sejam válidos. Considera-se que os dispositivos da rede IoT enviam os dados para o *gateway* com o tempo de validade associado. Dessa forma, o *gateway* é capaz de analisar se um dado em *cache* ainda é válido e se pode ser enviado como resposta às requisições.

No modelo proposto, a rede é representada por um conjunto de dispositivos, $\mathcal{N}$. Cada dispositivo $D_n \in \mathcal{N}$, possui um conjunto de sensores capaz de coletar dados de conteúdos distintos, k_i, que são representados pelo conjunto $\mathcal{K}_\mathcal{S} = \{k_1, k_2, ..., k_m\}$. Os dispositivos possuem informação sobre sua própria coordenada geográfica e alcance de cobertura dos sensores e enviam essas informações ao *gateway* durante o processo de registro na rede, na inicialização. Com a informação de coordenada e alcance de cobertura de cada tipo de sensor, o *gateway* é capaz de mapear a cobertura de toda área de interesse, identificando quais sensores cobrem qual parte da área. Assim, é possível realizar esse mapeamento para cada conteúdo distinto disponível para coleta. Retornando até a Figura 1, assume-se que os nós em vermelho, selecionados pelo *gateway*, são capazes de cobrir toda a área de interesse para a coleta do conteúdo principal.

O controle das informações armazenadas em *cache* pelo *gateway* é feito através do uso de tabelas que registram os dispositivos e os tipos de conteúdos associados a cada um deles. Através dos registros, por dispositivo, dos instantes de tempo das atualizações de conteúdos, o *gateway* é capaz de ter um mapeamento granularizado dos dados que ainda não estão expirados. Ao associar essas informações com as informações de coordenadas e cobertura enviadas pelos dispositivos na inicialização, o *gateway* torna-se capaz de mapear detalhadamente quais regiões da Área de Interesse têm informações atualizadas em *cache*.

4. Mecanismo de coleta proposto

A proposta desenvolvida neste trabalho é uma evolução do Agente-knap [Filho e Campista, 2019], trabalho anterior, no qual um estudo inicial sobre a coleta oportunista de dados com o uso de AMs em redes IoT com armazenamento de dados em *cache* no *gateway* foi apresentado. O Agente-knap, porém, não considerava a fusão dos dados ao longo do itinerário, o que por si só, já altera substancialmente a coleta de

dados. Enquanto o Agente-Knap considerava a mera concatenação dos dados coletados, o que aumentava o tamanho do AM; a proposta atual considera a fusão dos dados relativos a um mesmo conteúdo, o que tende a conter o aumento do tamanho do AM. Dessa forma, critérios diferentes para a coleta dos dados provenientes dos nós fontes e dos nós oportunistas são adotados. Além da fusão e das mudanças nos algoritmos de seleção, a arquitetura do sistema foi ampliada e adaptada para redes de dispositivos IoT mais complexos, equipados com um ou múltiplos sensores. A seguir, as principais características da coleta proposta são apresentadas.

4.1. Seleção dos nós fontes

A seleção dos nós fontes é utilizada para determinar, a cada rodada, o grupo de dispositivos visitados para a coleta do conteúdo principal. Esse grupo de nós fontes é selecionado tendo em vista a maior cobertura possível da área de interesse, de preferência total, referente ao conteúdo solicitado na requisição recebida.

O mecanismo de seleção dos nós fontes é modelado como um problema clássico de cobertura, o *Weigthed Set Cover*. Nesse problema, a área de interesse é representada por pontos equidistantes e cada dispositivo D_n é associado a um subconjunto desses pontos. Esse subconjunto é formado pelos pontos cobertos pelos sensores do dispositivo que coletam o conteúdo principal. Para cada dispositivo também é associado um peso, W_{D_n}, para diferenciar a prioridade de cada dispositivo no processo de seleção. A solução do problema é, portanto, encontrar o conjunto de nós fontes, $\mathcal{N}_f \subset \mathcal{N}$, capaz de cobrir a maior parte da área de interesse e que tenha o menor peso total associado – somatório dos pesos dos dispositivos em $\mathcal{N}_f$. Neste trabalho, o peso W_{D_n} associado a cada dispositivo é baseado no instante de tempo registrado na última atualização do conteúdo de interesse no *cache* do *gateway*. Por exemplo, se a rodada da vez estiver associada a uma coleta do conteúdo temperatura, o peso usado para cada dispositivo seria inversamente proporcional ao tempo decorrido desde o instante de tempo registrado na última atualização da informação de temperatura, daquele dispositivo, no *cache* do *gateway*. Assim, dispositivos com informação mais antiga registrada em *cache* têm menor peso associado e, consequentemente, maior probabilidade de serem selecionados na nova rodada de coleta. Partindo desse raciocínio, o *gateway* executaria as seguintes etapas:

1. Verifica os dados de temperatura em *cache* e mapeia a região da área de interesse com informações de temperatura atualizadas;
2. Caso os dados de temperatura em *cache* não contemplem toda área de interesse, executa o algoritmo *Weigthed Set Cover* para a sub-região necessária para completar os dados da área de interesse (a sub-região pode coincidir com toda região caso não haja informações de temperatura atualizadas).

A solução do *Weigthed Set Cover* é o conjunto de nós fontes que, na próxima rodada, devem ser visitados para complementar as informações de temperatura no *cache*. Após a definição do grupo de nós fontes do conteúdo principal, a próxima etapa é a definição do itinerário I do AM que tem origem e destino no *gateway*. Nessa etapa, os algoritmos de *Dijkstra* e *Christofides* são usados para cálculo do itinerário. O objetivo é determinar a ordem de visita dos nós fontes pelo AM, podendo haver, nós intermediários para encaminhamento do AM entre dois nós fontes consecutivos. O algoritmo de *Christofides* é uma das heurísticas utilizadas para a solução do Problema do Caixeiro-Viajante

(PCV), que é o tipo de problema a ser resolvido para cálculo do itinerário do AM. A saída do algoritmo é a sequência de nós que devem ser visitados e a solução é determinada após os cálculos dos caminhos entre pares de nós e os respectivos custos determinados pelo algoritmo de *Dijkstra*. O algoritmo de *Christofides* é uma solução para o PCV que, no pior caso, garante uma razão de menos de 3/2 entre a solução encontrada e a solução ótima [Christofides, 1976].

4.2. Seleção de nós para coleta oportunista

O itinerário I definido é uma lista com a sequência de nós que precisam ser visitados na próxima rodada de coleta. A etapa seguinte é definir quais tipos de conteúdos disponíveis em I serão selecionados para a coleta oportunista. A solução para essa seleção pode ser modelada como um problema *0-1 knapsack*. Como cada tipo de conteúdo coletado oportunisticamente é concatenado ao *payload* do AM; cada k_n presente em I pode ser representado por um tamanho, em Bytes, e um valor. Esse valor está relacionado à quantidade de informação que os nós intermediários de I podem complementar aos dados de cada conteúdo k_n em *cache*. A restrição para solução do problema de otimização *knapsack* é o tamanho máximo do *payload* do AM, C, que deve ser informado como um parâmetro de entrada do sistema.

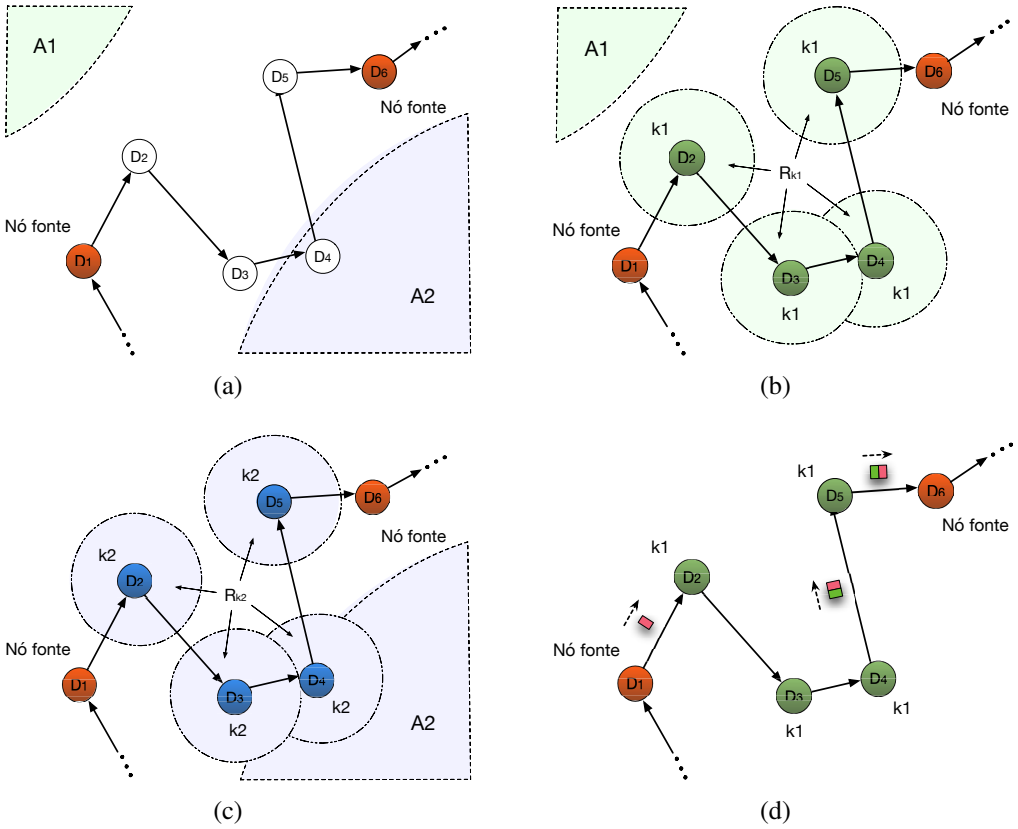


Figura 2. Mecanismo de seleção dos conteúdos nos nós intermediários.

A Figura 2 ilustra o processo de seleção de nós intermediários em todas etapas. Nela, o itinerário I e dois nós fontes (D_1 e D_6) estão representados, além de quatro nós intermediários (D_2, D_3, D_4 e D_5). Para simplificação, o *gateway* não está representado

na figura mas deve ser entendido que o AM é enviado do *gateway* diretamente para D_1 e de D_6 diretamente para o *gateway*, como visto na Figura 2(a). As áreas A1 e A2 representam as sub-regiões da área de interesse que, nessa rodada de coleta, correspondem aos dispositivos atualizados no *cache* respectivamente para os conteúdos k_1 e k_2. Nesse exemplo, a rede possui três conteúdos no total e todos nós intermediários são capazes de coletar dados dos conteúdos k_1 e k_2. O tamanho de C para coleta de dados oportunistas corresponde ao espaço para coleta de um tipo de conteúdo. O processo inicia com o *gateway* analisando cada uma das sub-regiões cobertas pelos nós intermediários – chamadas a partir daqui de R_{k_n}. Assim, nas Figuras 2(b) e 2(c) são representadas as sub-regiões R_{k_1} e R_{k_2}, referentes aos conteúdos k_1 e k_2. Para cada conteúdo, verifica-se o tamanho da interseção de R_{k_n} com as sub-regiões A1 ou A2 já com dados atualizados em *cache*. Quanto menor a interseção entre estas sub-regiões, mais prioritária torna-se a coleta do tipo de conteúdo analisado. Essa prioridade é representada pelo valor de cada conteúdo no problema *knapsack*[1]. Pelo exemplificado nas figuras, é possível identificar que R_{k_1} não tem nenhuma interseção com A1, porém R_{k_2} tem uma interseção considerável com A2. Dessa forma, o valor de k_1 neste problema *knapsack* é maior que o de k_2. Como C está limitado ao tamanho de um conteúdo, somente o conteúdo k_1 é coletado oportunisticamente pelo AM, além do conteúdo principal, como é indicado nas cores verde e vermelha na Figura 2(d).

Para controle das informações atualizadas no *cache* do *gateway*, o AM precisa registrar no seu código, durante o processo de coleta, de quais dispositivos foi executada a coleta com sucesso e de qual tipo de conteúdo. O detalhamento desse processo não está no escopo deste trabalho.

5. Ambiente de Simulação

Um ambiente de simulação utilizando a linguagem *Python* foi desenvolvido para avaliação da proposta deste trabalho. As topologias foram criadas com uso do pacote *NetworkX*. Para uma simplificação da análise, algumas premissas foram assumidas. A energia inicial de todos os dispositivos é a mesma; o tamanho dos dados de diferentes tipos de conteúdos é o mesmo; os dispositivos da rede possuem o mesmo número de sensores instalados, sendo um sensor para cada tipo de conteúdo; e o alcance de cada sensor nos dispositivos é o mesmo para toda rede. As requisições são recebidas pelo *gateway* e são enviadas com detalhamentos do tipo de conteúdo que deve ser considerado principal. A informação desejada sempre se refere a toda área de interesse.

Em todos os casos de simulação, os dispositivos são distribuídos aleatoriamente em uma área de tamanho fixo e formato quadrado. Os custos dos enlaces são diretamente proporcionais às distâncias Euclidianas entre os dispositivos. Cada dispositivo é capaz de se comunicar com os vizinhos que estão no raio de comunicação. O *gateway* da rede é fixo no ponto central da área de interesse. A Tabela 5 mostra detalhes dos outros parâmetros utilizados nas simulações.

[1]Mais precisamente, os valores de cada conteúdo no algoritmo *knapsack* são calculados com operações de conjuntos. Conforme descrito na Seção 4.1, toda área é dividida em pontos equidistantes. Cada dispositivo D_n é representado por conjuntos cujos elementos são os pontos que estão na área de cobertura dos sensores. A quantidade de pontos cobertos por toda R_{k_n} menos a quantidade de pontos da interseção com a sub-região atualizada em *cache* é o valor para o respectivo conteúdo no algoritmo *knapsack*.

Parâmetro	Valor
Quantidade de dispositivos na rede	100 e 180
Quantidade de conteúdos por dispositivo	4
Energia inicial por dispositivo	20 Joules
Energia de transmissão	200 η Joules/bit
Energia de recepção	150 η Joules/bit
Área de simulação	50 m x 50 m
Coordenada do *gateway*	(25, 25)
Raio de alcance sensores	20 m
Raio de comunicação dos dispositivos	20 m
Taxa de compressão de dados no processo de fusão	1.0
Tamanho inicial do AM	500 Bytes
Tamanho dos dados coletados	200 Bytes
Distribuição de chegada de requisições	Poisson, $\lambda = 5.0$

Tabela 1. Parâmetros das simulações.

As simulações foram realizadas para comparação do sistema de coleta proposto com a coleta do tipo tradicional. Na coleta tradicional, o AM não considera a coleta de conteúdos nos nós intermediários, conforme detalhado na Seção 3. Nos dois tipos de coleta, o *gateway* armazena os dados em *cache* para diferentes tempos de validade. As simulações foram divididas em dois tipos: Cenário sem perdas e Cenário com perdas. Para o cenário sem perdas foi avaliado principalmente o desempenho do sistema quando há variação do tamanho do *payload* do AM, C, e como esse desempenho varia para rede com diferentes quantidades de dispositivos. Foi avaliada a Energia Média Consumida na rede e a Quantidade de Envios do Agente Móvel.

Para o cenário com perdas, a rede foi fixada em uma quantidade de dispositivos e foi avaliada a Energia Média Consumida para diferentes níveis de perda na rede.

6. Resultados

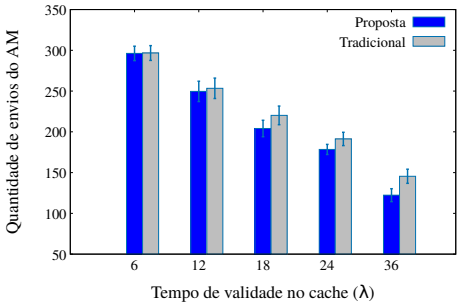
Os resultados foram obtidos com 10 rodadas de simulação para cada variação dos parâmetros. Uma rodada de simulação significa que o *gateway* executa as tarefas de coleta por um tempo de 400 vezes λ de Poisson. Nesse intervalo, o *gateway* recebe uma sequência de requisições segundo uma distribuição Poisson com $\lambda = 5$. Os tempos de validade do *cache* também são apresentados em referência ao tempo médio de chegadas de requisições, λ de Poisson. O intervalo de confiança para todos os resultados é de 95%.

6.1. Simulações em cenário sem perdas

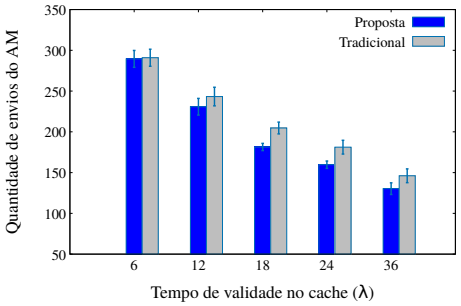
No cenário sem perdas, primeiramente foi avaliado o desempenho do sistema com relação à quantidade de envios do AM pelo *gateway* para diferentes tempos de validade dos conteúdos em *cache*. É esperado que, no sistema proposto, haja uma menor quantidade de envios pois a coleta oportunista permite que dados que serão solicitados no futuro possam estar disponíveis em *cache* de forma antecipada. Posteriormente, foi avaliado o comportamento do consumo médio de energia nos dispositivos para diferentes tempos de validade dos conteúdos em *cache*. Também foi avaliado como essas métricas se comportam com diferentes tamanhos de rede e espaço do *payload*, C.

As Figuras 3(a) e 3(b) apresentam os resultados para a rede com 100 e 180 dispositivos, respectivamente, com $C = 200$ Bytes, mostrando o impacto dos diferentes tempos

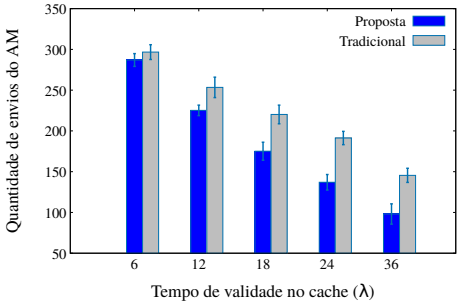
de validade de *cache*. Os resultados podem ser comparados também aos apresentados nas Figuras 3(c) e 3(d), para $C = 600$ Bytes. Observa-se que os ganhos da proposta se intensificam com um maior *payload* C pois mais dados são coletados oportunisticamente para o *cache*. Como consequência, menos envios do AM são necessários para atender as requisições. Um menor número de envios do AM proporciona também a redução do tráfego na rede e economia de recursos escassos dos dispositivos IoT. Os maiores ganhos da proposta observados foram para rede com 180 dispositivos, tempo de validade de *cache* de 24 e 36 λ e C igual a 600 Bytes, como visto na Figura 3(c). Particularmente para 24 λ, a média do número de envios com a coleta Tradicional foi de 181 e a com a coleta Proposta foi de 113, alcançando uma redução de aproximadamente 37% de envios.



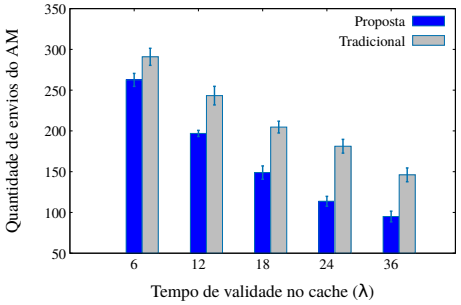
(a) Quantidade de envios do AM, rede com 100 dispositivos, C = 200 Bytes.



(b) Quantidade de envios do AM, rede com 180 dispositivos, C = 200 Bytes.



(c) Quantidade de envios do AM, rede com 100 dispositivos, C = 600 Bytes.

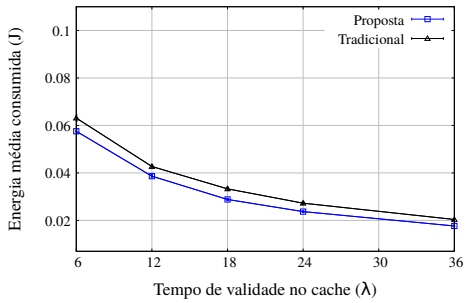


(d) Quantidade de envios do AM, rede com 180 dispositivos, C = 600 Bytes.

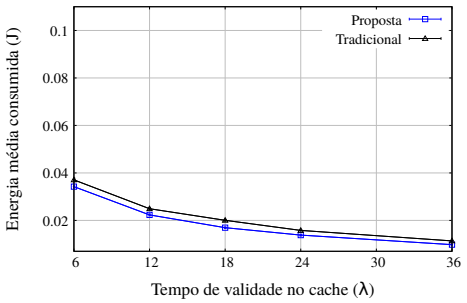
Figura 3. Cenário sem perda: Quantidade de envios do AM com variação do tempo de validade no *cache* e *payload*.

As Figuras 4(a) e 4(b) mostram os resultados de Energia Média Consumida para o cenário sem perdas para $C = 200$ Bytes. Já as Figuras 4(c) e 4(d) mostram os resultados para os casos com $C = 600$ Bytes. A energia média foi calculada considerando a energia consumida em todos dispositivos da rede após expirar o tempo de execução da tarefa de coleta de 400 λ. Observa-se que os ganhos da proposta também ficam maiores com o aumento do espaço do AM, conforme esperado. Os ganhos são relevantes no consumo de energia médio da rede, porém em proporções menores quando comparados aos ganhos do número de envios do AM. Por exemplo, no mesmo caso analisado anteriormente, rede com 180 dispositivos, 24 λ de tempo de validade e $C = 600$ Bytes, as Energias Médias Consumidas finais foram de 16 e 12 mili Joules, respectivamente, para a coleta Tradicional e Proposta. Esse resultado equivale a um ganho da proposta de 25%. Essa diferença

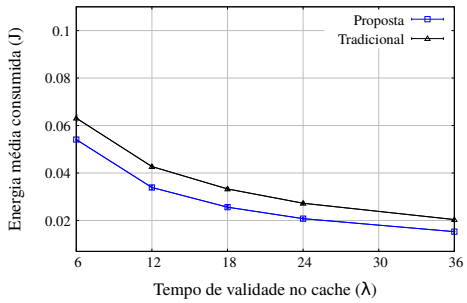
nos resultados está diretamente relacionada a um *tradeoff* intrínseco ao mecanismo de coleta proposto. O AM é enviado menos vezes, porém, como este aumenta o uso do espaço no *payload* para coletar dados oportunisticamente, há necessariamente um aumento do tamanho do agente que resultará no aumento do consumo médio de energia na rede. Para as simulações de Energia Média Consumida sem perdas os ganhos da proposta variaram entre 7,65% (180 dispositivos, C igual a 200 Bytes e tempo de validade de 6 λ) e 25 % (100 dispositivos, C igual a 600 Bytes e tempo de validade de 36 λ).



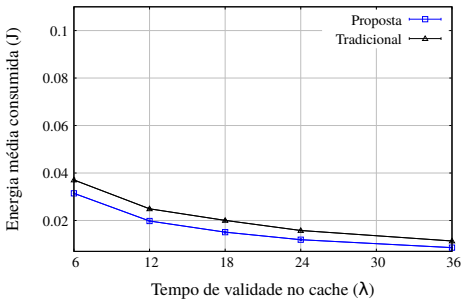
(a) Cenário sem perda, rede com 100 dispositivos, C = 200 Bytes.



(b) Cenário sem perda, rede com 180 dispositivos, C = 200 Bytes.



(c) Cenário sem perda, rede com 100 dispositivos, C = 600 Bytes.



(d) Cenário sem perda, rede com 180 dispositivos, C = 600 Bytes.

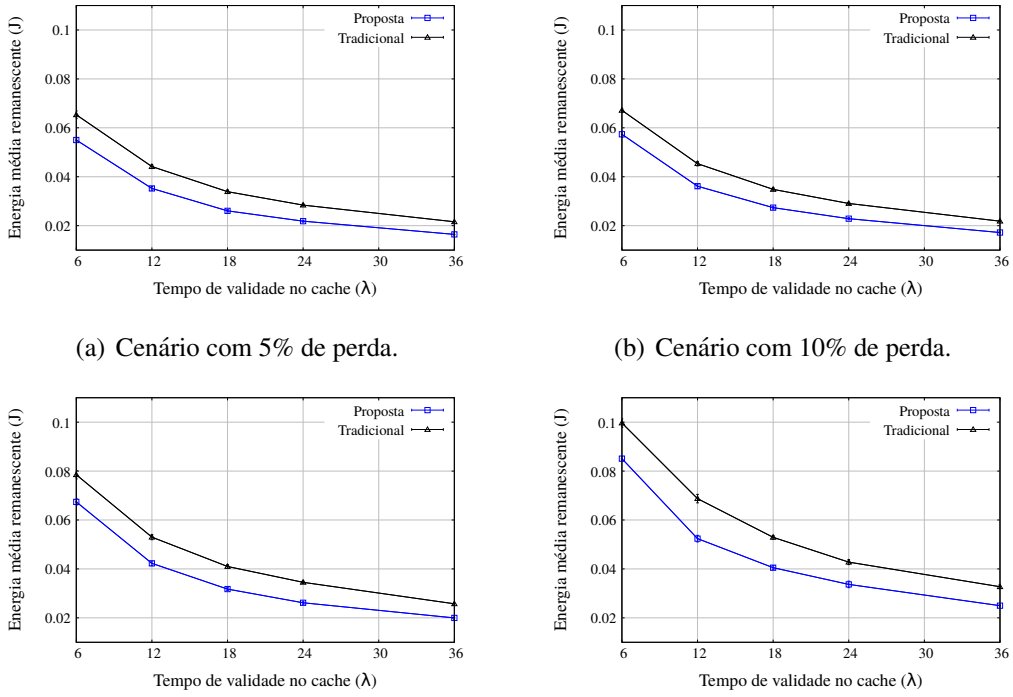
Figura 4. Cenário sem perda: Energia Média Consumida nos dispositivo.

6.2. Simulações em cenário com perdas

No cenário com perdas, foi variada a probabilidade de falha na transmissão do AM entre dispositivos vizinhos. Quando ocorre uma falha na transmissão, o AM é retransmitido a partir do último dispositivo do itinerário que fez a tentativa de transmissão. Para cada simulação, o mesmo valor de probabilidade de falha foi considerado em todos enlaces da rede. Para esse teste, não houve limite de retransmissões do AM para um mesmo dispositivo. As probabilidades de perda variaram entre os valores: 5%, 10%, 30% e 50%. Diferente das simulações apresentadas anteriormente, o número de dispositivos foi fixado em 100 e o tamanho de *payload* C foi fixado em 600 Bytes.

As Figuras 5(a), 5(b), 5(c) e 5(d) mostram, respectivamente, os cenários para 5, 10, 30 e 50% de perdas. Os resultados apresentam a Energia Média Consumida nos nós da rede após o término das rodadas de coleta. Comparando com o cenário sem perdas, é possível perceber a variação mais acentuada da Energia Média Consumida, principalmente quando o nível de perdas é alto, por exemplo, para 30 e 50%. Os ganhos para

as simulações sem perdas com 100 dispositivos variaram entre 14,15% (perda de 30 % e tempo de validade de 6 λ) e 24,21 % (perda de 30 % e tempo de validade de 24 λ). Apesar do gasto de energia ser maior em um cenário com perdas, o ganho da proposta não variou muito em relação aos resultados para a mesma configuração de simulação no cenário com perdas onde foram obtidos valores do ganho similares.



(a) Cenário com 5% de perda.

(b) Cenário com 10% de perda.

(c) Cenário com 30% de perda.

(d) Cenário com 50% de perda.

Figura 5. Cenário com perda: Energia Média Consumida nos dispositivos.

7. Conclusão e Trabalhos Futuros

Neste trabalho foram investigados os efeitos da seleção dinâmica de grupos de dispositivos para coleta de dados de cobertura em uma Área de Interesse. A coleta está associada à atualização de dados em *cache* com o uso de Agentes Móveis. O mecanismo proposto para coleta oportunista de conteúdos nos nós intermediários dos itinerários, que complementam as informações de cobertura existentes em *cache*, se mostrou eficiente proporcionando ganhos na redução do consumo de energia de até 25%. Também foi avaliado o desempenho do sistema em um cenário com perdas, onde foi possível observar que ganhos similares também podem ser obtidos. Como trabalhos futuros, pretende-se adaptar o mecanismo proposto para cenários com múltiplos Agentes Móveis despachados na rede simultaneamente para a tarefa de coleta. Além disso, planeja-se realizar a análise do desempenho com outras taxas de compressão no processo de fusão dos dados.

Referências

Alagha, A., Singh, S., Mizouni, R., Ouali, A. e Otrok, H. (2019). Data-driven dynamic active node selection for event localization in IoT applications-a case study of radiation localization. *IEEE Access*, 7:16168–16183.

Azzam, R., Mizouni, R., Otrok, H., Ouali, A. e Singh, S. (2016). Grs: A group-based recruitment system for mobile crowd sensing. *Journal of Network and Computer Applications*, 72:38–50.

Bonomi, F., Milito, R., Zhu, J. e Addepalli, S. (2012). Fog computing and its role in the internet of things. Em *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, p. 13–16. ACM.

Christofides, N. (1976). Worst-case analysis of a new heuristic for the travelling salesman problem. Relatório técnico, Carnegie-Mellon University, Pittsburgh.

Filho, E. A. L. e Campista, M. E. M. (2019). Coleta oportunista de dados em redes de internet das coisas através do uso de otimização discreta. Em *WPerformance - Workshop em Desempenho de Sistemas Computacionais e de Comunicação*.

Gavalas, D., Venetis, I. E., Konstantopoulos, C. e Pantziou, G. (2017). Mobile agent itinerary planning for WSN data fusion: considering multiple sinks and heterogeneous networks. *International Journal of Communication Systems*, 30(8):e3184.

Liu, X. e Baiocchi, O. (2016). A comparison of the definitions for smart sensors, smart objects and things in IoT. Em *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, p. 1–4. IEEE.

Lu, J., Feng, L., Yang, J., Hassan, M. M., Alelaiwi, A. e Humar, I. (2019). Artificial agent: The fusion of artificial intelligence and a mobile agent for energy-efficient traffic control in wireless sensor networks. *Future Generation Computer Systems*, 95:45–51.

Mišić, J. e Mišić, V. B. (2018). Proxy cache maintenance using multicasting in CoAP IoT domains. *IEEE Internet of Things Journal*, 5(3):1967–1976.

Rahmani, A.-M., Thanigaivelan, N. K., Gia, T. N., Granados, J., Negash, B., Liljeberg, P. e Tenhunen, H. (2015). Smart e-health gateway: Bringing intelligence to internet-of-things based ubiquitous healthcare systems. Em *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*, p. 826–834. IEEE.

Urra, O. e Ilarri, S. (2019). Spatial crowdsourcing with mobile agents in vehicular networks. *Vehicular Communications*, 17:10–34.

Xu, X., Li, X.-Y., Wan, P.-J. e Tang, S. (2012). Efficient scheduling for periodic aggregation queries in multihop sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 20(3):690–698.