

APLICAÇÃO DO RPL EM VANETS PARA AVISO PÓS COLISÃO

Fábio Gonçalves de Oliveira e Carlos Marcelo Pedroso

Resumo—O presente trabalho tem como objetivo avaliar o desempenho do protocolo de roteamento RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) em cenários de notificação pós colisão veicular, realizando uma variação sistemática dos parâmetros de configuração da função objetivo e do *trickle timer* a fim de verificar o seu impacto na aplicação.

Palavras-Chave—RPL, VANETs, COOJA, CONTIKI.

Abstract—The present work aims to evaluate the performance of the RPL routing protocol (IPv6 Routing Protocol for Low-Power and Lossy Networks) in post-vehicle collision notification scenarios, carrying out a systematic variation of the objective function and trickle timer configuration parameters in order to verify their impact on the application.

Keywords—RPL, VANETs, COOJA, CONTIKI.

I. INTRODUÇÃO

Estima-se que, em 2015, havia cerca de 1,1 bilhão de veículos de pequeno porte no mundo. Espera-se que, até 2050, esse número atinja 1,75 bilhão [1]. Esse aumento significativo na quantidade de veículos nas estradas traz consigo uma preocupação relacionada à segurança. Conforme indicado no Relatório Global de Segurança Rodoviária publicado pela OMS (Organização Mundial da Saúde) em 2018, aproximadamente 1,35 milhão de vidas são perdidas anualmente em acidentes de trânsito, e até 50 milhões de pessoas sofrem ferimentos [2]. Neste contexto, tem se consolidado cada vez mais o conceito de ITS (*Intelligent Transportation System*), que utiliza uma variedade de tecnologias avançadas para melhorar a eficiência e, principalmente, a segurança das estradas [3]. Dentro desse cenário, as VANETs (*Vehicle Ad Hoc Networks*) têm emergido como uma abordagem promissora para aprimorar a segurança viária, permitindo a comunicação entre veículos e infraestruturas rodoviárias [4], com características próprias decorrentes da presença de dispositivos em movimento em alta velocidade. Entre essas características, destaca-se [5] a alta mobilidade, que resulta em atrasos, a topologia altamente dinâmica, uma vez que os nós mudam de posição de forma rápida e constante, a relação crítica com o tempo e a volatilidade.

De modo geral, a estrutura básica de uma VANET consiste em três diferentes tipos de dispositivos: OBU (*On-Board Unit*), RSUs (*Road-Side Unit*) e TAs (*Trusted Authority*) [6]. As OBUs coletam e processam dados nos veículos. Elas também são responsáveis por realizar a comunicação com as RSUs e com outros veículos [4]. As RSUs são os dispositivos

instalados nas ruas, estradas e rodovias. Elas permitem que os veículos tenham acesso à internet, aumentam o alcance das VANETs e alertam sobre possíveis incidentes e condições de trânsito [7]. Por fim, os TAs são os elementos responsáveis por gerenciar a rede como um todo, fazendo a autenticação e remoção de veículos e estão conectados via cabo com as RSUs, possuindo maior poder de processamento e armazenamento quando comparados aos outros dispositivos [4] [7].

A transmissão de mensagens de emergência representa um grande desafio às VANETs, já que este tipo de rede possui características intrínsecas que propiciam o aumento do atraso e da perda de pacotes. O alerta pós-colisão nada mais é do que um tipo de mensagem de emergência, com restrições de atraso máximo e PDR (*Packet Delivery Ratio*), que devem ser atingidos para garantir a sua eficiência. Para sua categoria, são estabelecidos um atraso fim-a-fim máximo de 100 ms, um alcance de 300 metros, um PDR mínimo de 95% e uma taxa de atualização de 1 Hz [8] [9] [10] [11].

O presente trabalho tem como objetivo avaliar o desempenho do protocolo de roteamento RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) em cenários de notificação pós-colisão veicular. Para alcançá-lo, foram conduzidas simulações de cenários de colisão por meio do software Cooja, com o uso do sistema operacional Contiki, variando sistematicamente os parâmetros de configuração da função objetivo e temporizadores. As principais modificações no RPL para atender cenários de mobilidade comparam resultados com a configuração original do protocolo, que foi projetado para redes estáticas, porém a variação de parâmetros de configuração e temporização podem melhorar consideravelmente o desempenho do protocolo original.

Além desta seção introdutória, o artigo está organizado da seguinte forma: na seção 2 é conduzida uma revisão de conceitos fundamentais referentes ao RPL, abrangendo os conceitos de DODAGs (*Destination Oriented Directed Acyclic Graph*), funções objetivo, *trickle timer* e um resumo de trabalhos que propõem melhorar o desempenho do RPL em cenários com mobilidade. Em seguida, na seção 3, é discutido o desempenho do protocolo efetivamente, explicando como a aplicação foi desenvolvida, os cenários de teste realizados, mostrando e explicando os resultados obtidos. Por fim, na seção 4 é realizada a conclusão do trabalho.

II. RPL

Definido como padrão em 2012, o RPL é um protocolo de roteamento IPv6 (*Internet Protocol Version 6*) padronizado pela IETF através da RFC 6550 para redes LLN (*Low Power*

and Lossy Network) [12], utilizando o algoritmo de vetor de distância baseado em DODAGs, nas quais as conexões entre os nós são direcionadas a um nó raiz [13].

A. DODAGs

As DODAGs são estruturas hierárquicas, cujos nós estabelecem entre si uma relação de pai e filho, na qual um nó elege outro como seu pai caso seja definido que ele está em uma posição melhor em relação ao raiz. Essa qualidade de posicionamento é denominada *ranking* e é definida através de uma função objetivo. A Figura 1 ilustra a estrutura de uma DODAG demonstrando seu sentido de comunicação.

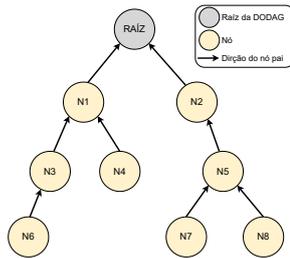


Fig. 1. Topologia da DODAG.

O processo de formação da rede é iniciado com o raiz enviando uma mensagem do tipo DIO (*DODAG Information Object*), que carrega as informações relevantes referentes à DODAG, como seu ID, o ID da instância à qual ela pertence, seu modo de operação e o ponto de código da função objetivo [14]. O nó raiz estabelece para si o *rank* 0 [15]. Ao receber um DIO, o nó o retransmite, atualizando o valor do *rank* com o valor obtido através da função objetivo. Além disso, o nó realiza o envio de uma mensagem chamada DAO (*Destination Advertisement Object*) para que seja definida a rota de *downward* [16].

B. Função objetivo

Encarregada de estabelecer as métricas e os cálculos para a obtenção do *ranking* do nó [17], a função objetivo não se vincula a um algoritmo ou equação específicos definidos pela RFC. Essa flexibilidade permite sua adaptação a uma diversidade de cenários de uso [18]. A RFC 6551 delinea uma ampla gama de métricas, agrupando-as em duas categorias principais: métricas de *link* e métricas de nó [19]. Apesar de seu uso não ser mandatário, foram padronizadas duas funções objetivo: a *Objective Function Zero* (OF0), conforme estabelecido pela RFC 6552 e a *Minimum Rank Hysteresis Objective Function* (MRHOF), conforme definido pela RFC 6719 [20].

A OF0 foi a primeira proposta oficial de função objetivo, ela se baseia no número de saltos do nó até o raiz e quanto menor esse número, menor é o valor do *ranking* [21]. Logo, para ela, o menor caminho será sempre o melhor. Por utilizar apenas essa métrica, a função pode acabar gerando problemas, já que mesmo com um baixo número de saltos o nó pode não ter uma conexão confiável com o raiz, gerando retransmissões e perdas desnecessárias de pacotes. Como essa métrica é estática

e referente ao dispositivo, por mais que a condição da rota do ponto de vista do *link* esteja ruim, ela não é alterada já que o número de saltos se mantém o mesmo [5].

Por conta desses problemas apresentados pela OF0, a MRHOF foi desenvolvida focando o seu funcionamento em uma métrica dinâmica de *link*, que é especificada na mensagem de DIO. A função a utiliza como base para calcular o custo do caminho de todos os vizinhos, escolhendo inicialmente o de menor custo e definindo o seu *rank* a partir disso [20]. Além disso, existe um segundo mecanismo que é utilizado para a troca de parentesco, que só é realizada caso seja descoberto um novo caminho, cujo custo seja menor que o atual menos um valor de histerese [20].

C. Trickle timer

O *trickle timer* é um dos componentes responsáveis pela manutenção da rede, definido pela RFC 6206 [22], é o algoritmo responsável por controlar o disparo das mensagens de DIO após a sua formação, regulando assim o fluxo de mensagens de controle [12]. A regulação é realizada através da supressão de mensagens e com a alteração exponencial do intervalo de envio [22]. O algoritmo utiliza como base três parâmetros: um intervalo mínimo I_{min} , um máximo I_{max} e uma constante de redundância k [23].

Seu funcionamento se dá da seguinte forma: Durante a inicialização, o intervalo de disparo é configurado para ser o mesmo de I_{min} e o contador de inconsistências c é zerado. No começo de cada intervalo I , c é zerado e é definido um valor de tempo aleatório t , que é restringido entre o intervalo $I/2$ e I [23]. Quando o tempo t é atingido a mensagem DIO é transmitida caso c seja menor que k , no momento em que I é atingido o seu valor é duplicado, sendo isto feito até que alcance I_{max} . Caso seja detectada alguma inconsistência, o valor de I retorna para I_{min} e o contador c é zerado [24].

D. Trabalhos relacionados

A fim de atenuar os impactos relacionados à mobilidade, diversos trabalhos foram desenvolvidos com diferentes propostas e focos de alterações. A seguir, serão descritos os principais trabalhos desta área.

O mRPL [25] tem como base de seu funcionamento um mecanismo de *hand-off* denominado *smart-HOP*, que visa reduzir o tempo de troca de parentesco. Fica sob responsabilidade do pai identificar a movimentação do nó filho e solicitar que ele se desconecte e busque outro pai quando o ARSSI (*Average Received Signal Strength*) atingir um valor limite mínimo. Ao se desconectar, o nó faz o disparo de uma série de mensagens de DIS (*DODAG Information Solicitation*), que são utilizadas para solicitar informações de uma DODAG, e aguarda as respostas. Os testes foram realizados com nove nós estáticos e um nó se movimentando com uma velocidade de 2 m/s. Neste cenário, o mecanismo atingiu um PDR de quase 100% com atrasos menores que 95 ms, enquanto o RPL padrão atingiu um PDR de cerca de 50%, não sendo informado o seu atraso.

Em sua proposta Hoghooghi e Javidan [26], assim como no mRPL, os autores visam reduzir os problemas decorrentes

da desconexão dos nós. O que a difere dos outros trabalhos citados anteriormente é que seu mecanismo não foca apenas em reagir às mudanças de topologia, mas sim detectar ativamente a mobilidade do nó, fazendo com que ele busque um novo pai antes de perder a conexão com o anterior, resultando na continuidade da conexão daquele dispositivo na rede. Essa detecção é realizada pelo pai, que solicita que o nó em movimento busque um novo pai, o qual faz isso através de mensagens de DIS. Os testes foram realizados com os nós movendo-se a até 2 m/s. Com esta velocidade, consistindo no pior caso, a solução atingiu um PDR de 96,12% contra 89,26% do RPL padrão, já seu atraso fim-a-fim ficou em 231 ms contra os 359 ms do padrão, resultando em uma melhora tanto no atraso quanto no PDR.

Através do MARPL, o autor propõe um novo mecanismo de detecção de movimento e de detecção de indisponibilidade do nó pai [27]. Sua inovação em relação às abordagens citadas está no fato de que a detecção de movimento é realizada através da variação do RSSI. Além disso, ele utiliza um temporizador para monitorar se o pai está ou não fora de alcance. Quando o temporizador estoura, caso o nó não tenha recebido nenhuma mensagem dele nesse intervalo, ele dispara mensagens de DIS para procurar novos pais. Assim como o trabalho anterior, os testes foram realizados em baixas velocidades, até 3 m/s, contendo 30 nós móveis e um raiz. Neste cenário, o método atingiu um PDR de 22,13% e um atraso de 305,93 ms. Já o RPL padrão obteve 8,76% de PDR e um atraso de 251,21 ms.

Por fim, o POF (*Programmable Objective Function*), proposto por Fabian et Al. [28], utiliza lógica fuzzy para adaptar o uso das métricas de acordo com o cenário, podendo focar em apenas uma ou em um conjunto delas. Seu principal objetivo é aumentar o QoS (*Quality of Service*) em cenários IoV (*Internet of Vehicles*). Os testes propostos foram os mais completos dos avaliados, com quatro cenários distintos, nos quais foram variadas as velocidades máximas que atingiram até 54 km/h. Na velocidade mais alta, o PDR atingido foi de cerca de 14% com um atraso fim-a-fim de cerca de 400 ms. Em contrapartida, a OF0 atinge 10% de PDR e 390 ms de atraso, já a MRHOF não atinge nem 1% de PDR e seu atraso fica por volta de 200 ms.

III. AVALIAÇÃO DE DESEMPENHO

A. Aplicação

O fluxo da aplicação se dá da seguinte forma: o veículo que sofreu o acidente é responsável por enviar ao nó raiz da rede, no caso uma RSU, a mensagem informando a colisão. Ela irá conter as coordenadas do incidente, e será reenviada a cada 1 segundo. Ao recebe-la, o nó raiz irá confirmar ao veículo e, a partir disso, a RSU irá retransmitir a mensagem para redes próximas que possam estar no alcance dos 300 metros requisitados pela aplicação. Por fim, a mensagem é retransmitida para todos os veículos dessas redes.

B. Cenários

Os testes foram projetados para analisar o impacto de diferentes fatores no desempenho do RPL e a performance

das funções objetivo OF0 e MRHOF (com métricas ETX e RSSI). O mapa, o modelo de perda, o tipo de rádio, o modelo de mobilidade e a faixa de velocidade dos veículos foram constantes em todos os cenários, em ambas as etapas foram realizadas uma rodada de testes.

O mapa tinha 10 quadras horizontais e 10 verticais, cobrindo uma área de 1000 m x 1000 m, simulando uma zona urbana. O modelo de mobilidade utilizado foi o Manhattan Grid. O modelo de perda foi o UDGM (*Unit Disk Graph Medium*), disponível no emulador Cooja. O rádio utilizado foi o IEEE 802.15.4, operando a 2,4 GHz, com 16 canais, cada um com 2 MHz de largura de banda e 5 MHz de espaçamento entre eles [29] [30]. A velocidade dos veículos variou entre 10 km/h e 80 km/h.

Os testes foram realizados em duas etapas. Na primeira, variaram-se apenas valores externos ao protocolo, como o número de veículos (85, 150, 200 e 300), o número de RSUs (1 e 5) e o alcance do rádio (100 m e 400 m).

Na segunda etapa, variaram-se os valores do *trickle timer* e a histerese das métricas na função objetivo MRHOF. Para estas simulações, o número de veículos foi fixado em 150, foi utilizada uma RSU e o alcance dos rádios foi definido como 100 m, conforme a limitação da tecnologia [31].

C. Resultados e discussão

A Tabela I apresenta os resultados obtidos na primeira etapa de testes para as funções objetivo OF0, MRHOF com métrica de ETX e MRHOF com métrica de RSSI. Ela está organizada da seguinte forma: as três primeiras colunas correspondem aos parâmetros das simulações, em seguida estão presentes os resultados para as duas funções testadas, apresentando o PDR, atraso, número de carros notificados (NCN), número de carros notificados em porcentagem e distância média do primeiro aviso (DMPA)

Os resultados indicam que componentes externos têm um grande impacto na aplicação. Em cenários com menor alcance, a OF0 obteve melhores resultados quanto ao número de carros notificados e à distância da primeira notificação. Já em cenários de maior alcance, a MRHOF com RSSI se destacou em todas as métricas, especialmente no menor atraso fim-a-fim em comparação com outras funções. Embora o RSSI tenha um PDR maior em cenários de menor alcance, apresentou as menores taxas de notificação de veículos. A MRHOF com ETX destacou-se apenas na maior distância de primeiro aviso em cenários de maior alcance de rádio. É importante notar que os resultados variam significativamente entre os cenários, com o PDR variando de 1,76% a 76,65%, e o mesmo se aplica às demais métricas.

Já para a segunda etapa, partindo da função objetivo, o primeiro parâmetro variado foi a histerese de cada métrica da MRHOF. Para o RSSI, seu valor de histerese foi variado de 1 dB até 85 dB, conforme mostrado no gráfico da Figura 2

Em seguida, o mesmo foi feito para o ETX, variando o valor de histerese de 0,0078125 até 8. Conforme a RFC 6551 [32], esses valores devem ser multiplicados por 128 e arredondados para o inteiro mais próximo, resultando em um intervalo de 1 a 1024. Os resultados estão na Figura 3

Parâmetros		OF0					MRHOF – ETX					MRHOF – RSSI					
RSUs	Alcance rádio (m)	NC	PDR (%)	Atraso (ms)	NCN (%)	DMPA (m)	PDR (%)	Atraso (ms)	NCN (%)	DMPA (m)	PDR (%)	Atraso (ms)	NCN (%)	DMPA (m)			
1	100	85	24,14%	75,40	44	51,76%	81,21	24,14%	75,40	44	51,76%	76,85	21,30%	55,23	24	28,24%	72,41
		150	11,86%	86,38	74	49,33%	85,35	11,74%	82,69	45	30,00%	70,87	24,92%	58,51	45	30,00%	58,8
		200	16,92%	140,56	108	54,00%	103,92	14,75%	116,21	41	20,50%	80,38	28,88%	65,25	36	18,00%	64,03
	400	85	46,07%	603,38	85	100,00%	436,71	40,21%	2.044,84	85	100,00%	436,78	76,65%	72,71	85	100,00%	397,19
		150	10,95%	3.016,88	142	94,67%	401,72	24,01%	2.405,77	148	98,67%	396,95	63,31%	93,45	148	98,67%	370,69
		200	3,57%	5.424,16	101	50,50%	339,16	11,30%	4.100,25	149	74,50%	342,8	31,49%	322,69	182	91,00%	351,51
5	100	85	14,58%	52,97	51	60,00%	232,76	14,45%	52,97	51	60,00%	232,77	19,38%	51,68	36	42,35%	216,57
		150	12,77%	65,18	75	50,00%	237,68	11,88%	56,40	51	34,00%	228,8	25,13%	52,50	69	46,00%	242,97
		200	15,07%	70,71	115	57,50%	251,21	15,49%	63,88	63	31,50%	241,1	25,22%	50,80	79	39,50%	217,41
	400	85	49,93%	605,31	85	100,00%	443,69	54,01%	708,60	85	100,00%	449,56	70,54%	75,38	85	100,00%	434,76
		150	8,13%	8.022,37	143	95,33%	378,58	22,81%	5.877,56	150	100,00%	423,64	57,42%	150,43	150	100,00%	374,52
		200	5,75%	16.099,66	113	56,50%	399,8	13,82%	5.036,56	195	97,50%	439,48	43,22%	213,25	199	99,50%	393,94
		300	2,50%	10.552,07	28	9,33%	309,57	5,72%	11.548,19	124	41,33%	356,017	16,73%	1.950,39	203	67,67%	354,98

TABELA I
RESULTADOS DAS SIMULAÇÕES VARIANDO PARÂMETROS EXTERNOS

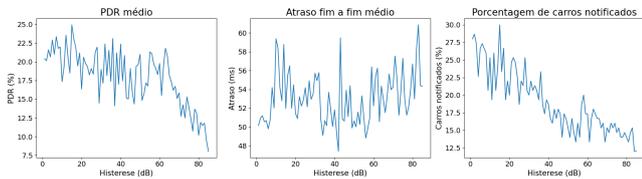


Fig. 2. Resultados com a variação da histerese de RSSI.

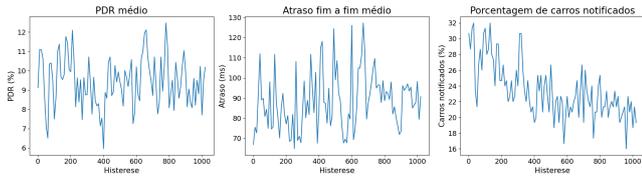


Fig. 3. Resultados com a variação da histerese de ETX.

Os gráficos mostram o impacto da histerese no desempenho do protocolo. Em ambas as métricas, há uma tendência de diminuição no número de carros notificados à medida que a histerese aumenta, pois a rede sofre menos alterações, com os nós trocando menos de país. No caso do RSSI, essa queda é mais drástica do que no ETX. Contudo, o atraso fim-a-fim e o PDR do RSSI foram melhores na maioria dos casos, com o atraso nunca ultrapassando 100 ms, ao contrário do ETX, que excedeu esse valor em muitos casos.

Por fim, as funções foram submetidas à variação do valor mínimo do *trickle timer*, variando de 4 ms até 16,384 s, calculado conforme mostrado a seguir

$$I_{min} = 2^n \quad (1)$$

onde n é o expoente configurado no teste e I_{min} é o tempo dado em ms. Os gráficos obtidos estão em função do expoente n , que varia de 2 até 14. A Figura 4 demonstra, respectivamente, os Gráficos de PDR, atraso fim-a-fim e de números de carros notificados em relação à variação do valor do temporizador para a função objetivo OF0, MRHOF com métrica de ETX e MRHOF com métrica de RSSI.

O valor do *trickle timer* altera diretamente o desempenho do RPL. Valores mais baixos tendem a aumentar o atraso,

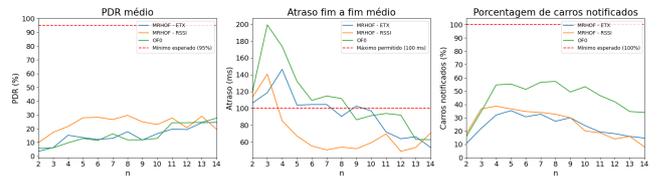


Fig. 4. Resultados com a variação do *trickle timer*

enquanto o PDR tende a diminuir. Em relação ao número de carros notificados, este acaba sofrendo tanto com valores muito baixos quanto com valores muito altos. O OF0 foi o que conseguiu atingir o maior número de veículos, mas, em compensação, resultou em maiores valores de atraso. Nesta configuração de histerese, o RSSI obteve um número maior de veículos notificados do que o ETX, ficando atrás apenas do OF0. Além disso, o MRHOF com RSSI apresentou os melhores valores de PDR e atraso, com este último estando abaixo dos 100 ms na maior parte do tempo.

IV. CONCLUSÕES E TRABALHOS FUTUROS

O RPL, com suas funções objetivo padrão, não foi concebido para lidar com sistemas que apresentam mobilidade. A principal dificuldade reside na capacidade da rede em manter a convergência mesmo diante de alta mobilidade dos nós. Enquanto os nós que transitam rapidamente pela área de alerta precisam ser integrados à rede e informados sobre o acidente sem demora, eles não devem comprometer a comunicação dos nós já estabelecidos. Isso requer que o protocolo seja ágil na detecção e tratamento de mudanças na topologia da rede.

O desempenho do RPL é fortemente influenciado pelas configurações do *trickle timer* e da função objetivo. A alteração dos parâmetros do RPL original resultou em ganhos comparáveis ou até maiores, em alguns casos, quando comparados com as propostas de alteração do protocolo para suportar redes móveis. Desta forma, a adaptação do RPL a situações de mobilidade permanece um tema de pesquisa em aberto.

REFERÊNCIAS

- [1] A. Ghandi and S. Paltsev, "Global co2 impacts of light-duty electric vehicles," *Transportation Research Part D: Transport and Environment*, vol. 87, p. 102524, 2020.

- [2] WHO, “Global status report on road safety 2018,” World Health Organization, Geneva, Tech. Rep., 2018.
- [3] M. N. Mejri, J. Ben-Othman, and M. Hamdi, “Survey on vanet security challenges and possible cryptographic solutions,” *Vehicular Communications*, vol. 1, no. 2, pp. 53–66, 2014.
- [4] M. S. Sheikh, J. Liang, and M. A. Khan, “A comprehensive survey on vanet security services in traffic management system,” *Wirel. Commun. Mob. Comput.*, vol. 2019, jan 2019.
- [5] A. Brachman, “Rpl objective function impact on llns topology and performance,” in *Internet of Things, Smart Spaces, and Next Generation Networking*, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 340–351.
- [6] F. Azam, S. K. Yadav, N. Priyadarshi, S. Padmanaban, and R. C. Bansal, “A comprehensive review of authentication schemes in vehicular ad-hoc network,” *IEEE Access*, vol. 9, pp. 31 309–31 321, 2021.
- [7] P. Mundhe, S. Verma, and S. Venkatesan, “A comprehensive survey on authentication and privacy-preserving schemes in vanets,” *Computer Science Review*, vol. 41, p. 100411, 2021.
- [8] Q. Xu, T. Mak, J. Ko, and R. Sengupta, “Vehicle-to-vehicle safety messaging in dsrc,” in *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, ser. VANET ’04. New York, NY, USA: Association for Computing Machinery, 2004, p. 19–28.
- [9] V. Nguyen, O. T. T. Kim, C. Pham, T. Z. Oo, N. H. Tran, C. S. Hong, and E.-N. Huh, “A survey on adaptive multi-channel mac protocols in vanets using markov models,” *IEEE Access*, vol. 6, pp. 16 493–16 514, 2018.
- [10] A. T. Sasongko, G. Jati, B. Hardian, and W. Jatmiko, “The reliability of routing protocols as an important factor for road safety applications in vanet-based autonomous cars,” *Journal of Computer Science*, vol. 16, no. 6, pp. 768–783, Jun 2020.
- [11] “Vehicle safety communications project task 3 final report: Identify intelligent vehicle safety applications enabled by dsrc,” Mar 2005, tech Report. [Online]. Available: <https://rosap.ntl.bts.gov/view/dot/3925>
- [12] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” RFC 6550, Mar. 2012.
- [13] F. ARAT and S. DEMİRCİ, “Experimental analysis of energy efficient and QoS aware objective functions for RPL algorithm in IoT networks,” *Sakarya University Journal of Computer and Information Sciences*, vol. 4, pp. 192–203, 08 2021.
- [14] O. Gaddour, A. Koubâa, S. Chaudhry, M. Tezeghdanti, R. Chaari, and M. Abid, “Simulation and performance evaluation of DAG construction with rpl,” in *Third International Conference on Communications and Networking*, 2012, pp. 1–8.
- [15] H.-S. Kim, J. Ko, D. E. Culler, and J. Paek, “Challenging the ipv6 routing protocol for low-power and lossy networks (rpl): A survey,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2502–2525, 2017.
- [16] F. Gara, L. Ben Saad, E. Ben Hamida, B. Tourancheau, and R. Ben Ayed, “An adaptive timer for RPL to handle mobility in wireless sensor networks,” in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2016, pp. 678–683.
- [17] V. C. Diniesh, G. Murugesan, M. J. Auxilius Jude, A. Harshini, S. Bhavataarani, and R. G. Krishnan, “Impacts of objective function on RPL-routing protocol: A survey,” in *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2021, pp. 251–255.
- [18] B. Safaei, A. A. Mohammad Salehi, A. M. Hosseini Monazzah, and A. Ejlali, “Effects of RPL objective functions on the primitive characteristics of mobile and static IoT infrastructures,” *Microprocessors and Microsystems*, vol. 69, pp. 79–91, 2019.
- [19] H. Lamaazi and N. Benamar, “A comprehensive survey on enhancements and limitations of the RPL protocol: A focus on the objective function,” *Ad Hoc Networks*, vol. 96, p. 102001, 2020.
- [20] O. Gnawali and P. Levis, “The Minimum Rank with Hysteresis Objective Function,” RFC 6719, Sep. 2012.
- [21] P. Thubert, “Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL),” RFC 6552, Mar. 2012.
- [22] A. Musaddiq, Y. B. Zikria, and S. W. Kim, “Energy-aware adaptive trickle timer algorithm for RPL-based routing in the internet of things,” in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–6.
- [23] P. Levis, T. H. Clausen, O. Gnawali, J. Hui, and J. Ko, “The Trickle Algorithm,” RFC 6206, Mar. 2011.
- [24] K. C. Lee, R. Sudhaakar, L. Dai, S. Addepalli, and M. Gerla, “Rpl under mobility,” in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*, 2012, pp. 300–304.
- [25] H. Fotouhi, D. Moreira, and M. Alves, “mRPL: Boosting mobility in the internet of things,” *Ad Hoc Networks*, vol. 26, pp. 17–35, 2015.
- [26] S. Hoghooghi and R. Javidan, “Proposing a new method for improving RPL to support mobility in the internet of things,” *IET Networks*, vol. 9, no. 2, pp. 48–55, 2020.
- [27] V. Marques and J. Kniess, “Mobility aware RPL (marpl): Providing mobility support for RPL protocol,” in *Anais do XXXVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Porto Alegre, RS, Brasil: SBC, 2019, pp. 211–223.
- [28] P. Fabian, A. Rachedi, and C. Guéguen, “Programmable objective function for data transportation in the internet of vehicles,” *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 5, p. e3882, 2020.
- [29] C. M. Ramya, M. Shanmugaraj, and R. Prabakaran, “Study on zigbee technology,” in *2011 3rd International Conference on Electronics Computer Technology*, vol. 6, 2011, pp. 297–301.
- [30] R. Natarajan, P. Zand, and M. Nabi, “Analysis of coexistence between IEEE 802.15.4, ble and IEEE 802.11 in the 2.4 ghz ISM band,” in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, pp. 6025–6032.
- [31] N. H. Hussein, C. T. Yaw, S. P. Koh, S. K. Tiong, and K. H. Chong, “A comprehensive survey on vehicular networking: Communications, applications, challenges, and upcoming research directions,” *IEEE Access*, vol. 10, pp. 86 127–86 180, 2022.
- [32] D. Barthel, J. Vasseur, K. Pister, M. Kim, and N. Dejean, “Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks,” RFC 6551, Mar. 2012.