

# XORBF: Roteamento Baseado em uma Métrica de OU-Exclusivo e Filtros de Bloom para Redes Veiculares Urbanas

Ederval P. F. da Cruz<sup>1</sup>, Bruno Silva<sup>2</sup>, Carlos A. V. Campos<sup>2</sup>,  
Rodolfo Oliveira<sup>3</sup>, Rafael Pasquini<sup>4</sup>

<sup>1</sup>Instituto Federal do Espírito Santo (IFES) – Campus Itapina

<sup>2</sup>Universidade Federal do Estado do Rio de Janeiro (DIA/UNIRIO)

<sup>3</sup>Universidade Nova de Lisboa (FCT/UNL)

<sup>4</sup>Universidade Federal de Uberlândia (FACOM/UFU)

ederval.cruz@ifes.edu.br, {bruno.silva, beto}@uniriotec.br

rado@fct.unl.pt, pasquini@facom.ufu.br

**Abstract.** *This paper presents an XOR-based routing protocol, called XORBF, for vehicular ad hoc networks (VANETs) developed in urban scenarios, that leverages the probabilistic properties of Bloom filters for neighborhood discovery purposes. It describes the XOR-based routing principle, detailing the neighborhood discovery process developed using Bloom filters. In short, the proposed procedure provides signaling hints useful for routing tables' convergence, that is critical due to the high mobility level of VANETs. Its performance is compared through simulation with other protocols available in the literature, showing its superiority of up to 95% in the end-to-end delay for establishing paths between vehicles, an increase of the their duration and stability which is achieved by use of Bloom filters.*

**Resumo.** *Este artigo apresenta um protocolo de roteamento baseado em uma métrica de OU-Exclusivo (XOR), chamado XORBF, desenvolvido para redes ad hoc veiculares urbanas, e que faz uso das propriedades probabilísticas dos filtros de Bloom para efetuar a descoberta de vizinhança. Ele irá descrever o princípio do roteamento baseado na métrica XOR, detalhando o processo de descoberta de vizinhos usando filtros de Bloom. Para isso, o mecanismo proposto oferece informações de sinalização, úteis para a convergência das tabelas de roteamento, que é crítico dado o elevado nível de mobilidade encontrado nas redes veiculares. O desempenho do protocolo proposto é comparado, através de simulação, com outros protocolos disponíveis na literatura, revelando-se superior em até 95% no atraso fim-a-fim para o estabelecimento de caminhos entre pares de veículos, um aumento de sua duração e estabilidade, obtido pelo uso dos filtros de Bloom.*

## 1. Introdução

Uma área de pesquisa que tem recebido atenção de diversos grupos de pesquisa e da indústria automobilística é a rede ad hoc veicular (*Vehicular ad hoc Network* - VANET)

[Lin et al. 2010]. Esta tecnologia introduz o conceito de comunicação de dados entre veículos, oferecendo suporte ao desenvolvimento de novas aplicações que forneçam, por exemplo, entretenimento, conforto aos ocupantes dos veículos e maiores níveis de segurança para a utilização de veículos e auxílio no atendimento mais eficaz a vítimas de acidentes ou no transporte de pacientes.

Algumas das características desafiadoras em VANET incluem o alto grau de mobilidade, tempos reduzidos de comunicação e o tratamento de um elevado número de veículos em situações de congestionamento. As VANETs constituem um vasto campo de pesquisa, em que há frequentes investigações por soluções escaláveis em todas as camadas de comunicação, desde as voltadas ao meio físico de transmissão até o desenvolvimento de novas aplicações. Neste trabalho, elas estão centradas no desenvolvimento de soluções escaláveis de roteamento, visando uma comunicação eficaz entre os veículos que constituem uma VANET. Basicamente, os protocolos propostos para VANET são classificados em dois grupos: 1) protocolos baseados na topologia da rede e 2) protocolos baseados na localização geográfica dos nós.

Os protocolos baseados na topologia, tais como o *Ad Hoc On-Demand Distance Vector* (AODV) [Perkins et al. 2003], e *Optimized Link State Routing* (OLSR) [Clausen and Jacquet 2003], desempenham sua função de roteamento onde caminhos fim-a-fim são criados. Por outro lado, os protocolos baseados na localização geográfica dos nós, tais como o GPSR [KARP 2000], GeoSVR [Xiang et al. 2013] e GPCR [Lin et al. 2010] baseiam-se no conhecimento da posição atual dos nós na rede para efetuar o roteamento de pacotes. Em ambos os casos há uma troca frequente de mensagens de sinalização para manter as informações sempre atualizadas, que é um fator preocupante em VANETs, especialmente em cenários com elevado número de veículos, como os ambientes urbanos. Isso pode levar a atrasos significativos durante o estabelecimento de caminhos fim-a-fim entre pares de veículos.

Dentro desse contexto, é proposto o protocolo de roteamento pró-ativo XORBF (*Exclusive-OR Bloom Filter*) para VANETs baseado em uma métrica de OU-Exclusivo (XOR) e filtros de Bloom (*Bloom Filter - BF*). A métrica XOR utiliza uma tabela de roteamento cuja ordem é logarítmica ao espaço total de identificação (IDs) dos veículos [CRUZ et al. 2012], requerendo menor quantidade de informação sobre os nós de uma VANET, tornando, desta forma, mais eficiente a convergência das tabelas de roteamento em VANET. Além disso, o mecanismo de descoberta de vizinhança proposto no XORBF utiliza as propriedades probabilísticas dos filtros de Bloom, de tal forma a impedir que informações de vizinhança repetidas e/ou desnecessárias sejam trocadas entre os veículos da VANET. As avaliações apresentadas neste artigo incluem a análise do protocolo XORBF em relação a protocolos baseados na topologia (AODV e OLSR) e baseado na localização dos nós (GPSR), considerando o uso de VANET em cenários urbanos, através de simulação em que são comparadas a taxa de caminhos criados, seu comprimento médio, o tempo médio de duração e o atraso médio fim-a-fim para seu estabelecimento em sete cenários.

O restante deste artigo está organizado da seguinte forma: A Seção 2 apresenta os trabalhos relacionados. A Seção 3 detalha o protocolo XORBF, apresentando o mecanismo de roteamento XOR e o processo de descoberta de vizinhança que utiliza os filtros de Bloom. A Seção 4 descreve os cenários utilizados nas simulações realizadas e

apresenta os resultados obtidos. A Seção 5 finaliza o artigo.

## 2. Trabalhos Relacionados

Diversos trabalhos comparam o desempenho de propostas de roteamento em VANETs [Lin et al. 2010, Miao et al. 2013, Martinez et al. 2013, HAERRI et al. 2006], sendo recorrente a presença dos protocolos AODV e OLSR. Ambos são especificados em RFCs do IETF, sendo natural a utilização destes nas avaliações apresentadas neste trabalho. Uma vez que ambos os protocolos são baseados em informação topológica, adotamos o GPSR como um terceiro protocolo nas avaliações, por se tratar de um protocolo baseado na localização geográfica dos nós e que também é avaliado em diversos trabalhos envolvendo VANETs [Nzouonta et al. 2009, Naumov et al. 2007].

No trabalho de [Oliveira et al. 2011] é apresentado o protocolo XOR, onde é realizada a primeira avaliação deste protocolo em redes móveis e o mesmo é avaliado em um cenário de rodovia. Devido ao fato de haver muitas trocas de mensagens de sinalização do protocolo XOR, em busca de preencher as tabelas de roteamento dos veículos, ele obtém baixo desempenho no atraso. Já em ambiente urbano, o trabalho de [CRUZ et al. 2012] demonstra que quanto maior o valor do parâmetro  $H$  (detalhado na Seção 3 deste trabalho) maior é o valor de atraso nos cenários avaliados, além de impactar em outras métricas citadas. Isso ocorre devido a alta carga de sinalização gerada para preenchimento das tabelas de roteamento dos veículos, especialmente nos cenários com maior densidade veicular. Por sua vez, [Xiang et al. 2013] é proposto um protocolo de roteamento geográfico avaliado em cenário urbano onde os pacotes são roteados levando em consideração a posição dos nós e um mapa digital. Este protocolo consiste de dois algoritmos principais chamados de algoritmo de encaminhamento ótimo e encaminhamento restrito.

É proposto em [Correia et al. 2011] uma otimização do protocolo reativo DYMO, chamado de MAR-DYMO, que é inspirado em colônia de formigas. Nele, as informações como velocidade, deslocamento, posição, mobilidade e um timestamp são adicionados na mensagem de HELLO e enviados somente quando necessário. Dessa forma o nó terá uma tabela com informações atualizadas de seus vizinhos. Isso faz com que um veículo faça predições sobre eles tais como sua posição em um certo instante e até que momento eles ainda serão vizinhos.

O AODV é um protocolo baseado em informação topológica que opera de maneira reativa, ou seja, quando um nó de origem deseja enviar dados para um nó de destino, um processo de descoberta de rotas é iniciado. O problema deste protocolo é que o nó de origem, em busca de nós para preencher sua tabela de roteamento, realiza *broadcast* de mensagens para todos os seus vizinhos. Caso os nós vizinhos desconheçam uma rota para o nó de destino requisitado, é realizado um *broadcast* para os vizinhos dos vizinhos, até que uma rota seja encontrada ou que um número máximo de saltos seja alcançado, caracterizando a inexistência de rota para aquele destino. Em ambientes VANETs, esta abordagem leva a quebras constantes nos caminhos estabelecidos, devido ao alto número de saltos. Além disso, muitos pacotes são descartados, e o *overhead* gerado devido a notificações de perdas ou reparo de rotas aumenta significativamente.

O OLSR é um protocolo baseado em informação topológica que opera de maneira proativa, ou seja, cada nó periodicamente constrói e mantém um conjunto de vizinhos que possam ser alcançados em 1 ou 2 saltos. Além disso, apresenta uma extensão chamada

*MPR (Multi Point Relay)*, que procura minimizar o número de nós transmissores de forma que consigam alcançar vizinhos localizados a no máximo 2 saltos. Um nó encaminha um pacote somente se ele foi eleito como nó MPR pelo nó originador da mensagem. Para gerar e manter as tabelas de roteamento, o protocolo OLSR periodicamente transmite informações através do backbone de nós MPR.

O GPSR é um protocolo de roteamento baseado na localização dos nós, e adota um mecanismo de roteamento guloso para encontrar rotas entre pares de nós origem-destino. Porém, quando um determinado nó não consegue encontrar um nó vizinho mais próximo ao destino do que ele mesmo, o GPSR adota o algoritmo de roteamento em perímetro. Basicamente, a topologia da rede é transformada em um grafo planar e o encaminhamento dos dados é realizado utilizando a regra da mão direita. Porém, tal estratégia não se mostra ser estável em cenários urbanos, devido a presença de obstáculos neste tipo de cenário, o que gera muitos saltos entre os nós origem-destino, elevado atraso fim-a-fim e menor taxa de caminhos criados.

Nos trabalhos anteriores percebe-se que os mesmos não obtêm bons resultados conforme a densidade veicular é aumentada. Este trabalho se mostra diferente dos citados, pois além de não utilizar nenhum encaminhamento baseado na topologia (AODV e OLSR), no encaminhamento geográfico (GPSR e GeoSVR) ou em predição (MARDYMO) o mesmo diferencia de [CRUZ et al. 2012] e [Oliveira et al. 2011] pelo fato de propor um método mais eficiente de preenchimento das tabelas de roteamento, utilizando menor carga de sinalização, especialmente nos cenários com maior densidade veicular.

### 3. O protocolo XORBF

O protocolo de roteamento proposto, chamado de XORBF, é uma modificação do protocolo XOR proposto em [Oliveira et al. 2011]. O protocolo XORBF tem como principal característica em relação ao seu antecessor, o fato de incorporar os filtros de Bloom [TARKOMA et al. 2012] em sua estrutura.

O protocolo XOR usa identificadores planos de  $n$ -bits para organizar as tabelas de roteamento em  $n$  colunas e rotar pacotes pela rede. Seu princípio de roteamento usa como base a distância entre dois identificadores planos  $a$  e  $b$  calculada a partir da realização de uma operação lógica do tipo OU-Exclusivo(XOR), que é representada por  $d(a, b) = a \oplus b$ , sendo  $d(a, a) = 0$  e  $d(a, b) > 0, \forall a, b$ . Os identificadores dos nós podem ser gerados, por exemplo, aplicando funções *hash* no nome do nó, nos endereços IP/MAC ou usando uma chave privada. Considere um pacote originado pelo nó  $a$  e que o destino de tal pacote seja o nó  $c$ , e denotando  $\mathbb{Y}$  como o conjunto de identificadores contido na tabela de roteamento de  $a$ , o mecanismo de roteamento aplicado no nó  $a$  escolhe o nó  $b \in \mathbb{Y}$  que minimiza a distância em direção ao nó  $c$ , cuja política de roteamento é expressa conforme Equação 1.

$$\mathcal{R} = \underset{b \in \mathbb{Y}}{\operatorname{argmin}} \{d(b, c)\}. \quad (1)$$

Inicialmente, os nós da rede procuram outros nós vizinhos, transmitindo mensagens de *HELLO* em modo *broadcast* para aqueles nós que estão no seu raio de comunicação, de forma a adicioná-los nas  $n$  colunas, conhecidas como *buckets* e que são representadas pela condição  $\beta_i, 0 \leq i \leq n-1$ . Toda vez que um nó  $a$  obtém conhecimento

acerca de um nó  $b$ , aquele armazena a informação deste no *bucket*  $\beta_{n-1-i}$  correspondente, considerando que a variável  $i$  é o valor mais elevado que atende a Equação 2. Caso um nó ainda tenha *buckets* a serem preenchidos, o nó irá procurar ativamente por vizinhos que preencham tais *buckets*.

$$d(a, b) \text{ div } 12^i = 1, a \neq b, 0 \leq i \leq n - 1. \quad (2)$$

Para compreender melhor como é utilizada a Equação 2, vejamos um cenário que possua identificadores de 4 bits ( $n = 4$ ). Considere o nó  $a$  com o identificador 1011 e  $b$  com o identificador 0010. A distância entre eles ( $d(a, b)$ ) é obtida através da operação XOR entre os identificadores dos nós  $a(1011)$  e  $b(0010)$  cujo resultado é 1001. Em seguida, tal resultado é convertido para numeração decimal de onde é obtido o valor 9. Assim, a Equação 2 é resolvida, onde o maior valor de  $i$  que atende a equação é  $i=3$ , chegando a conclusão de que o identificador do nó  $b$  deve ser armazenado no *bucket*  $\beta_{n-1-i} = 0$ . É possível perceber que, basicamente, o número de bits iguais da esquerda para a direita, tanto do nó  $a$  (1011) quanto do nó  $b$  (0010) define em que *bucket* o identificador do nó  $b$  será armazenado na tabela de roteamento do nó  $a$ . No caso, o número de bits iguais entre os nós  $a$  e  $b$  é 0 (zero), e assim, o *bucket* escolhido é o  $\beta_0$ . É verificada essa relação entre os identificadores da Tabela 1, onde é possível ver a quantidade de bits semelhantes entre o nó 11 (1011) e os outros nós armazenados nos *buckets* de sua tabela de roteamento. Tais formas de construção das tabelas de roteamento apresentadas pelo protocolo XOR, fazem com que o protocolo seja menos complexo, tendo conhecimento de somente  $O(n)$  nós dos  $2^n$  nós presentes na rede para que o pacote seja roteado.

A seguir é apresentado o **processo de descoberta** (através de mensagens de *QUERY* e *RESPONSE*) onde o mesmo é iniciado com o objetivo de preencher os *buckets* das tabelas de roteamento dos nós.

A Figura 1 mostra o processo de construção das tabelas de roteamento do ponto de vista do nó  $a$ , assumindo que os nós  $b, c, d$  e  $e$  são suficientes para que o nó  $a$  atenda ao valor de  $N$  definido na Equação 3. O processo começa com o envio de uma mensagem de *QUERY* partindo do nó  $a$ , descrevendo quais *buckets* precisam de identificadores, para o nó  $b$  que já está armazenado em sua tabela de roteamento devido a mensagem de *HELLO*. Ao receber a mensagem de *QUERY*, o nó  $b$  envia de volta ao nó  $a$  uma mensagem de *RESPONSE* informando ao nó  $a$  da existência do nó  $c$ . Ao receber a mensagem de *RESPONSE*, o nó  $a$  armazena em sua tabela de roteamento o nó  $c$ . Como o nó  $a$  ainda precisa de informações para preencher a sua tabela de roteamento, ele envia uma mensagem de *QUERY* para o nó  $c$ , e assim, o nó  $c$  envia de volta para o nó  $a$  uma mensagem do tipo *RESPONSE* informando da existência dos nós  $d$  e  $e$ , preenchendo assim, todos os *buckets* necessários em sua tabela de roteamento, parando assim de enviar mensagens de *QUERY*.

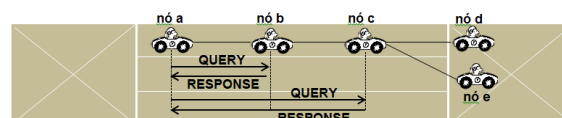


Figura 1. Exemplo da construção da tabela de roteamento.

<sup>1</sup>Operador aritmético que realiza uma divisão inteira, desconsiderando o resto

Em paralelo ao processo de descoberta, acontece o **processo de aprendizagem** que é responsável por obter informações sobre os outros nós existentes na rede, sem custo algum. Graças ao mecanismo de obter informações dos pacotes que atravessam os nós, é possível que tais nós obtenham informas sobre os outros nós. No cenário mostrado na Figura 1, o nó  $a$  inicialmente envia uma mensagem *QUERY* para o único vizinho que tem em sua tabela de roteamento, no caso  $b$ . O nó  $b$  envia uma mensagem *RESPONSE* para o nó  $a$  informando da existência do nó  $c$ . Quando o nó  $a$  envia a mensagem de *QUERY* para o nó  $c$ , o nó  $c$  aprende sobre a existência do nó  $a$ . Quando o nó  $c$  envia a mensagem *RESPONSE* para o nó  $a$ , tal mensagem passa também pelo nó  $b$  que, de forma passiva, acaba aprendendo sobre a existência dos nós  $d$  e  $e$ .

Sempre que há o recebimento de uma mensagem *QUERY* ou *RESPONSE*, está acontecendo o **processo de aprendizagem**. Todo nó que recebe tais mensagens verifica, dentro do corpo da mensagem, todos os identificadores ali contidos e insere-os em sua tabela de roteamento assim como os vizinhos físicos associados a tais identificadores. Assim, são disponibilizadas mais informações nas tabelas de roteamento com custo de aprendizagem praticamente zero, pois todo o processo acontece de forma passiva, e cada nó obtém mais informações para completar os *buckets* da tabela de roteamento dos nós.

Objetivando reduzir a quantidade de mensagens de sinalização geradas pela criação das tabelas de roteamento e buscar estabilidade da rede, é apresentado o conceito de **visibilidade local**. Tal conceito possui como característica dar prioridade em adicionar nas tabelas de roteamento dos nós os seus vizinhos físicos mais próximos. Por isso, em uma dada mensagem *QUERY* gerada por um nó  $a$  para um nó  $b$ , o nó  $b$  dará prioridade em inserir na sua mensagem *RESPONSE* não somente os nós que de fato atendam aos *buckets* ainda vazios do nó  $a$ , mas que também estejam mais próximos em número de saltos. Por exemplo, considere que um nó  $b$  recebe uma mensagem *QUERY* do nó  $a$  pedindo por identificadores que preencham o seu  $\beta_0$ . Considere que o nó  $b$  também informa acerca de dois nós que podem preencher o  $\beta_0$ . O primeiro nó é o nó  $c$ , localizado a 1 salto de distância do nó  $b$  e o segundo é o nó  $d$ , localizado a 2 saltos de distância. O conceito de **visibilidade local** faz com que o nó  $b$  gere uma mensagem *RESPONSE* com o nó que está mais próximo, em número de saltos, ou seja, nó  $c$  que está localizado a 1 salto de distância.

Além de priorizar a inserção de nós mais próximos, em número de saltos, nas mensagens de *RESPONSE*, o conceito de visibilidade local apresenta um mecanismo chamado *HOP-QUERY* ( $H$ ) que controla a quantidade de saltos da mensagem de *QUERY*, a partir do nó originador da *QUERY* evitando assim a inundação na rede, que gera maior consumo de banda e degrada o desempenho da rede. Na Figura 2 é mostrada uma situação onde o nó  $a$  utiliza  $H = 3$ . A dinamicidade da estrutura da rede VANET requer um mecanismo de controle de tempo dos identificadores contidos nas tabelas de roteamento. Assim, para cada entrada na tabela de roteamento existe um temporizador indicando o momento da sua inserção. Basicamente, vizinhos são removidos da tabela de roteamento depois de  $T_\beta$  segundos, e caso ocorram *buckets* vazios após a remoção, novas mensagens de *QUERY* são enviadas para preencher tais *buckets*.

Além disso, o mecanismo de roteamento do protocolo XOR considera um fator  $K$ , que define a quantidade de informação necessária para preenchimento dos *buckets* das tabelas de roteamento dos nós. Considerando que um *bucket*  $\beta_i$  pode possuir no máximo

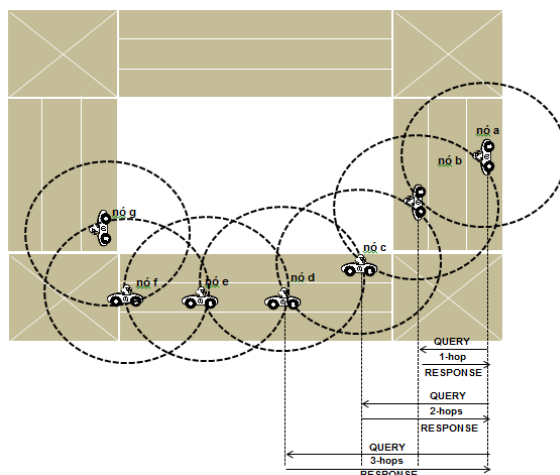


Figura 2. Mensagem de *QUERY* limitada a 3 saltos.

$2^{n-1-i}$  nós, se  $K > 2^{n-1-i}$ ,  $K$  é limitada para tal *bucket* segundo a equação  $K = 2^{n-1-i}$ . Então, ao ser alterado o valor de  $K$ , pode-se ter um maior ou menor conhecimento da rede. Assim, um número desejável de vizinhos deveria atender a Equação 3

$$N = \sum_{i=0}^{n-1} \min(K, 2^{n-1-i}). \quad (3)$$

Através da Equação 1, cada pacote pode ser encaminhado para o nó que, de fato, minimiza a distância lógica para o nó de destino. Considere o caso onde o identificador de cada nó é composto por 4 bits, assim tendo no máximo 16 nós na rede. Na Tabela 1 é apresentado um exemplo da tabela de roteamento do nó 11 (1011), que deseja encaminhar um pacote para o nó 5 (0101). Portanto, pela Equação 1 o nó 11 irá encaminhar o pacote para o nó 4 (0100) como próximo salto.

Tabela 1. Tabela de roteamento do nó 11 (1011).

$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$
0000	<b>1100</b>	<b>1000</b>	<b>1010</b>
0100	<b>1111</b>		
0111			
0010			

### 3.1. Os Filtros de Bloom

Os filtros de Bloom, apresentados por Burton Bloom em 1970, são estruturas probabilísticas que possuem como principal vantagem a economia de espaço, graças a sua estrutura compacta para armazenamento de dados, e permitem a consulta de elementos dentro dessa estrutura. Inicialmente, os filtros de Bloom eram utilizados em aplicações voltadas para banco de dados, porém, com a popularização da Internet, os filtros de Bloom ganharam maior visibilidade dos pesquisadores e da indústria desta área [TARKOMA et al. 2012].

Os filtros de Bloom caracterizam-se por resultar em falsos positivos na consulta de elementos inseridos nele, isto é, um dado elemento ser dito como presente em um conjunto, mas na verdade não se encontra inserido naquele conjunto. Além disso, os filtros de Bloom não resultam em falsos negativos, isto é, um dado elemento ser dito não estar presente em um conjunto de elementos, quando na verdade está presente.

Formalmente, o filtro de Bloom é um vetor de  $m$  bits que representa um conjunto  $S = \{x_1, x_2, \dots, x_p\}$  de  $p$  elementos, onde todos os bits de tal vetor são inicializados com o valor de zero. O tamanho  $m$  do vetor e o número  $h$  de funções *hash* são definidos levando em consideração a quantidade de elementos esperados a serem inseridos no filtro além da taxa de falso positivo a ser aceita. Para cada elemento  $x$  inserido dentro do filtro, os bits  $f_i(x)$  são configurados para bit com valor 1 para  $1 \leq i \leq h$ . O nível de eficiência na utilização dos filtros de Bloom dependerá de três características: (1) Tamanho do filtro, ou seja número de elementos do conjunto; (2) Número de funções *hash* utilizadas no filtro; (3) Número de elementos adicionados ao conjunto.

Porém, existe uma possibilidade de que falsos positivos aconteçam. A probabilidade de que isso ocorra é dada pela Equação 4:

$$P_{fp} = \left( 1 - \left( 1 - \frac{1}{m} \right)^{hn} \right)^h \quad (4)$$

Analisando a Equação 4, é visto que quanto menor o número de identificadores ( $n$ ) de um dado conjunto e quanto maior o tamanho do filtro  $m$ , menor a probabilidade de falsos positivos. Assim sendo, para  $n = 15$ ,  $m = 128$  e  $h = 3$ , a probabilidade de falsos positivos é igual a 0,0262. Portanto, mostra-se vantajoso utilizar os filtros de Bloom, devido a alguns benefícios tais como:

- Vários identificadores podem ser transmitidos simultaneamente sem aumentar o tamanho do filtro;
- Qualquer identificador, independente do seu tamanho, pode ser adicionado ao filtro de Bloom, permitindo assim uma maior flexibilidade.

Para obter redução na carga de sinalização é utilizada a técnica dos filtros de Bloom nas mensagens *QUERY* geradas pelo protocolo XOR. Basicamente, a ideia é introduzir nas mensagens de *QUERY* enviadas para os vizinhos o filtro de Bloom contendo os identificadores dos *buckets* que ainda não estão preenchidos reduzindo a quantidade de informação a ser enviada na *QUERY*.

Na Figura 3 é visto como ocorre a comunicação entre dois nós  $A$  e  $B$  com identificadores 1011 e 0101, respectivamente. Um filtro de Bloom de 8 bits, no qual cada bit representa uma posição iniciando da esquerda, com valor de 0 (zero) até o valor de 7, é utilizado para armazenar os identificadores do nó  $A$ . É considerado que três identificadores (0000, 0101 e 1100) são adicionados no filtro de Bloom utilizando duas funções *hash* ( $h1$ ) e ( $h2$ ), nas posições do filtro. Os bits correspondentes terão o valor de 1. O nó  $B$  recebe a mensagem *QUERY* com o filtro de Bloom e aplicando as mesmas funções *hash* ao identificador 1101, presente em sua tabela de roteamento, é retornado corretamente que tal identificador não está presente na mensagem de *QUERY*, logo inserindo tal identificador na mensagem de *RESPONSE*, caso o nó  $A$  o necessite. Porém, o filtro de Bloom



retorna que o identificador 0011 está presente, mesmo ele não estando de fato presente no filtro de Bloom, pois ao aplicar as funções *hash* no identificador 0011, retorna nas posições 0 e 5 os bits com valor 1 e que no filtro de Bloom tais posições também possuem o valor 1, ocorrendo em um falso-positivo.

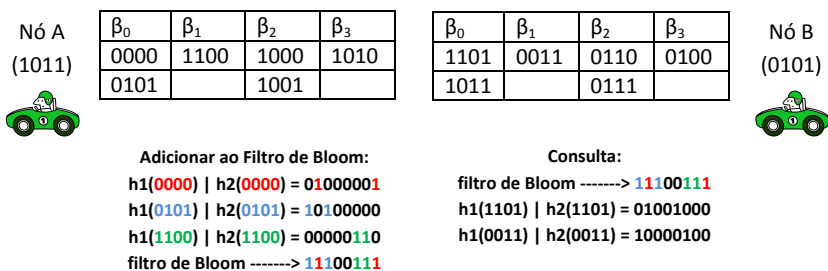


Figura 3. Exemplo da utilização do filtro de Bloom na mensagem QUERY.

Um recurso do protocolo XOR é o mecanismo de *HOP-QUERY (H)*. Quanto mais informação for gerada por um nó com o objetivo de preencher seus *buckets*, maior será a carga na rede. Para reduzir a geração de mensagens de *QUERY* na rede a técnica dos filtros de Bloom é adotada, de forma a inserir dentro desses filtros os IDs dos *buckets* que ainda não estão completamente preenchidos, reduzindo o tamanho da mensagem de *QUERY*.

#### 4. Avaliação de desempenho do protocolo proposto

A VANET simulada neste artigo foi obtida utilizando a ferramenta *TraNS* [TraNS 2013]. Para representar um cenário urbano, uma topologia retangular extraída da base TIGER foi utilizada, conforme Figura 4. O cenário possui as dimensões de 4.100 x 3.100 metros [CRUZ et al. 2012].

Nas simulações, os pontos de origem, destino e as rotas dos veículos foram definidos aleatoriamente para cada cenário avaliado. Além disso, com o objetivo de dar maior realismo de um cenário urbano, estão presentes na área da Figura 4, usado nas simulações, semáforos e várias faixas. Foram configurados 7 diferentes cenários de mobilidade, em que o número de veículos varia de 40 a 280 veículos, conforme detalhado na Tabela 2.

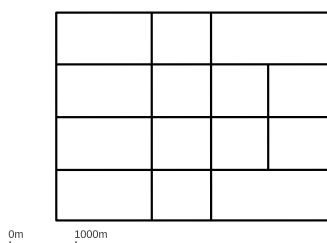


Figura 4. Cenário urbano utilizado na simulação.

Foi usado o simulador NS-2.33 [NS2 2013] configurado com o padrão IEEE 802.11<sup>2</sup>. A geração dos pacotes está dividida em duas fases. Na primeira fase, os caminhos foram criados de origens aleatórias para destinos aleatórios, a uma taxa de 3

<sup>2</sup>11 Mbps e 2Mbps foram usados para transmissão unicast e broadcast, respectivamente.

**Tabela 2. Cenários de mobilidade considerados na simulação.**

Parâmetros	Valores						
Nº Veículos	40	80	120	160	200	240	280
Velocidade máxima	20 m/s						
Raio de Alcance	400 metros						
Tempo de Simulação	437	424	561	598	556	566	614
Taxa de Falso Positivo	3,26-e5	2,79-e5	2,37-e5	5,82-e5	3,69-e5	3,19-e5	7,36-e5

caminhos por segundo e seguindo uma distribuição exponencial. Se o caminho é criado, pode-se dizer que ele está disponível. Na segunda fase, o nó de origem periodicamente envia pacotes pelo caminho disponível até o momento que ele seja quebrado.

De forma a obter resultados estatisticamente mais confiáveis, foram realizadas 30 rodadas de simulações para cada cenário e para cada protocolo usando diferentes sementes aleatórias. Os resultados mostrados possuem um nível de confiança de 95%. Os parâmetros utilizados pelo protocolo XOR e XORBF foram: o tempo em que um nó fica armazenado na tabela de roteamento ( $T_\beta$ ) = 5s; *QUERY-Timeout* ( $T_Q$ ) = 5s; e o fator  $K = 1$ . Esses valores foram definidos levando em consideração a dinamicidade da VANET urbana, o que demanda constante atualização das tabelas de roteamento dos nós.

Ainda para o protocolo XORBF foram utilizadas 3 funções *hash* e o tamanho do filtro foi definido em 128 bits. O foco das avaliações é demonstrar como a troca de mensagens de controle pode impactar no desempenho da rede. Dessa forma, os protocolos XOR e XORBF são comparados com diferentes valores do parâmetro  $H$  variando de 2 a 5. Porém, por restrições de espaço serão mostrados apenas valores de  $H = 2$  e  $H = 5$ . O valor de  $H = 1$  não é comparado por corresponder ao conjunto de vizinhos físicos o que não demonstraria a efetividade do mecanismo de preenchimento das tabelas de roteamento do protocolo proposto. As métricas de desempenho utilizadas foram:

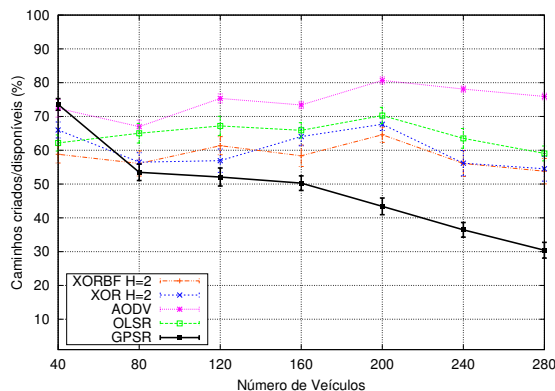
- **Caminhos criados/disponíveis** - Um caminho disponível é a situação em que um nó de origem consegue enviar um pacote de ping, responsável por originar o caminho, e este é recebido pelo nó de destino. Caso haja uma comunicação entre os dois nós (origem-destino), logo é considerado um caminho disponível/criado.
- **Duração do caminho** - Considerando um caminho  $z$  chamado de  $C = \{v_1, v_2, \dots, v_z\}$  em um instante  $t_1$ , a duração é considerado o maior intervalo de tempo  $[t_1, t_2]$  para a sequência  $C$ .
- **Comprimento do caminho** - Em redes ad hoc, o comprimento dos caminhos é representado pela quantidade de saltos que um pacote dá entre um nó de origem ( $v_1$ ) e o seu destino ( $v_z$ ). Assim, para um dado caminho  $C = \{v_1, v_2, \dots, v_z\}$ , consistindo em  $z$  nós, o comprimento dos caminhos corresponde  $z - 1$  saltos.
- **Atraso fim-a-fim** - Considerando um caminho criado com a sequência de nós chamado de  $C = \{v_1, v_2, \dots, v_z\}$  onde  $v_1$  é o nó de origem e  $v_z$  é o nó de destino, o atraso do pacote é a diferença entre o tempo final  $t_f$  (o pacote chega ao nó de destino  $v_z$ ) e o tempo inicial  $t_1$  (o pacote sai do nó de origem  $v_1$ ).

#### 4.1. Resultados obtidos

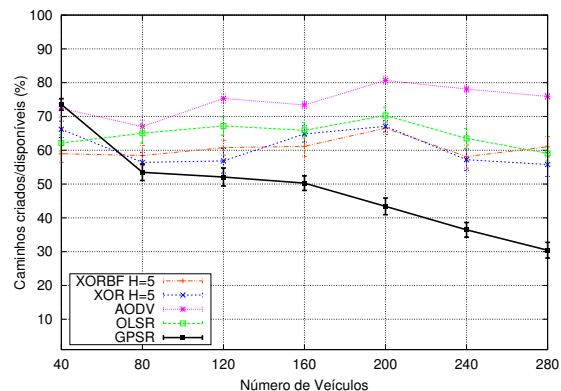
Nas figuras a seguir são apresentados os resultados obtidos pelo protocolo XORBF e XOR, com valores de *HOP-QUERY* ( $H = 2$ ) e ( $H = 5$ ) para as quatro métricas de

desempenho analisadas. Tais resultados são comparados com os dos protocolos AODV, OLSR e GPSR.

Nas Figuras 5 e 6, é visto que a implementação dos filtros de Bloom não melhora o desempenho do protocolo XORBF no que diz respeito ao percentual de caminhos criados/disponíveis nos cenários com menor densidade veicular e que com o valor de  $H = 2$  não impacta de forma significativa nos resultados. Porém, quanto maior o valor de  $H$ , maior a carga de sinalização gerada pelo protocolo XOR, em especial nos cenários com maior densidade de nós, onde a estrutura dos filtros de Bloom, mostra-se eficiente já que, ao conseguir compactar informações, a carga de sinalização também é reduzida alcançando resultados melhores que o protocolo XOR e o OLSR no cenário de 280 nós. O protocolo GPSR consegue os piores resultados. Um problema do GPSR é que em ambientes urbanos complexos, o encaminhamento guloso geralmente falha, e consequentemente, os pacotes entram em modo de perímetro que parece não ser efetivo para cenários urbanos, pois frequentemente se torna falho. Apesar de o protocolo AODV ter obtido os melhores resultados, é importante mencionar que esta métrica não é suficiente para caracterizar a qualidade dos caminhos disponíveis, dado que ele, para obtenção dos caminhos disponíveis, utiliza-se de *broadcast* de mensagens de controle, o que interfere nas outras métricas de performance mostradas a seguir.



**Figura 5. Taxa de caminhos criados/disponíveis pelos protocolos XOR e XORBF ( $H = 2$ ), AODV, OLSR, e GPSR**

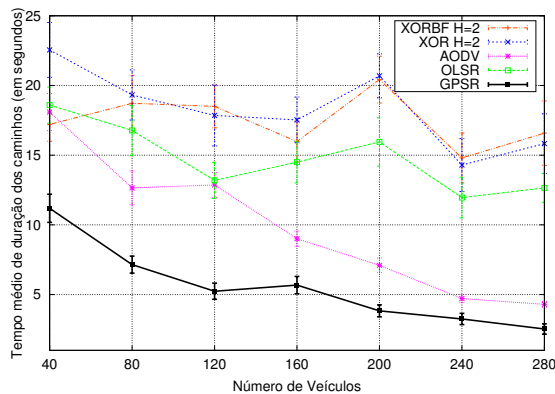


**Figura 6. Taxa de caminhos criados/disponíveis pelos protocolos XOR e XORBF ( $H = 5$ ), AODV, OLSR, e GPSR**

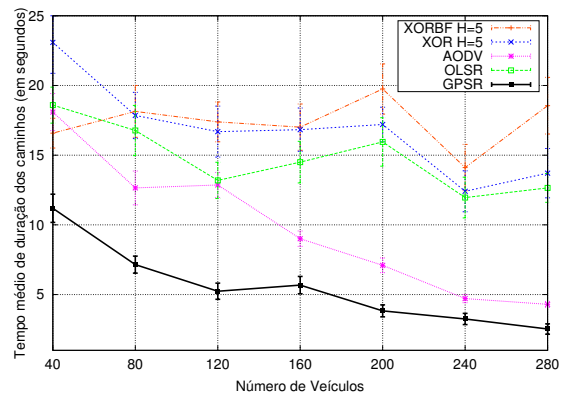
Nas Figuras 7 e 8 são apresentados os resultados da duração média dos caminhos disponíveis. É visto que com a variação de  $H = 2$  os resultados dos protocolos XOR e XORBF são semelhantes em todos os cenários. Porém, com  $H = 5$ , especificamente no cenário com maior densidade veicular (280 veículos) a técnica dos filtros de Bloom é eficiente no preenchimento dos *buckets*, e na redução das mensagens de controle, fazendo assim, com que o protocolo XORBF crie caminhos com maior tempo de duração.

Os protocolos AODV e GPSR possuem desempenhos insatisfatórios, mostrando que em cenários com maior densidade de nós os mesmos não se mostram como boas soluções de roteamento. O protocolo OLSR mesmo com a existência dos nós MPR, o que reduz a quantidade de mensagens de controle, não supera o protocolo XORBF nos cenários com 80 nós e superiores.

As Figuras 9 e 10 mostram o comprimento dos caminhos em número de saltos. Os



**Figura 7. Tempo médio de duração dos caminhos criados/disponíveis pelos protocolos XOR e XORBF (H = 2), AODV, OLSR, e GPSR**

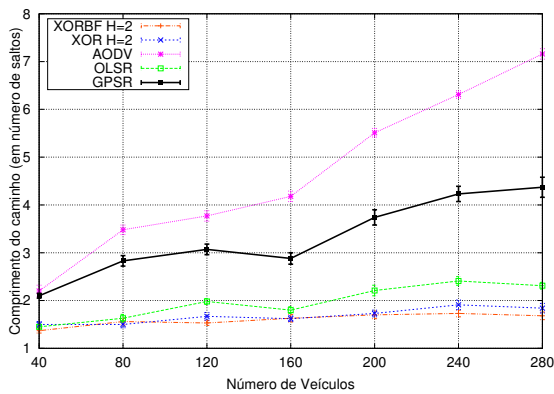


**Figura 8. Tempo médio de duração dos caminhos criados/disponíveis pelos protocolos XOR e XORBF (H = 5), AODV, OLSR, e GPSR**

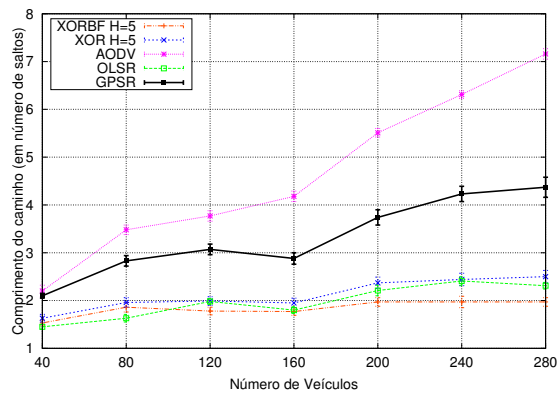
protocolos AODV e GPSR tiveram valores superiores em relação aos outros protocolos avaliados, levando a conclusão de que a probabilidade dos caminhos criados por eles sejam desfeitos é alta. O fato de inundar a rede com mensagens de sinalização em busca de nós para preenchimento e atualização das tabelas de roteamento faz com que o AODV tenha valores altos. O GPSR por entrar em situações de máximo local, conforme há o aumento da densidade de nós, maior é o número de saltos dados. Os caminhos criados por estes protocolos tendem a não serem caminhos de boa qualidade, pelo fato de que os caminhos terão curto tempo de duração conforme mostrado nas Figuras 7 e 8.

Acerca dos valores obtidos pelos protocolos XORBF e XOR, nota-se que os recursos incorporados nestes protocolos tais como a visibilidade local e o mecanismo de limitação de alcance da mensagem de *QUERY*, fazem com que o processo de encaminhamento das informações seja eficiente. Com a diminuição das mensagens de sinalização, por causa da utilização dos filtros de Bloom, o protocolo XORBF consegue superar o XOR e os demais em praticamente todos os cenários e nas variações de *H* mostradas. O protocolo XORBF, por ter baixo número de saltos, consegue criar caminhos de melhor qualidade, onde a probabilidade de quebra do enlace é pequena, e assim, os caminhos criados tendem a ter maior tempo de duração, conforme visto nas Figuras 7 e 8.

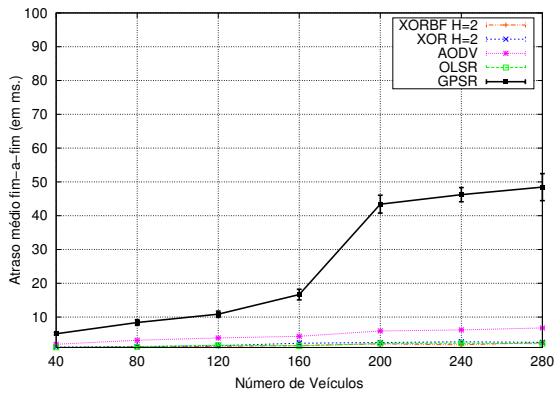
Nas Figuras 11 e 12 são comparados os valores de atraso obtidos na entrega dos pacotes dos protocolos XORBF e XOR com as variações de *H* mostradas em relação aos outros (OLSR, AODV, e GPSR). De forma geral, o comportamento do XOR mostra que, com o maior valor de *H* e maior a densidade veicular, os valores de atraso tendem a ser maiores, devido a elevada quantidade de mensagens de sinalização geradas com o objetivo de preencher os *buckets* dos nós. Em relação ao XORBF, a técnica dos filtros de Bloom é bastante eficiente na redução das mensagens de sinalização. Por exemplo, o protocolo XORBF quando comparado com o XOR no cenário de 200 nós com o valor de  $H = 5$ , alcança desempenho superior em 95%, pois o XOR, conforme é aumentado o número de nós, necessita enviar mensagens de *QUERIES* de forma a preencher os *buckets* mais elevados, cuja probabilidade de se encontrar os nós que preencham tais *buckets* é reduzida, o que gera maior tráfego de sinalização.



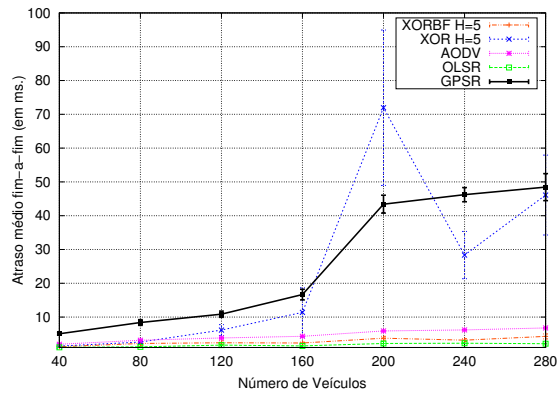
**Figura 9.** Comprimento dos caminhos criados/disponíveis pelos protocolos XOR e XORBF ( $H = 2$ ), AODV, OLSR, e GPSR



**Figura 10.** Comprimento dos caminhos criados/disponíveis pelos protocolos XOR e XORBF ( $H = 5$ ), AODV, OLSR, e GPSR



**Figura 11.** Atraso na entrega dos pacotes pelos protocolos XOR e XORBF ( $H = 2$ ), AODV, OLSR, e GPSR



**Figura 12.** Atraso na entrega dos pacotes pelos protocolos XOR e XORBF ( $H = 5$ ), AODV, OLSR, e GPSR

## 5. Conclusão

Os protocolos AODV, GPSR e OLSR, não possuem características que lidem com a constante mudança topológica das VANETs. Este trabalho apresentou o protocolo de roteamento XORBF, que incorpora os filtros de Bloom nas mensagens de *QUERIES* do protocolo XOR reduzindo as mensagens de sinalização e que visa ser escalável. Os resultados apresentados mostram que o XORBF, de fato, consegue reduzir a carga da rede reduzindo o atraso. Caminhos mais curtos tendem a aumentar a estabilidade da rede, devido a menor quebra de caminhos e caminhos mais estáveis é sinônimo de caminhos com maior tempo de duração. Os resultados mostram que o protocolo XORBF, de forma geral, supera os outros protocolos avaliados no cenário urbano avaliado. O protocolo XORBF consegue (com  $H = 5$ ), em cenários com alta densidade de nós, uma melhora nas métricas de tempo médio dos caminhos criados assim como no atraso e no comprimento dos caminhos, quando comparado especialmente com o XOR, usando o mesmo valor de  $H$ .

Como trabalho futuro, os autores explorarão novas técnicas para acelerar o processo da construção das tabelas de rotas, utilizando maiores valores de  $H$  e  $K$ .

## Referências

- Clausen, T. and Jacquet, P. (2003). Optimized Link State Routing Protocol (OLSR). *IETF RFC 3626*.
- Correia, S., Celestino, J., and Cherkaoui, O. (2011). Mobility-aware ant colony optimization routing for vehicular ad hoc networks. In *IEEE WCNC*, pages 1125–1130.
- CRUZ, E. P. F., CAMPOS, C. A. V., PASQUINI, R., Faina, L., and Oliveira, R. (2012). Performance Analysis of XOR-based routing in urban vehicular ad hoc networks. In: *IEEE WCNC*, pages 2521–2525.
- HAERRI, J., FILALI, F., and BONNET, C. (2006). Performance Comparison of AODV and OLSR in VANETs Urban Environments under Realistic Mobility Patterns. In: *5th IFIP Mediterranean Ad-Hoc Networking Workshop*.
- KARP, B. and KUNG, H. (2000). GPSR: greedy perimeter stateless routing for wireless networks. In: *ACM Mobicom*, pages 243–254.
- Lin, Y., Chen, Y., and Lee, S. (2010). Routing protocols in vehicular ad hoc networks: A survey and future perspectives. *Journal of Inf. Science Engineering*, vol. 26, no. 3, pp. 913-932.
- Martinez, J. A., Viguera, D., Ros, F. J., and Ruiz, P. M. (2013). Evaluation of the use of guard nodes for securing the routing in vanets. *Journal of Comm. and Networks*, 15(2):122–131.
- Miao, L., Djouani, K., van Wyk, B., and Hamam, Y. (2013). Performance evaluation of ieee 802.11p mac protocol in vanets safety applications. In *IEEE WCNC*, pages 1663–1668.
- Naumov, Valery, Gross, and R, T. (2007). Connectivity-Aware Routing (CAR) in Vehicular Ad-hoc Networks. In *INFOCOM*, volume 26, pages 1919–1927.
- NS2 (2013). Information Sciences Institute. NS-2 network simulator (version 2.33). [Online]. Disponível em: <http://nslam.isi.edu/nslam/index.php/Main Page>.
- Nzouonta, J., Rajgure, N., Wang, G., and Borcea, C. (2009). VANET routing on city roads using real-time vehicular traffic information. *IEEE Transactions on Vehicular Technology*, 58(7):3609–3626.
- Oliveira, R., Garrido, A., Pasquini, R., Liu, M., Bernardo, L., Dinis, R., and Pinto, P. (2011). Towards the use of xor-based routing protocols in vehicular ad hoc networks. In *IEEE 73rd VTC Spring*, pages 1–6.
- Perkins, C., Belding-Royer, E., and Das, S. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. *IETF RFC 3561*.
- TARKOMA, S., ROTHENBERG, C. E., and LAGERSPETZ, E. (2012). Theory and practice of Bloom Filters for distributed systems. *IEEE Communications Surveys and Tutorials*, Vol. 14, no. 2, pages 131–155.
- TraNS (2013). TraNS: open source tool for realistic simulations of VANET applications, [Online]. Disponível em: <http://trans.epfl.ch>.
- Xiang, Y., Liu, Z., Liu, R., Sun, W., and Wang, W. (2013). GeoSVR: A map-based stateless VANET routing. *Ad Hoc Networks*, pages 2125–2135.