

Sistema para Monitoramento Descentralizado de Trânsito Baseado em Redes Veiculares Infraestruturadas

José Geraldo Ribeiro Júnior^{1,2}, Igor M. Quintanilha¹, Miguel Elias M. Campista¹,
Luís Henrique M. K. Costa¹

¹Grupo de Teleinformática e Automação – Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – RJ – Brasil

²Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG
Leopoldina – MG – Brasil

{jgrjunior, quintanilha, miguel, luish}@gta.ufrj.br

Resumo. *As propostas atuais para o monitoramento automatizado de trânsito em vias públicas exigem um alto custo de instalação e manutenção, especialmente por serem dependentes de um elemento central responsável por calcular e divulgar as condições da via. Este artigo propõe um sistema para o monitoramento e divulgação das condições de trânsito de forma descentralizada, onde unidades de bordo e unidades de acostamento, que não precisam estar interligadas entre si, nem a um ponto central, trocam informações a fim de atualizar suas tabelas de condições sobre cada trecho da via. Para validar o sistema proposto, foram executados experimentos em um cenário montado na Universidade Federal do Rio de Janeiro utilizando uma rede IEEE 802.11b/g. A comparação dos resultados obtidos com os dados de um GPS mostrou alto grau de precisão tanto na detecção da posição dos veículos quanto na estimativa da condição da via.*

Abstract. *Current proposals of automated traffic monitoring systems in highways require high investments in installation and maintenance, especially because these systems depend on a central element, responsible to infer and disseminate the traffic conditions. This paper proposes a system to monitor and disseminate traffic conditions, using a decentralized structure. On board Units and Road Side Units do not need to be neither interconnected, nor connected to a central point. These elements exchange information to update their tables concerning traffic conditions of each route section. To validate the proposed system, a prototype was implemented using an IEEE 802.11b/g and experiments were performed on the Federal University of Rio de Janeiro. The comparison of the obtained results with data obtained from a GPS shows a high accuracy degree in detecting both the position of the vehicle and the estimative of the road condition.*

1. Introdução

Conhecer as condições de uma via em tempo real é fundamental para minimizar os problemas no trânsito. Apesar da maioria dos sistemas de monitoramento em todo o mundo utilizar câmeras de vídeos, propostas de sistemas automatizados começam

a se tornar uma realidade. Entre as propostas estão sistemas que usam: (1) sensores nas vias e nos carros [Edelmayer et al., 2010, Kassem et al., 2012], (2) sensores magnéticos [Cheung et al., 2004], (3) sensores acústicos [Fazenda et al., 2009], (4) técnicas de computação gráfica [Cucchiara et al., 2000], ou (5) a combinação de um GPS (*Global Positioning System*) [Yoon et al., 2007, Mohan et al., 2008], utilizado para definir a posição do veículo na via, com 3G [Google, 2011, Valerio et al., 2009] ou redes IEEE 802.11 [Thiagarajan et al., 2009], para enviar os dados sobre a movimentação do veículo. Outros sistemas dispensam o uso do GPS para localização do veículo [Ribeiro Júnior et al., 2012a], o que reduz o consumo de bateria, especialmente em equipamentos como *smartphones*. No entanto, esses sistemas dependem de um elemento centralizador, responsável por inferir e divulgar as condições na via. Essa central pode ser um elemento externo, conectado via Internet, ou fazer parte da rede local.

Em paralelo ao aumento constante do número de congestionamentos, está a popularização do uso de *smartphones*. Segundo o relatório da Cisco sobre previsão do crescimento do tráfego móvel global, até 2015 haverá um *smartphone* por habitante no planeta [Cisco, 2012]. Por outro lado, a alta demanda da tecnologia 3G/4G está deteriorando a qualidade e a garantia do serviço prestado aos usuários devido a congestionamentos na rede [Balasubramanian et al., 2010], fazendo as operadoras limitarem os planos a uma taxa máxima de uso.

Este trabalho propõe um sistema colaborativo para o monitoramento e divulgação das condições de trânsito de forma descentralizada. Nessa proposta, unidades de bordo e unidades de acostamento trocam informações a fim de atualizar suas TCTs (Tabelas de Condições dos Trechos), que contêm informações sobre cada trecho da via. Um diferencial desta proposta é o fato das unidades de acostamento não precisarem estar conectadas, nem entre si e nem tampouco a um ponto central. Utilizando kits de acostamento, já propostos na literatura [Ribeiro Júnior et al., 2011], o sistema pode ser utilizado mesmo em rodovias onde não existe infraestrutura, como energia ou cobertura celular.

No sistema proposto, as informações geradas pelos veículos são ponderadas para privilegiar as informações mais recentes e ainda para auxiliar na definição de um tempo de vida útil para a informação, uma vez que o objetivo é inferir a condição da via em tempo real. Por não haver um elemento centralizador, não há garantia de sincronismo entre os relógios, dessa forma foi necessário utilizar um mecanismo de controle semelhante ao controle de empréstimo (*lease*) do DHCP (*Dynamic Host Configuration Protocol*) [rfc, 1997], onde o tempo de vida útil é dado em segundos e cada dispositivo decrementa o tempo, baseado no relógio local, até chegar a zero.

Para validar o sistema proposto, um protótipo foi implementado no campus da Universidade Federal do Rio de Janeiro, na Ilha do Fundão, utilizando uma rede IEEE 802.11b/g. O cenário escolhido possui características semelhantes a vias públicas com grande fluxo de veículos, além de outras redes na mesma frequência. Foram coletados dados sobre cada etapa do processo utilizando um veículo. Entre as medições estão o tempo de associação e configuração de rede, o tempo para envio e atualização da tabela de condição do trecho e a detecção de passagem da unidade de bordo pela unidade de acostamento. A comparação dos resultados obtidos com os dados de um GPS de alta precisão (informa a posição 4 vezes por segundo) mostrou um alto grau de precisão tanto na detecção da posição dos veículos quanto na estimativa da condição da via.

O restante deste artigo possui a seguinte estrutura: a Seção 2 apresenta detalhes do sistema proposto, enquanto a Seção 3 apresenta as especificações de um protótipo implementado para testes utilizando redes IEEE 802.11b/g. A Seção 4 apresenta os experimentos e os resultados obtidos no campus da UFRJ. Finalmente, a Seção 5 apresenta as conclusões e indica os trabalhos futuros.

2. Sistema Proposto

A Figura 1 apresenta a arquitetura do sistema proposto, onde as unidades de acostamento estão instaladas em pontos de ônibus. Considera-se que um usuário do veículo possui um equipamento com a aplicação proposta em execução, a via possui trânsito nos dois sentidos e as distâncias entre as unidades de acostamento sejam previamente conhecidas. A conexão das unidades de acostamento a elementos externos à rede local permite a divulgação das informações da via para dispositivos externos.

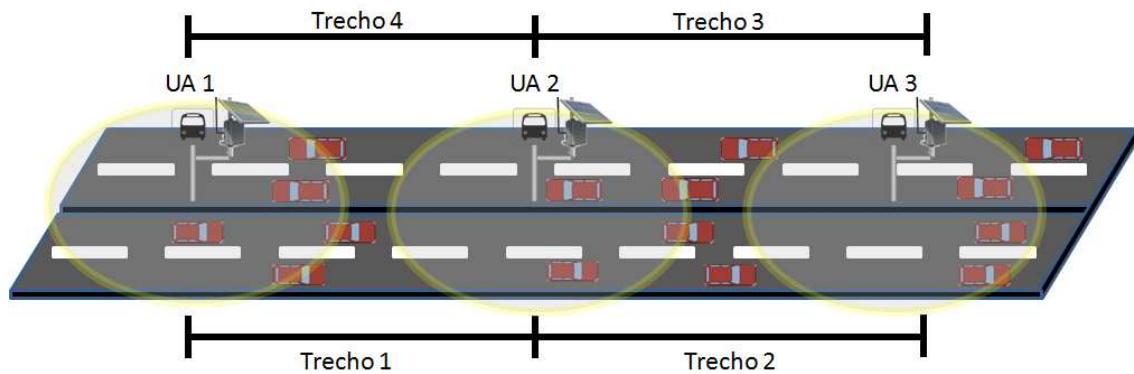


Figura 1. Arquitetura do sistema proposto.

Conhecida a distância entre as unidades de acostamento e o tempo utilizado para ir de um ponto ao outro, é possível calcular a velocidade do veículo no trecho. Considera-se um trecho o segmento da via que fica entre as unidades de acostamento (Figura 1). O número de trechos é proporcional à quantidade de unidades de acostamento. Para cada sentido da via é considerado um trecho diferente. O número de trechos (N_T) é dado pela equação $N_T = ((N_{UnA} - 1) * N_D)$, onde N_{UnA} , e N_D são, respectivamente, o número de unidades de acostamento e o número de direções, que nesta proposta será sempre múltiplo de 2.

A Tabela de Condição dos Trechos (TCT), vista na Figura 2, é usada para a troca de informação entre a unidade de bordo e a unidade de acostamento. Nessa tabela estão as seguintes informações: Trecho, Condição e TTL (*Time To Live*). O Trecho representa o identificador único do trecho em questão, que será utilizado na comparação entre as tabelas; a Condição representa a velocidade média atual no trecho; e o TTL representa o tempo de vida de cada entrada da tabela. O TTL tem basicamente duas funções: definir um tempo de vida útil para a informação e atribuir maior peso para informações mais recentes. Uma vez que a ideia é inferir a condição atual nos trechos, as informações defasadas recebem menor ou nenhum peso (no caso do TTL ser zero).

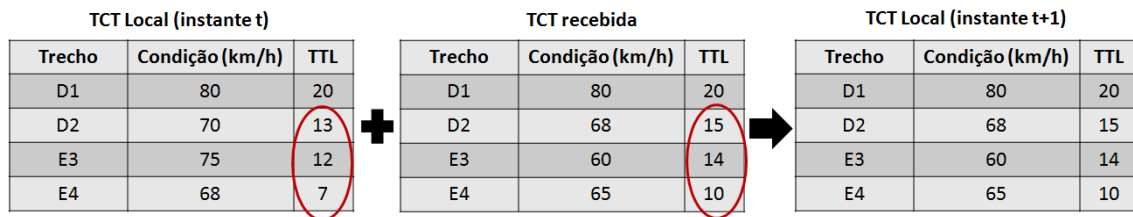


Figura 2. Atualização da TCT local.

2.1. Funções da Unidade de Bordo

A unidade de bordo é responsável por quatro funções: (1) calcular em tempo de execução a velocidade dentro do trecho, (2) atualizar a TCT local, (3) detectar o melhor momento pra envio da TCT à unidade de acostamento e (4) enviar a TCT a unidade de acostamento mais próxima. Ao enviar essas informações, o veículo atualiza a TCT da unidade de acostamento em, pelo menos, todas as linhas referentes aos trechos anteriores já que os valores do TTL são maiores. Analisando a Figura 1, por exemplo, quando um veículo acaba de passar pelo trecho 1, ele envia informações sobre as condições do trânsito para a unidade de acostamento 2, onde ao menos as informações sobre os trechos 1 e 4 serão mais atuais. Como não há conexão entre as unidades de acostamento, os veículos cumprem o papel dos enlaces de comunicação da arquitetura proposta, possibilitando que cada unidade de acostamento tenha informação sobre toda a via.

Como é apresentado no Algoritmo 1, ao receber a TCT, o veículo compara cada linha com a TCT local. Cada linha representa um trecho da via. As informações recebidas sobre trechos com TTL maior que as existentes na TCT atual são usadas para atualizar a TCT local. A Figura 2 ilustra a comparação, onde a tabela local, no instante $t + 1$, é atualizada após a comparação das tabelas no instante t . Ao detectar o momento em que o veículo passou pela unidade de acostamento, este calcula a velocidade no trecho. Em seguida a TCT local é atualizada e finalmente enviada para a unidade de acostamento.

2.2. Função da Unidade de Acostamento

Como mostra o Algoritmo 2, cada unidade de acostamento divulga sua TCT atual ao cliente após terminar o processo de conexão. Assim, cada unidade de acostamento divulga para os veículos próximos, em ambos os sentidos, uma visão das condições em trechos futuros. Quando a unidade de acostamento recebe uma TCT, ela também precisa comparar com a informação local e, se o TTL de cada entrada for maior do que a informação atual, atualizar seus dados. No entanto, como são vários veículos enviando informação simultaneamente, e os veículos podem apresentar velocidades distintas, de acordo com a condição de cada pista, a unidade de acostamento utiliza a média harmônica simples para calcular a condição atual do trecho. Utilizando a Equação 1, o sistema atribui um peso maior para as informações mais recentes.

$$MHS_{atual_{Trecho}} = \frac{2}{\frac{1}{v_i} + \frac{1}{MHS_{ant}}}, \quad (1)$$

onde v_i é a velocidade mais recente recebida do último nó móvel.

Algoritmo 1 Pseudo-código executado pela unidade de bordo.

```

...
 $N_{Trechos} = ((N_{UnA} - 1) * N_D);$ 
while true do
  if Entrou na área de cobertura da unidade de acostamento then
    Faz pedido de conexão;
  else
    Procura ESSID conhecido;
  end if
  Recebe TCT da unidade de acostamento;
  while  $num_{Linha} < N_{Trechos}$  do //Compara TCT recebida com TCT local
    if  $TTL_{atual_{LinhaX}} \leq TTL_{recebida_{LinhaX}}$  then
       $TTL_{atual_{LinhaX}} \leftarrow TTL_{recebida_{LinhaX}};$ 
       $Condicao_{Atual_{LinhaX}} \leftarrow Condicao_{Recebida_{LinhaX}};$ 
    end if
  end while
  Detecta momento em que passou pela unidade de acostamento;
  Guarda o tempo em que passou pela unidade de acostamento;
   $Veloc_{Trecho} \leftarrow \frac{dist_{Trecho}}{Tempo_{UnA2} - Tempo_{UnA1}};$ 
  Atualiza TCT local;
  Envia a TCT atualizada para a unidade de acostamento;
end while
...

```

Algoritmo 2 Pseudo-código executado pela unidade de acostamento.

```

...
 $N_{Trechos} = ((N_{UnA} - 1) * N_D);$ 
while true do
  Recebe pedido de conexão;
  Envia, junto com as configurações de rede, a TCT local atual;
  Aguarda por TCT atualizada;
  Recebe TCT atualizada;
   $num_{Linha} \leftarrow 1;$ 
  while  $num_{Linha} < N_{Trechos}$  do //Compara TCT recebida com TCT local;
    if  $TTL_{Atual_{LinhaX}} = 0$  then
       $TTL_{Atual_{LinhaX}} \leftarrow TTL_{Recebida_{LinhaX}};$ 
       $Condicao_{Atual_{LinhaX}} \leftarrow Condicao_{Recebida_{LinhaX}};$ 
    else
      if  $TTL_{Atual_{LinhaX}} \leq TTL_{Recebida_{LinhaX}}$  then
         $MHS_{Atual_{Trecho}} = \frac{2}{\frac{1}{Condicao_{Recebida}} + \frac{1}{MHS_{ant}}};$ 
         $TTL_{Atual_{LinhaX}} \leftarrow TTL_{Recebida_{LinhaX}};$ 
         $Condicao_{Atual_{LinhaX}} \leftarrow MHS_{Atual_{Trecho}};$ 
      end if
    end if
  end while
   $num_{Linha} \leftarrow num_{Linha} + 1;$ 
end while
...

```

2.3. Cálculo do TTL

O sistema proposto não assume o sincronismo entre os relógios das unidades de bordo e de acostamento, de complexa realização uma vez que o cenário é parcialmente desconectado. Como solução mais simples, definiu-se que cada dispositivo fica responsável por decrementar o valor do TTL baseado no horário local. Dessa forma, o valor do TTL é decrementado a cada segundo. A falta de sincronismo é minimizada, ficando restrita apenas ao tempo de envio entre um elemento e outro.

O valor máximo do TTL depende de características da via, como número de pistas, extensão, velocidade máxima permitida, entre outras. Um tempo de TTL muito pequeno pode não ser suficiente para atualizar as condições da via em trechos distantes ou em vias onde o limite de velocidade é baixo. Um tempo de TTL muito grande pode fazer com que as unidades de acostamento tenham uma informação desatualizada sobre trechos distantes. O valor do TTL deve ser maior que tempo de defasagem, ou seja, maior que o tempo necessário para atualizar as informações sobre o trecho mais distante. O cálculo para encontrar o valor do tempo de defasagem (T_{Def}) é dado pela equação $T_{Def} = \sum_{i=1}^n T_{Trecho}$, onde n é o número de trechos e T_{Trecho} é o tempo médio para atravessar cada trecho.

3. Protótipo do Sistema Proposto Baseado em uma Rede IEEE 802.11

Um protótipo do sistema foi implementado utilizando uma rede IEEE 802.11b/g. As rotinas necessárias foram implementadas em *Python*. As seções a seguir apresentam detalhes da implementação da unidade de bordo e da unidade de acostamento.

3.1. Requisitos Básicos - Estrutura

O uso de rotinas específicas para a troca de TCT nas unidades de acostamento necessitou equipamentos que permitam a alteração do *firmware* padrão. Cada unidade de acostamento foi composta por um kit com autonomia de energia, ilustrado na Figura 3. Esse equipamento foi baseado no kit usado pelo projeto ReBUS [Ribeiro Júnior et al., 2011]. Cada kit usado no protótipo proposto é composto por um roteador D-Link modelo DIR-320, um pendrive USB 2.0 de 32 GB, um circuito regulador de tensão e uma bateria de 12V/7 Ah. A posição de cada unidade de acostamento era previamente conhecida. Foram utilizadas as antenas padrão que vem com os equipamentos.



Figura 3. Kit de acostamento.

Para determinar a distância entre dois pontos de acesso, foi utilizada a função de Haversine [Nordin et al., 2012]. Essa função é bastante usada em sistemas de navegação

e é capaz de fornecer a distância entre dois pontos de uma esfera utilizando coordenadas geográficas (latitude e longitude). Realizando uma aproximação da Terra como uma esfera perfeita, é possível apresentar um erro médio de 0.3% nos cálculos. A função de Haversine ($haversine(\theta)$) é definida como:

$$haversine(\theta) \equiv \sin^2\left(\frac{\theta}{2}\right) \quad (2)$$

Considerando dois pontos de uma esfera de raio R , com latitudes e longitudes (ϕ_1, λ_1) , (ϕ_2, λ_2) , respectivamente, a distância d é:

$$d = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cos(\phi_2) \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right). \quad (3)$$

Ao utilizar um GPS com uma amostragem σ , a distância total D percorrida por um veículo no intervalo de tempo $[0, k * \sigma]$ é igual a $\sum_{n=0}^{k-1} d_{n,n+1}$, onde $d_{n,n+1}$ representa o cálculo da n -ésima distância utilizando a fórmula de Haversine para amostras espaçadas no tempo de σ .

3.2. Etapas Executadas no Nó Móvel

A execução do sistema na unidade de bordo do protótipo requer pelo menos seis passos: (1) varredura pelos ESSID's conhecidos, associação e conexão com o ponto de acesso; (2) recebimento da TCT pelo veículo; (3) comparação e atualização da TCT recebida com a TCT local (no veículo); (4) detecção do momento em que o veículo passou pelo ponto de acesso, terminando o trecho; (5) cálculo da velocidade no trecho; e (6) envio da TCT do veículo para o ponto de acesso. É fundamental que o veículo envie a velocidade média calculada no trecho para o ponto de acesso antes que perca o sinal. A Figura 4 apresenta um diagrama com a execução dos passos para conexão e troca de informação entre veículo e ponto de acesso. Para que possa receber os quadros de sondas provenientes dos pontos de acesso, enquanto não estiver conectada, a interface de rede do cliente deve estar em modo monitor.

Como é possível observar na Figura 4, o cliente processa todos os quadros *beacon* recebidos a fim de determinar se o ESSID faz parte do sistema de monitoramento. Essa verificação é feita em arquivo XML local (Listagem 1). Para identificar em qual trecho o veículo se encontra ou definir a direção na via, usa-se a combinação dos dois últimos pontos de acesso, anterior e atual. Utilizando os momentos em que passou por cada ponto de acesso e a extensão do trecho, o veículo calcula a velocidade média. No código a seguir, é possível visualizar que trecho é composto por dois pontos de acesso e pela extensão. O arquivo armazena ainda as informações sobre cada ponto de acesso, que são: o ESSID, ou seja, o nome da rede; o endereço MAC e as coordenadas geográficas.

Listagem 1. Arquivo para definição do Trecho.

```

1 <root>
2   <path id='ID'>
3     <stretch>trecho</stretch>

```

```

4   <distance>extensao</distance>
5   <encodedPath>encodedPath</encodedPath>
6   <from>
7     <ssid>ssid</ssid>
8     <macAddress>00:00:00:00:00:00</macAddress>
9     <location>
10      <latitude>latitude1</latitude>
11      <longitude>longitude1</longitude>
12    </location>
13  </from>
14  <to>
15    <ssid>ssid</ssid>
16    <macAddress>ff:ff:ff:ff:ff:ff</macAddress>
17    <location>
18      <latitude>latitude2</latitude>
19      <longitude>longitude2</longitude>
20    </location>
21  </to>
22 </path>
23 </root>

```

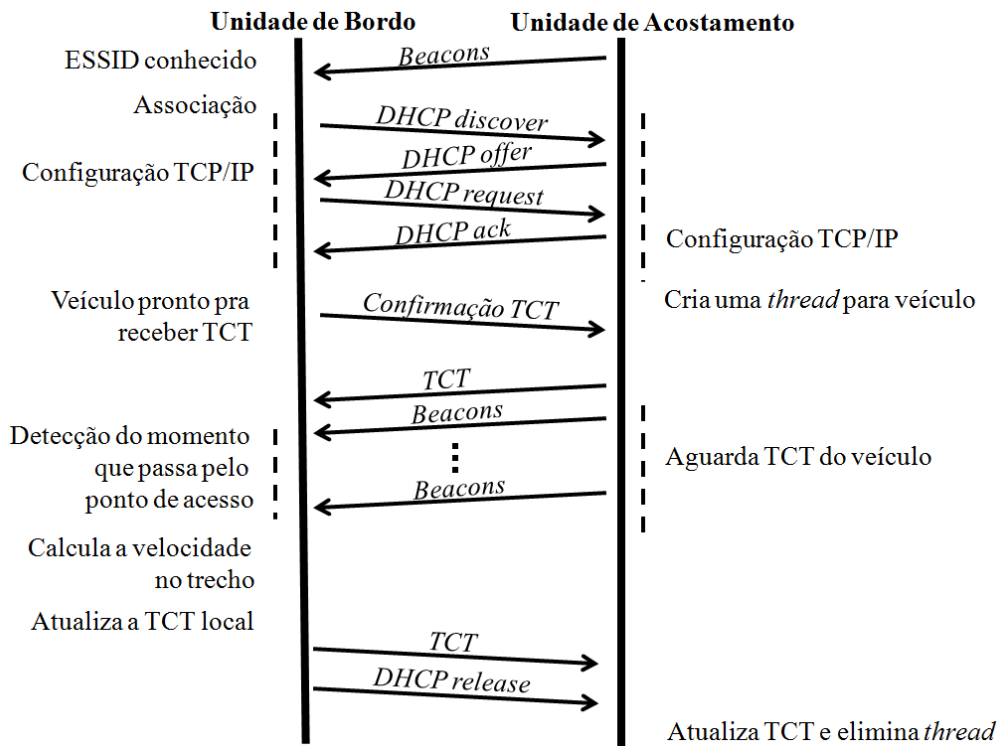


Figura 4. Diagrama das etapas de funcionamento do sistema proposto.

Uma vez que o ESSID destacado é conhecido, o veículo se associa ao ponto de acesso e aguarda o recebimento das informações de trânsito (via TCT) enviadas pelo ponto de acesso. Vale mencionar que a interface do cliente é retirada do modo monitor logo depois da associação com o ponto de acesso. Assim como demonstrado na Seção 2.1, o veículo atualiza a TCT local, levando em consideração o TTL em cada trecho. Após esse processo, o cliente volta a analisar os *beacons* a fim de determinar o

momento em que se encontra mais próximo do ponto de acesso. O algoritmo proposto em [Ribeiro Júnior et al., 2012b] considera tal momento como sendo aquele em que se recebe o *beacon* com maior potência de sinal. O veículo detecta que passou pelo ponto de acesso, finalizando o percurso do trecho, quando recebe um *beacon* com potência de sinal 10 dBm maior que a maior potência recebida do mesmo ponto de acesso. Após passar pelo ponto de acesso, o veículo calcula a velocidade no trecho e atualiza a TCT com a nova Condição e TTL. A tabela resultante é enviada para o ponto de acesso. Por fim, há a desassociação com o ponto de acesso e a procura pelo próximo ponto na trajetória.

Após o processo de desassociação, para evitar a reassociação, já que ele ainda pode receber *beacons* do ponto de acesso anterior, o endereço MAC passa a identificar o ponto de acesso anterior. Assim, o veículo sempre armazena o endereço MAC do ponto de acesso anterior e o atual (ou o próximo) para que possa definir o trecho.

3.3. Unidade de Acostamento

Na unidade de acostamento (ponto de acesso), há a necessidade de instalar um *firmware* de código aberto como o OpenWRT [Openwrt, 2011] para o funcionamento do algoritmo. Como se pode observar no diagrama da Figura 4, a unidade espera pelo momento em que um veículo se associa. Em seguida, o ponto de acesso espera uma confirmação de que o cliente está pronto para receber a TCT. Ao receber esta confirmação, o ponto de acesso envia a TCT. Para cada novo cliente conectado é instanciada uma nova *thread* para tratar a etapa de troca de informações separadamente. Uma vez enviada a TCT, a *thread* aguarda o envio da TCT da unidade de bordo. Caso exista mais de uma conexão simultânea, a unidade de acostamento deve sincronizar os processos, para que, ao atualizar a TCT local, considere todas as condições recebidas do último. Para isso, é realizado o cálculo da média harmônica simples. Nesta proposta não há divisão dos veículos em categorias uma vez que, como demonstrado em [Ribeiro Júnior et al., 2012a], utilizando a média harmônica para inferir a velocidade no trecho, a interferência de veículos preferenciais na condição do trecho só acontece em cenários muito específicos. Por outro lado, como demonstrado em [Treiber e Kesting, 2010], se houver veículos mais lentos que o fluxo normal, eles afetarão diretamente o tráfego nas outras pistas.

Para o protótipo utilizou-se uma rede aberta, no entanto o uso de chaves de criptografia não impede o funcionamento do sistema proposto.

4. Experimentos

Para realizar os experimentos, dois cenários foram montados no campus da Ilha do Fundão, na Universidade Federal do Rio de Janeiro. As unidades de acostamento foram compostas pelos kits apresentados na Figura 3. A aplicação cliente foi executada em um *laptop* Sony Vaio com processador I5-3210m, 6 GB de RAM e disco rígido de 640 GB e interface de rede sem fio. Para comparação dos resultados utilizou-se um GPS modelo "u-blox 5", que informa a posição quatro vezes por segundo.

4.1. Primeiro Cenário

No primeiro cenário, foram utilizados uma unidade de acostamento e um dispositivo móvel para verificar o tempo gasto na associação ao ponto de acesso e no envio das configurações de rede. Foi avaliada também a eficiência da proposta descrita na Seção 3.2

para detecção do momento em que o veículo passa pelo ponto de acesso podendo então desconectar. O trecho apresentado na Figura 5 possui 900 m de extensão. Uma limitação desse cenário é o limite de velocidade permitido nas vias, de 40 km/h.



Figura 5. Cenário - Experimento 1 - UFRJ.

A Tabela 1 apresenta os resultados do tempo de conexão, definido como o intervalo de tempo em que o veículo recebe o primeiro *beacon* do ESSID até o momento em que ele recebe as configurações de rede. As limitações do padrão IEEE 802.11b/g no que diz respeito a mobilidade e a atenuação por múltiplos caminhos aumentam a perda de quadros com o aumento da velocidade do veículo. Dessa forma, quanto mais rápido o veículo estiver, mais tempo ele leva para estabelecer a conexão. Estando mais lento, os efeitos da atenuação são menores, permitindo que a conexão seja concluída mais rapidamente. Durante os experimentos, foram detectadas cerca de 11 redes IEEE 802.11b/g utilizando os canais 9 e 11 (o canal 9 era utilizado pela unidade de acostamento).

O veículo sempre desconecta logo após passar pelo ponto de acesso, ao receber um sinal com potência 10 dBm menor que a máxima.

Tabela 1. Tempo de conexão entre unidade de bordo e unidade de acostamento.

Tempo de Conexão	
Velocidades	Tempo
20 km/h	< 1 seg
25 km/h	4 seg
35 km/h	7 seg
40 km/h	9 seg

4.2. Segundo Cenário

No segundo cenário, o experimento conta com duas unidades de acostamento e uma unidade de bordo. A Figura 6 apresenta os dois trechos utilizados. Para calcular a distância

entre os dois pontos de acesso, utilizando a fórmula de Haversine (Equação 2), a taxa de amostragem utilizada foi de 1 segundo. Na unidade de bordo mediu-se o tempo de associação, o tempo para receber as configurações TCP/IP, o tempo para executar as trocas de tabelas e o momento de desconexão.

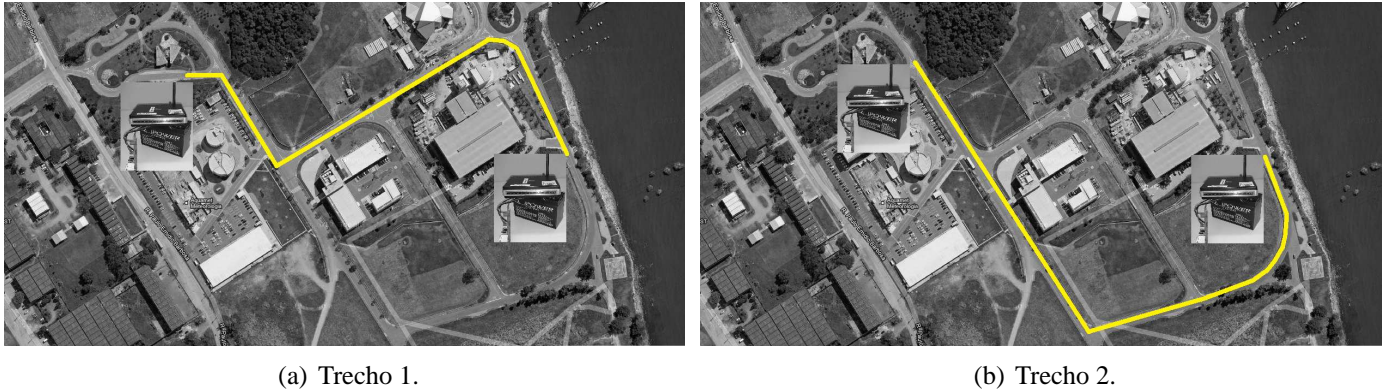


Figura 6. Experimento 2 - UFRJ.

A Figura 7 apresenta quatro momentos em que o veículo passa pelos pontos de acesso, executando todo o algoritmo. As linhas em destaque apresentam os momentos em que (1) o veículo encontrou o ponto de acesso, isto é, o momento em que o veículo recebeu o primeiro *beacon* daquele ESSID, (2) o momento em que o veículo terminou o processo de associação ao ponto de acesso e (3) o momento da desconexão. Os asteriscos representam os *beacons* recebidos. Como pode ser visto, a desconexão acontece logo após a unidade de bordo receber o sinal com maior potência, ou seja, quando a potência do sinal recebido é 10 dBm menor.

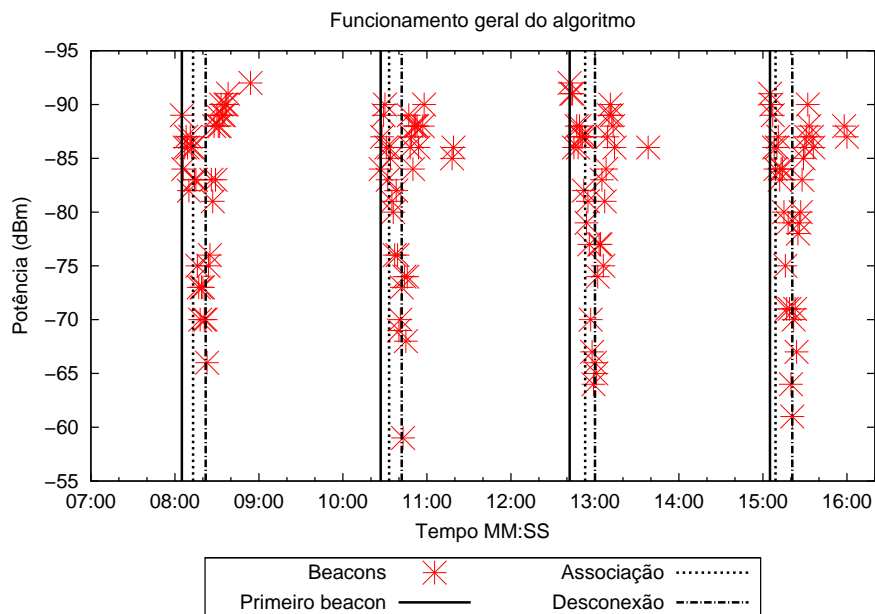


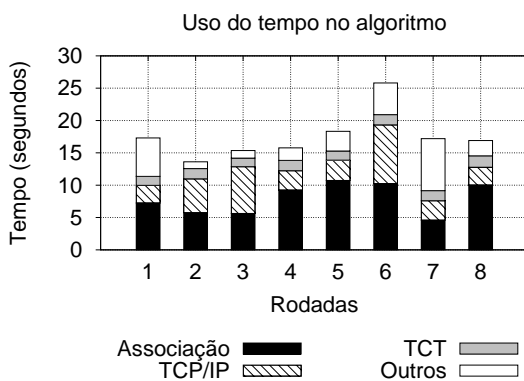
Figura 7. Funcionamento do algoritmo.

A Figura 8(a) detalha o processo para que seja possível visualizar o tempo gasto

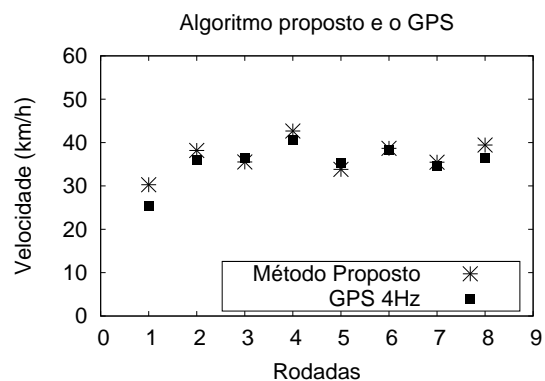
em cada etapa. O intervalo de tempo entre o momento em que o veículo detectou um ESSID conhecido e se associou variou entre 6 e 11 segundos. Foram necessários de 3 a 9 segundos para receber as configurações de rede e enviar o aviso para receber a tabela. O tempo médio para recebimento da tabela com as condições dos trechos foi de 2 segundos. O veículo detectou que passou pelo ponto de acesso com precisão de 3 metros, no pior caso.

Como é possível observar, fazer a associação e receber as configurações de rede é a etapa que necessita de mais tempo. Ao menos duas opções podem reduzir o tempo de execução do algoritmo ao utilizar IEEE 802.11b/g. A primeira seria que o veículo utilizasse um endereço IP pré-definido ao detectar o ESSID conhecido. Outra opção seria tornar o primeiro ponto de acesso como o único responsável por distribuir os endereços IP, fazendo com que essa etapa aconteça somente no momento em que o veículo entra na via. Experimentos utilizando IP fixo na unidade de bordo reduziram o tempo necessário para receber as configurações de rede de 3 a 9 segundos para menos de 1 segundo.

A Figura 8(b) apresenta a condição inferida em cada trecho, tanto pelo método proposto quanto pelo GPS. Para gerar uma condição única utilizando o GPS, foi calculada a média harmônica de todos os valores obtidos no trecho. Como pode ser visto, os resultados são muito semelhantes. Considera-se que a condição inferida está correta quando os resultados estão dentro do mesmo intervalo que define as condições do trecho. Assim como Ribeiro Júnior et. al [Ribeiro Júnior et al., 2012b], foram definidos três intervalos. Caso a velocidade média esteja abaixo de 40 km/h, significa que o trecho apresenta condição lenta. Entre 40 e 80 km/h o trecho apresenta condição intermediária e acima de 80 km/h condição rápida.



(a) Histograma.



(b) Velocidade experimental comparado com o GPS.

Figura 8. Tempo detalhado de cada etapa e Comparação de resultados.

A grande variação no tempo de associação com o ponto de acesso da Figura 8(a) é consequência do intervalo de tempo considerado, que inclui o primeiro *beacon* detectado e a associação completa com a unidade de bordo. Em regiões onde não há obstáculos, o alcance de um *beacon* é maior do que o necessário para que haja pareamento entre os dispositivos.

5. Conclusão e Trabalhos Futuros

Este artigo apresentou um sistema colaborativo para o monitoramento distribuído de veículos. Por não exigir que as unidades de acostamento estejam conectadas, o sistema pode ser utilizado mesmo em rodovias onde não existe infraestrutura, como energia ou cobertura celular.

Foram realizados experimentos em dois cenários com resultados muito similares aos obtidos por um GPS, com alta taxa de acerto ao inferir as condições dos trechos e a posição do veículo. Ao considerar a camada física no protótipo, uma parte expressiva do tempo de contato com o ponto de acesso era para executar rotinas de autenticação e endereçamento.

Como trabalho futuro pretende-se implementar um protótipo utilizando o padrão IEEE 802.11p e variar a estrutura proposta, interligando algumas unidades de acostamento para diminuir o tempo de defasagem entre pontos mais distantes. Pretende-se ainda fazer experimentos em maior escala, por meio de simulação, para analisar a quantidade de tráfego gerada pelo sistema e qual a relação com o aumento no número de carros.

Referências

- (1997). Dynamic Host Configuration Protocol. *IETF RFC 2131*.
- Balasubramanian, A., Mahajan, R. e Venkataramani, A. (2010). Augmenting Mobile 3G using WiFi. Em *Proceedings of the 8th international conference on Mobile systems, applications, and services, MobiSys '10*, p. 209–222, New York, NY, USA. ACM.
- Cheung, S. Y., Coleri, S., Dundar, B., Ganesh, S., Tan, C.-W. e Varaiya, P. (2004). Traffic Measurement and Vehicle Classification with a Single Magnetic Sensor. Institute of transportation studies, research reports, working papers, proceedings, Institute of Transportation Studies, UC Berkeley.
- Cisco (2012). Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016. Relatório técnico. Acessado em dezembro de 2012.
- Cucchiara, R., Piccardi, M. e Mello, P. (2000). Image Analysis and Rule-based Reasoning for a Traffic Monitoring System. *Intelligent Transportation Systems, IEEE Transactions on*, 1(2):119–130.
- Edelmayer, A., Miranda, M. e Nebehaj, V. (2010). Cooperative Federated Filtering Approach for Enhanced Position Estimation and Sensor Fault Tolerance in Ad-hoc Vehicle Networks. *Intelligent Transport Systems, IET*, 4(1):82–92.
- Fazenda, B., Atmoko, H., Gu, F., Guan, L. e Ball, A. (2009). Acoustic Based Safety Emergency Vehicle Detection for Intelligent Transport Systems. Em *ICROS-SICE International Joint Conference 2009*. IEEE Xplore.
- Google (2011). Google Maps. Disponível em <http://maps.google.com/support>. Acessado em dezembro de 2012.
- Kassem, N., Kosba, A. e Youssef, M. (2012). RF-Based Vehicle Detection and Speed Estimation. Em *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, p. 1–5.

- Mohan, P., Padmanabhan, V. N. e Ramjee, R. (2008). Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones. Em *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, p. 323–336, New York, NY, USA. ACM.
- Nordin, N., Zaharudin, Z., Maasar, M. e Nordin, N. (2012). Finding Shortest Path of the Ambulance Routing: Interface of A #x2217; Algorithm Using C# Programming. Em *Humanities, Science and Engineering Research (SHUSER), 2012 IEEE Symposium on*, p. 1569 –1573.
- Openwrt (2011). OpenWRT - Wireless Freedom. Disponível em <http://www.openwrt.org/>. Acessado em dezembro de 2012.
- Ribeiro Júnior, J. G., Campista, M. E. M. e Costa, L. H. M. K. (2012a). Opportunistic System for Collaborative Traffic Monitoring Using Existing IEEE 802.11 Networks. *Intelligent Vehicular Networking: V2V/V2I Communications and Applications - IEEE International Conference on Communications - ICC 2012*, p. 7294 –7298.
- Ribeiro Júnior, J. G., Costa, L. H. M. K., Campista, M. E. M., Moraes, I. M., Alves, R. S., Couto, R. S., Silva, F. O. B., Valverde, L. G., Lanza, M. L. D., Camilo, B. C. V. e Amorim, M. D. (2011). GT-ReBUS: Redes de Acesso em Ônibus Universitários. Disponível em <http://www.gta.ufrj.br/gt-rebus>. Acessado em dezembro de 2012.
- Ribeiro Júnior, J. G., Quintanilha, I. M., Campista, M. E. M. e Costa, L. H. M. K. (2012b). Evaluation of an Opportunistic Collaborative Traffic Monitoring System. *IFIP/IEEE Wireless Days Conference 2012*, p. 6.
- Thiagarajan, A., Ravindranath, L., LaCurts, K., Madden, S., Balakrishnan, H., Toledo, S. e Eriksson, J. (2009). VTrack: Accurate, Energy-aware Road Traffic Delay Estimation Using Mobile Phones. Em *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, p. 85–98, New York, NY, USA. ACM.
- Treiber, M. e Kesting, A. (2010). An Open-Source Microscopic Traffic Simulator. *Intelligent Transportation Systems Magazine, IEEE*, 2(3):6 –13.
- Valerio, D., D'Alconzo, A., Ricciato, F. e Wiedermann, W. (2009). Exploiting Cellular Networks for Road Traffic Estimation: A Survey and a Research Roadmap. Em *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, p. 1 –5.
- Yoon, J., Noble, B. e Liu, M. (2007). Surface Street Traffic Estimation. Em *Proceedings of the 5th international conference on Mobile systems, applications and services*, MobiSys '07, p. 220–232, New York, NY, USA. ACM.