

Relatório EP2/1

MAC 0463 / 5743
Computação Móvel
01/2012

Gabriel A. von Winckler (3313359)
Max Rosan dos Santos Junior (7795212)
Gesner Borges Taquary Junior(7797072)

Introdução

O objetivo desse exercício foi imaginar um cenário potencialmente realista para a utilização de redes DTN e tentar modelar e simular o seu comportamento.

Criar um cenário com essas características foi um dos grandes desafios desse exercício. Talvez por falta de criatividade, nenhum dos muitos cenários discutidos passava pelo critério da praticidade (onde uma simples rede estruturada faria muito mais sentido) ou por ser complexa demais para ser modelada utilizando os nossos poucos conhecimentos com a ferramenta ONE.

Porém, acreditamos que o problema escolhido atende bem aos requisitos, sendo um problema concreto, e que o uso da tecnologia de DTNs poderia ser adotado como solução. Esse problema também é modelável (com muito mais restrições do que imaginávamos inicialmente) no ONE em duas semanas.

Cenário

O cenário escolhido é a automatização de um estacionamento. Atualmente, o estacionamento de alguns shoppings centers já dispõe de sensores que detectam a ocupação da vaga, e informam quantas vagas estão disponíveis em determinado pavimento do estacionamento. Um exemplo é o Morumbi Shopping, como ilustrado na figura a seguir.



Figura 1: Estacionamento do Shopping Morumbi (Ernesto Rodrigues / AE)

O objetivo seria criar um sistema onde o usuário, através de um bilhete inteligente ou um smartphone, consultasse as vagas livres próximas a sua localização e reservasse uma.

De forma mais detalhada, seria criada uma rede composta pelos sensores e os clientes (bilhetes inteligentes ou smartphones) utilizando um protocolo de curta distância. Toda a comunicação seria iniciada pelo cliente, e o protocolo de reserva de vagas seria análogo ao protocolo do DHCP, com mensagens para (Discover, Offer, Request e Acknowledge).

Todos os sensores funcionariam como roteadores intermediários, quando necessário.

Dificuldades na modelagem

Acreditávamos que esse problema seria simples de ser modelado no ONE, porém não foi. Inicialmente foram adotadas algumas simplificações, como por exemplo, os carros não são criados e destruídos ao longo da simulação. Ao invés disso, e por simplicidade, mantivemos um número fixo de carros no sistema, que nunca param, indicando um número constante de carros procurando vagas. Da mesma forma, a comunicação não foi modelada de forma realista. Os sensores não possuem estado e sempre respondem da mesma forma. Isso não é um problema em si, pois ainda assim é possível extrair os parâmetros importantes da simulação, como o tempo médio e máximo para que a mensagem demora para atingir toda a rede, ou alcance da comunicação para que o sistema funcione de forma estável.

Ao longo do processo tivemos muitos problemas com o ONE, nos deixando a sensação que essa ferramenta ainda não está madura como produto, exigindo que qualquer um que deseje simular um caso, mesmo que simples, precise alterar o código, implementar um classe ou depurar uma funcionalidade.

Nesse sentido, o caso mais significativo foi o trabalho necessário para implementar os sensores. A rede de sensores deveria ser trivial de ser implementada, uma vez que consiste de nós com posições fixas. A ferramenta oferece uma classe para isso, chamada *StationaryMovement*. Porém, como viemos a descobrir depois, ela só consegue instanciar um grupo com um único objeto (ela não lê um arquivo de posições) e, o mais complicado, a posição informada é utilizada em um outro sistema de coordenadas do que os mapas e as rotas, fornecidas pelos arquivos .wkt. Para contornar isso, criamos um script que lê as posições em um .wkt e o mapa geral e gera um arquivo de configuração, criando os grupos necessários e transformando as coordenadas.

Modelo final

A simulação da rede envolve dois tipos de nós: sensores e carros. Os nós de sensores usam Bluetooth para formar uma rede mesh e então trocar informações sobre as vagas do estacionamento. Supõe-se que cada sensor está em cima de uma vaga de estacionamento. Os nós de carros ficam durante a simulação percorrendo aleatoriamente o mapa do estacionamento e estabelecendo conexão com os sensores para então saber quais vagas estão disponíveis.

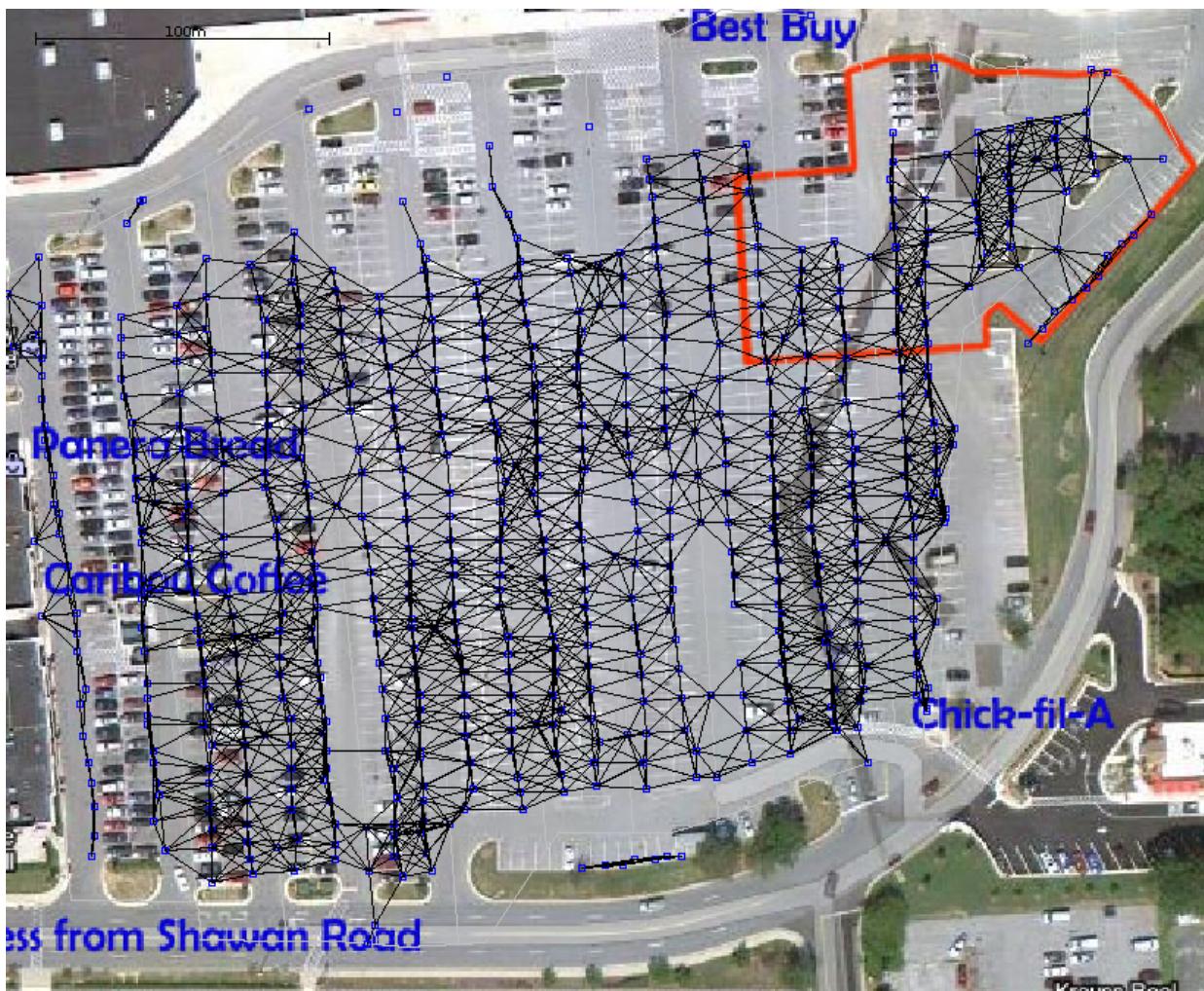
Para se obter um resultado com uma sensação realista, desenhamos um estacionamento de exemplo, através de uma imagem obtida na Internet, representando um caso típico (ver figuras abaixo).

No ONE os nós de carro são representados pelo grupo de ID *c*, sendo o movimento do grupo definido pela classe *CarMovement* e a rota é dada por *estacionamento_caminhos.wkt*. Esse grupo é composto por 70 nós.

Já cada sensor compõe um novo grupo. O padrão de movimentos dos sensores é definido pela classe *StationaryMovement*, fazendo com que os sensores fiquem estáticos.

Tanto os nós dos sensores como os nós dos carros usam a interface modelada por *SimpleBroadcastInterface*, simulando um Bluetooth com alcance de 20m e com um taxa de transferência 250 kbytes/s. O roteamento dos pacotes entre os nós é dado pelo *EpidemicRouter*.





Notamos que ao utilizar o alcance padrão do Bluetooth, 10 metros, o sistema apresentou alguma instabilidade, inexistente ao alterar o alcance para 20 metros. Isso pode estar relacionado com a velocidade dos carros e a necessidade (ou não), de se estabelecer uma sessão para iniciar a comunicação. Porém é necessária uma análise mais aprofundada para se obter qualquer conclusão.

A simulação gerou um relatório de conexões, que foi carregado no DTN TES e gerou alguns dados interessantes. Porém, para conseguir utilizar essa ferramenta, foi necessário adotar uma simulação muito curta, com apenas 128 segundos (registrado como Parking2, no sistema online). E Isso é insuficiente para permitir qualquer conclusão. Não sabemos a exata limitação da ferramenta, mas ao carregar arquivos maiores, o modelo não foi processado.