

## Acionamento Inteligente de Unidades Móveis em Situações de Emergência em Cidades

Sediane C. L. Hernandes<sup>1</sup>, Alcides Calsavara<sup>2</sup>, Luiz A. P. Lima Jr.<sup>2</sup>

<sup>1</sup>Universidade Tecnológica Federal do Paraná (UTFPR)  
85.053-525 – Guarapuava – PR – Brasil

<sup>2</sup>Programa de Pós-Graduação em Informática (PPGIA)  
Pontifícia Universidade Católica do Paraná (PUCPR)  
80.215-901 – Curitiba – PR – Brasil

{sedianec}@utfpr.edu.br, {alcides, laplima}@ppgia.pucpr.br

**Abstract.** *Calls to emergency services in cities, such as calls to the police and hospitals, often requires movement of mobile units. The protocol for activating mobile units should be efficient due to, mainly, risks that are intrinsic to emergency situations, and the high costs involved. However, typically, such protocol is performed by humans, including decision making regarding how many and which mobile units should move to the emergency site. In this article, the problem is formalized, the currently employed platforms for developing systems in Smart Cities are analyzed, and a more suitable middleware for building an intelligent mobile units activation system is proposed.*

**Resumo.** *Os chamados a serviços de emergência em cidades, tais como chamados a polícia e a hospitais, quase sempre demandam o deslocamento de unidades móveis. O protocolo de acionamento de unidades móveis deve ser eficiente devido, principalmente, aos riscos inerentes a situações de emergência e ao alto custo envolvido. No entanto, tipicamente, o protocolo é realizado por intervenção humana, com a tomada de decisão sobre quantas e quais unidades deslocar até o local de emergência. Neste artigo, o problema é formalizado, são analisadas as plataformas atuais para sistemas em Cidades Inteligentes e é proposto um middleware mais adequado para a construção de um sistema inteligente de acionamento de unidades móveis.*

### 1. Introdução

Cidades Inteligentes<sup>1</sup> buscam explorar a Tecnologia de Informação e Comunicação (TIC) na gestão dos assuntos públicos a fim de melhorar a qualidade de vida dos cidadãos [Cenedese et al., 2014]. Nos últimos anos, considerável esforço de pesquisa e desenvolvimento tem sido feito em diversas partes do mundo para se criar sistemas computacionais que auxiliem a enfrentar os grandes problemas das cidades, incluindo a gerência de grandes volumes de resíduos, a escassez de recursos, a maior dificuldade de mobilidade de pessoas e veículos, além de preocupações com segurança, saúde e educação. A Internet das Coisas<sup>2</sup>, a qual também pode ser chamada de Internet dos Objetos Inteligentes, aplicada

---

<sup>1</sup>Do Inglês, *Smart Cities*.

<sup>2</sup>Do Inglês, *Internet of Things*.

em um contexto urbano responde a muitos anseios governamentais em utilizar TIC para auxiliar na gestão de assuntos públicos, tornando possível, assim, automatizar muitos serviços de uma cidade. A chamada *Urban IoT* trata do uso de objetos inteligentes em um contexto urbano de modo a oferecer serviços que venham a auxiliar a administração pública das cidades, as empresas e os cidadãos [Zanella et al. 2014].

Um dos serviços de uma cidade que pode se beneficiar dessa tecnologia é o atendimento em situações de emergência, como incêndios, acidentes de trânsito, etc. No entanto, ainda há pouco relato na literatura sobre teorias e sistemas desenvolvidos para a implementação desse serviço. Trabalhos na área de *Emergency Response*, tais como [Chitumalla et al. 2008], [Patsakis et al. 2015], [Milis et al. 2016], [Abu-Elkheir et al. 2016] e [Dragoicea et al. 2014], focam em apoiar a fase de execução do serviço, especialmente em caso de catástrofes, e dão pouca ênfase no apoio ao processo de tomada de decisão sobre o deslocamento de unidades móveis (veículos propriamente equipados e com profissionais qualificados) para o local de uma emergência, isto é, sobre o problema de acionamento das unidades móveis mais adequadas para atuar em uma situação em particular. Este artigo discute esse problema, chegando a uma definição formal do problema e sua solução, avalia alternativas de projeto e de implementação e, finalmente, propõe um novo middleware para ser usado como plataforma para o acionamento de unidades móveis.

O restante deste artigo está organizado como segue. A Seção 2 discute sobre a complexidade do problema de prestar serviços de emergência em cidades e mostra como pode ser dividido em problemas menores. A Seção 3 formaliza o problema de acionamento de unidades móveis em situações de emergência, com a sua respectiva solução. A Seção 4 discute os requisitos de um sistema computacional para o acionamento de unidades móveis. A Seção 5 discorre sobre a adequação dos middlewares baseados em eventos como plataforma para sistemas computacionais em Cidades Inteligentes, em especial para o acionamento de unidades móveis. A Seção 6 apresenta a proposta de um novo middleware como plataforma para um sistema computacional de acionamento de unidades móveis. Finalmente, a Seção 7 lista as contribuições do trabalho de pesquisa realizado.

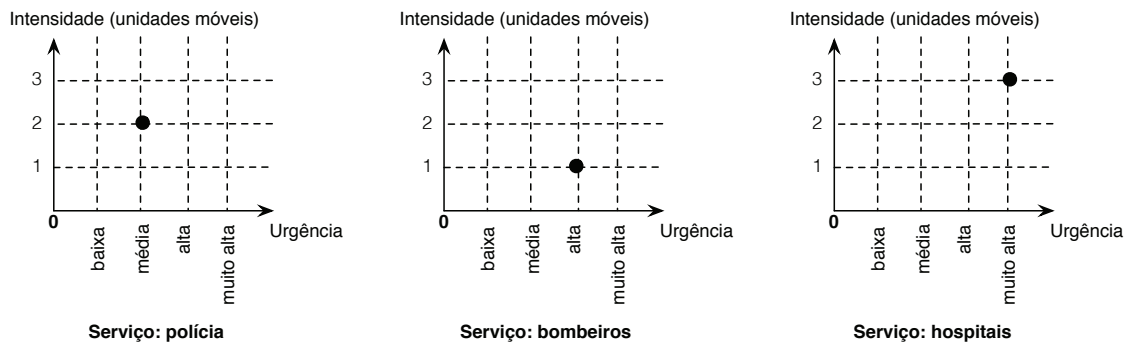
## 2. Serviço de Emergência em Cidades Inteligentes

As situações de emergência possíveis em uma cidade são inúmeras e podem variar significativamente quanto às necessidades de urgência, de intensidade e de diversidade de ação, como ilustram os seguintes exemplos:

- Uma situação de assalto em um estabelecimento comercial requer mais *urgência* de ação por parte da polícia do que uma situação de vandalismo contra edificações.
- Uma situação de incêndio de grandes proporções em um edifício requer uma ação de maior *intensidade*, isto é, com a participação de um número maior de bombeiros e equipamentos contra incêndio do que em uma situação de princípio de afogamento de uma pessoa em um rio.
- Uma situação de acidente de trânsito, pode requerer uma ação conjunta entre polícia, bombeiros e hospitais, isto é, essa ação apresentaria uma *diversidade* de três tipos de serviços, e não de apenas um tipo, como nos exemplos anteriores (somente polícia ou somente bombeiros).

Invariavelmente, uma situação de emergência requer o deslocamento de uma ou mais *unidades móveis*, isto é, veículos apropriadamente equipados e com pessoal técnico especializado.

Como critério de classificação, pode-se posicionar uma situação de emergência em um espaço tridimensional com os seguintes eixos: *Serviço*, *Urgência* e *Intensidade*. No eixo *Serviço* estão representados os vários tipos de serviços, tais como polícia, bombeiros e hospitais. No eixo *Urgência*, a grau de urgência da situação, tais como baixa, média, alta e muito alta, sendo que cada tipo de serviço pode ter seu próprio grau de urgência. No eixo *Intensidade*, a quantidade de unidades móveis alocadas, sendo que pode ser definida uma intensidade diferente para cada tipo de serviço. Assim, o posicionamento de uma situação de emergência é definido por um conjunto de pontos, sendo cada um associado a um tipo de serviço e representado no plano definido por urgência e intensidade. A Figura 1 mostra um exemplo de classificação de uma situação de emergência que requer uma diversidade de três tipos de serviços: polícia, bombeiros e hospitais. Para cada um desses serviços, há um grau específico de urgência e de intensidade.



**Figura 1. Exemplo de classificação de uma situação de emergência**

Na prática, observa-se que, independentemente do grau de urgência, espera-se que o provimento dos serviços ocorra sempre com a maior brevidade possível; qualquer atraso no provimento é justificável somente se houver outra situação com maior grau de urgência sendo atendida, de forma que as unidades móveis sejam, momentaneamente, insuficientes. O provimento de serviços em uma situação de emergência ocorre em, ao menos, duas fases, a saber:

1. *Fase de acionamento dos serviços:* Nesta fase, os correspondentes serviços devem ser comunicados da situação de maneira rápida e confiável. A decisão sobre quais serviços comunicar pode ocorrer em um único passo, quando todos os serviços necessários são comunicados por um único agente (pessoa ou sistema) que tenha detetado a situação de emergência, ou em uma sequência de passos, quando esse agente comunica somente parte dos serviços necessários, os quais, então, decidem comunicar outros, e assim sucessivamente, até que todos os serviços necessários sejam comunicados. Quando comunicado, um serviço determina a urgência e a intensidade da sua ação específica e, com isso, decide quantas e quais unidades móveis serão enviadas ao local da emergência.
2. *Fase de execução dos serviços:* Nesta fase, as unidades móveis deslocam-se até o local da emergência e, quando chegam, executam a ação propriamente dita. Nessa execução, as unidades móveis podem precisar de informações atualizadas sobre

a situação de emergência para fazer ajustes nas suas ações. Ainda nesta fase, é possível que novos acionamentos de serviço ocorram como consequência de revisão das dimensões diversidade e intensidade da ação. Ou seja, a classificação da situação de emergência pode mudar dinamicamente.

Portanto, o problema de provimento de serviços em situações de emergência é complexo e, para efeito de análise e pesquisa, pode ser dividido em problemas menores, mais fundamentais, da seguinte forma:

1. *Definição da estratégia inicial de ação*: Classificar a situação de emergência, determinando os tipos de serviços necessários, bem como a urgência e a intensidade de cada um. Essa definição de estratégia pode se beneficiar de informações coletadas a partir de drones e smartphones próximos do local de emergência, bem como de informações disponíveis em redes sociais.
2. *Acionamento de unidades móveis*: Selecionar e comunicar um conjunto de unidades móveis que satisfaça a necessidade de serviços (em diversidade e intensidade), de forma que os serviços sejam executados pelas unidades mais adequadas no momento, considerando as suas características particulares, incluindo disponibilidade de equipamentos e técnicos, bem como a sua distância relativa do local da emergência para que o serviço seja executado com a maior brevidade possível. Apesar de móvel, uma unidade poderia operar de forma que, em princípio, fosse acionada somente quando estivesse parada num local padrão (por exemplo, uma ambulância somente seria acionada quando estivesse parada no hospital), isto é, o fato de estar fora do local padrão (parada ou em movimento) significa que já está atuando em uma situação de emergência, logo está indisponível para atuar em outras situações. No entanto, é possível que, por uma definição de estratégia de ação, ocorra o cancelamento de um acionamento enquanto a unidade móvel já atua em uma ação e, então, seja acionada para atuar em outra situação de emergência, possivelmente mais urgente. Portanto, no caso geral, o problema de acionamento deve considerar que as unidades móveis possam se deslocar constantemente.
3. *Apoio na execução*: Supervisionar a execução da ação e orientar as unidades móveis na execução das suas ações específicas, fornecendo informações atualizadas sobre a situação de emergência.
4. *Avaliação permanente da estratégia de ação*: Avaliar permanentemente a eficácia da estratégia de ação em execução e, se necessário, readequar a estratégia através de alterações na sua classificação, isto é, nos tipos de serviços e correspondentes urgência e intensidade.

Uma cidade inteligente deve dispor de sistemas computacionais que auxiliem na solução de cada um desses problemas, preferencialmente, com o mínimo de intervenção humana, tanto para se obter maior eficácia no provimento dos serviços, como para o uso mais eficiente dos recursos, em especial, as unidades móveis.

### 3. Problema de Acionamento de Unidades Móveis

O foco deste trabalho de pesquisa está no problema de acionamento de unidades móveis, conforme apresentado na Seção 2, e visa desenvolver uma solução computacional para o mesmo. Para tanto, nesta seção, o problema é formalizado como segue. Sejam:

- $\Gamma = \{S_1, \dots, S_n\}$  o conjunto dos  $n \geq 1$  serviços disponíveis

- $\Phi = \{\alpha_1, \dots, \alpha_r\}$  o conjunto de  $r \geq 1$  pontos geográficos de referência de uma cidade (cada ponto corresponde ao centro de uma região da cidade)
- $\epsilon$  uma situação de emergência
- $\Psi_\epsilon \in \Phi$  é o ponto geográfico de referência associado a  $\epsilon$
- $\Theta_i = \{v_{i,1}, \dots, v_{i,m}\}$  o conjunto de  $m \geq 1$  unidades móveis de  $S_i \in \Gamma$
- $\Delta_{i,\epsilon} \in \mathbb{N}$  a intensidade de  $S_i \in \Gamma$  para atuar em  $\epsilon$ , dada em número de unidades móveis
- $\beta_{i,j,\epsilon} = \begin{cases} 1 & \text{se } v_{i,j} \in \Theta_i \text{ está selecionada para atender } \epsilon \\ 0 & \text{caso contrário} \end{cases}$
- $\Omega_{i,j,\alpha_k}, 0 \leq \Omega_{i,j,\alpha_k} \leq 1$ , o nível de adequação da unidade móvel  $v_{i,j} \in \Theta_i$  relativo a  $\alpha_k \in \Phi$

Então, a solução do problema de acionamento de unidades móveis para atender  $\epsilon$  consiste em determinar  $\beta_{i,j,\epsilon}$ ,  $1 \leq j \leq |\Theta_i|$ , para cada serviço  $S_i \in \Gamma$  que faz parte da estratégia de ação, tal que o nível de adequação geral das unidades móveis selecionadas de  $\Theta_i$  para atuar em  $\epsilon$ , denominado  $\lambda_{i,\epsilon}$ , seja maximizado. Assim, o problema pode ser formalizado da seguinte forma:

$$\text{Maximizar } \lambda_{i,\epsilon} = \sum_{j=1}^{|\Theta_i|} (\Omega_{i,j,\Psi_\epsilon} \times \beta_{i,j,\epsilon})$$

$$\text{Sujeito a } \sum_{j=1}^{|\Theta_i|} \beta_{i,j,\epsilon} \leq \Delta_{i,\epsilon}$$

Dessa forma, o subconjunto de unidades móveis do serviço  $S_i \in \Gamma$  selecionadas para atuação é dado por:

$$\{v_{i,j} \in \Theta_i \mid \beta_{i,j,\epsilon} = 1\}$$

A Figura 2 mostra um exemplo de situação de emergência. A área da cidade é mapeada em uma matriz  $8 \times 8$  de células quadradas idênticas. Como consequência desse mapeamento, é definido um sistema de coordenadas sobre a área da cidade. São definidos 16 pontos geográficos de referência de forma que cada um seja a referência da área definida pelas quatro células à sua volta. O ponto  $\rho = (3, 3)$  é referência para a área definida pelo quadrado com vértices  $(2, 2)$ ,  $(2, 4)$ ,  $(4, 4)$  e  $(4, 2)$ . A situação de emergência  $\epsilon$  ocorre no ponto  $P = (2.5, 3.5)$ , logo o seu ponto de referência é  $\Psi_\epsilon = \rho = (3, 3)$ . O serviço  $S_i$  demandado dispõe de quatro unidades móveis para atuação:  $\Theta_i = \{v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}\}$ . Se a intensidade da situação de emergência for dois ( $\Delta_{i,\epsilon} = 2$ ), a resolução do problema selecionará as duas unidades móveis mais adequadas com relação ao ponto  $\Psi_\epsilon = (3, 3)$ . Assim, se os níveis de adequação forem  $\Omega_{i,1,\Psi_\epsilon} = 0.4$ ,  $\Omega_{i,2,\Psi_\epsilon} = 0.8$ ,  $\Omega_{i,3,\Psi_\epsilon} = 0$  e  $\Omega_{i,4,\Psi_\epsilon} = 0.5$ , as unidades  $v_{i,2}$  e  $v_{i,4}$  serão selecionadas para atuar na situação de emergência.

#### 4. Sistema Computacional para Acionamento de Unidades Móveis

O processo de resolução do problema de acionamento de unidades móveis, conforme definido na Seção 3, pode ser realizado através de um sistema computacional a fim de reduzir e, possivelmente, eliminar a intervenção humana no processo. Pela formulação do

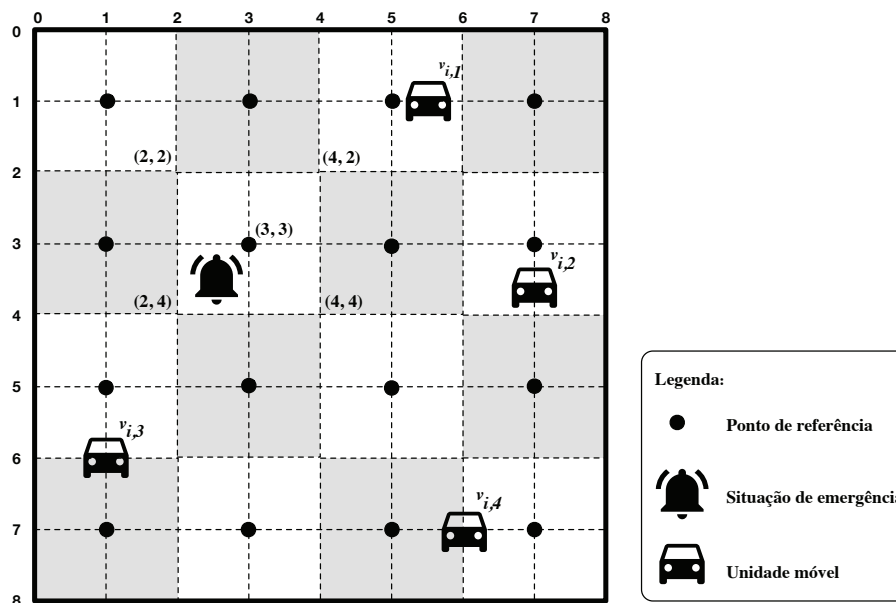


Figura 2. Exemplo de uma situação de emergência

problema, observa-se que a principal dificuldade na sua resolução para um dado serviço  $S_i \in \Gamma$  está na determinação de  $\Omega_{i,j,\Psi_\epsilon}$  para todo  $v_{i,j} \in \Theta_i$ . Tendo-se tal informação, a resolução do problema é relativamente trivial do ponto de vista matemático. Em outras palavras, a dificuldade está na determinação do nível de adequação de cada unidade móvel de um serviço para a sua atuação em uma situação de emergência detetada em certo ponto geográfico de referência.

A dificuldade na determinação da adequação de uma unidade móvel decorre de vários fatores. Primeiramente, o nível de adequação varia com a condição corrente da unidade móvel: quando está livre, o seu nível de adequação pode ser qualquer valor entre zero e um, mas, enquanto estiver atuando em uma situação de emergência, o seu nível de adequação é necessariamente zero, uma vez que está indisponível para atuar em novas situações de emergência. Depois, há uma variação intrínseca desse nível de adequação à medida que uma unidade móvel aproxima-se ou afasta-se de um ponto geográfico de referência (quanto mais próximo, maior é o nível de adequação). Finalmente, pode variar de acordo com características particulares de cada unidade móvel, tais como a sua capacidade padrão de prestar serviço (por exemplo, um veículo novo é mais adequado que um veículo antigo, pois desloca-se mais rapidamente), e a disponibilidade ou indisponibilidade momentânea de certos equipamentos ou técnicos.

A eficácia do sistema computacional que resolve o problema de acionamento de unidades móveis está diretamente relacionada com a atualização do estado referente ao nível de adequação das unidades móveis, pois é o fator determinante para a precisão do resultado obtido com a resolução do problema. Essa atualização do estado para um dado serviço pode ser feita segundo uma das seguintes abordagens:

- *Atualização reativa*: o estado é atualizado apenas no momento em que é necessário, isto é, na resolução do problema.
- *Atualização pró-ativa*: o estado é permanentemente atualizado, de forma que es-



teja pronto sempre que for necessário resolver o problema.

A abordagem pró-ativa permite que o processo de tomada de decisão, isto é, resolução do problema e consequente determinação do subconjunto de unidades móveis selecionadas, ocorra de forma imediata, uma vez que todos os dados necessários estão disponíveis e atualizados, enquanto que a abordagem reativa impõe um atraso no processo (tempo para atualização do estado) que pode ser significativamente grande. Por outro lado, a abordagem pró-ativa implica em consumo relativamente muito maior de recursos de sistema, incluindo processamento e comunicação.

Outra questão a ser considerada na arquitetura do sistema diz respeito à abordagem do protocolo de tomada de decisão, a qual pode ser uma das seguintes:

- *Decisão centralizada*: a tomada de decisão é feita por um único processo que concentra todo o estado do sistema (níveis de adequação das unidades móveis para todos os pontos de referência).
- *Decisão descentralizada*: a tomada de decisão é feita por um conjunto de processos cooperantes, sendo que cada processo mantém uma parte do estado do sistema. Esses processos, denominados *brokers*, são distribuídos, tal que cada um gerencie uma região da cidade, sem que haja sobreposição. As funções de gerenciamento de um broker incluem:
  1. Monitorar as unidades móveis presentes na sua região para a atualização do correspondente estado (nível de adequação de cada unidade móvel para cada ponto de referência da cidade).
  2. Iniciar o processo de tomada de decisão para todas as situações de emergência na sua região; a tomada de decisão baseia-se em um protocolo de consenso entre os brokers.

A Figura 3 mostra um exemplo de arquitetura na qual a tomada de decisão é feita em cooperação por quatro brokers, sendo cada um responsável por uma região correspondente a 1/4 da área da cidade. Assim, a tomada de decisão para uma situação de emergência que demanda o serviço  $S_2$  será iniciada pelo broker  $B_1$ , mas envolverá todos os demais brokers em cujas regiões esteja alguma unidade móvel daquele serviço, isto é, brokers  $B_3$  e  $B_4$ . Da mesma forma, a tomada de decisão para a situação de emergência que demanda o serviço  $S_1$  será iniciada pelo broker  $B_4$ , mas envolverá os brokers  $B_2$  e  $B_3$ .

A função da camada física do sistema computacional na abordagem centralizada para tomada de decisão é prover um servidor para hospedar o processo que toma as decisões e a comunicação (para fins de para atualização de estado e acionamento das unidades) entre esse servidor e as unidades móveis. Na abordagem descentralizada, a função da camada física depende da estratégia de alocação dos brokers, que pode ser baseada em cluster ou distribuída por regiões da cidade, como segue.

- *Alocação baseada em cluster*: A estratégia é alocar todos os brokers em um cluster implementado como um *data center* próprio ou através de computação em nuvem. Assim, a função da camada física é semelhante à sua função na abordagem centralizada: prover um cluster e a comunicação entre esse e as unidades móveis (a comunicação entre os brokers é resolvida internamente no próprio cluster).

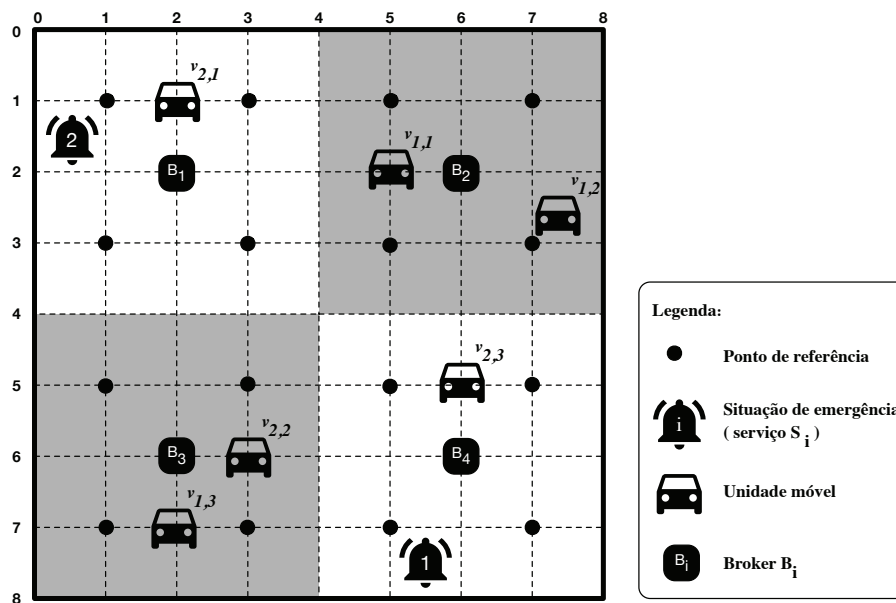


Figura 3. Exemplo de arquitetura descentralizada para tomada de decisão

- *Alocação distribuída por regiões da cidade:* A estratégia é alocar cada broker em um servidor próprio que esteja fisicamente instalado na correspondente região gerenciada. Assim, a camada física deve fornecer um servidor para cada região, comunicação dentro de cada região (entre o respectivo servidor e as unidades móveis presentes na região) e comunicação entre os servidores.

Na decisão descentralizada, a atualização de estado devido a deslocamento é tratada especificamente pelo broker da região onde se encontra a unidade móvel após o deslocamento, enquanto que, na abordagem centralizada, todo e qualquer deslocamento é tratado globalmente pelo processo único do sistema. Ainda, na decisão descentralizada, a tomada de decisão envolve apenas um subconjunto dos brokers (da região da situação de emergência e das regiões com unidades móveis pertinentes), podendo haver muitas tomadas de decisão em paralelo, enquanto que a abordagem centralizada permite apenas uma tomada de decisão por vez. Ou seja, na decisão descentralizada, o processamento é realizado em paralelo por muitos processos, enquanto que, na decisão centralizada, todo o processamento é feito em único ponto, o que pode comprometer a escalabilidade do sistema. Além disso, a decisão descentralizada com alocação distribuída de brokers favorece a disponibilidade e a escalabilidade do sistema, pois, em princípio, não apresenta um ponto único falha e nenhum ponto crítico de sobrecarga (de processamento, armazenamento ou comunicação). A decisão descentralizada com alocação de brokers em cluster também favorece a disponibilidade do sistema, uma vez que clusters, em geral, são tolerantes a falhas, mas o favorecimento da escalabilidade fica restrito ao aspecto de processamento, uma vez que a comunicação é altamente concentrada em um único ponto da rede: o próprio cluster. Finalmente, a abordagem descentralizada, em especial com alocação distribuída de brokers, é naturalmente mais complexa, tanto no seu desenvolvimento quanto na sua operacionalização. A Tabela 1 resume as principais vantagens e desvantagens esperadas das abordagens, considerando as combinações possíveis das suas alternativas.



**Tabela 1. Vantagens e desvantagens das abordagens esperadas para a arquitetura de um sistema computacional para acionamento de unidades móveis**

Tomada de Decisão	Atualização do Estado	
	<i>reativa</i>	<i>pró-ativa</i>
<i>centralizada</i>	<ul style="list-style-type: none"> <li>● baixa complexidade</li> <li>● poucas mensagens</li> <li>● decisão demorada</li> <li>● ponto único de falha</li> <li>● baixa escalabilidade</li> </ul>	<ul style="list-style-type: none"> <li>● baixa complexidade</li> <li>● muitas mensagens</li> <li>● decisão rápida</li> <li>● ponto único de falha</li> <li>● baixa escalabilidade</li> </ul>
<i>descentralizada, com alocação de brokers em cluster</i>	<ul style="list-style-type: none"> <li>● média complexidade</li> <li>● poucas mensagens</li> <li>● decisão demorada</li> <li>● alta disponibilidade</li> <li>● média escalabilidade</li> </ul>	<ul style="list-style-type: none"> <li>● média complexidade</li> <li>● muitas mensagens</li> <li>● decisão rápida</li> <li>● alta disponibilidade</li> <li>● média escalabilidade</li> </ul>
<i>descentralizada, com alocação distribuída de brokers</i>	<ul style="list-style-type: none"> <li>● alta complexidade</li> <li>● poucas mensagens</li> <li>● decisão demorada</li> <li>● alta disponibilidade</li> <li>● alta escalabilidade</li> </ul>	<ul style="list-style-type: none"> <li>● alta complexidade</li> <li>● muitas mensagens</li> <li>● decisão rápida</li> <li>● alta disponibilidade</li> <li>● alta escalabilidade</li> </ul>

A escolha entre as alternativas de abordagens para que o sistema computacional apresente uma relação custo-benefício satisfatória deve se basear nas características particulares da cidade, nas restrições de recursos e na qualidade desejada para os serviços. Entre as variáveis determinantes para essa escolha, destacam-se as seguintes:

- *Quantidade de pontos de referência*: uma quantidade maior de pontos propicia maior precisão na resolução do problema, mas com maior custo de processamento e comunicação.
- *Quantidade de unidades móveis*: uma quantidade maior de unidades móveis permite obter soluções mais satisfatórias para o problema, mas aumenta o impacto em processamento e comunicação.
- *Frequência de situações de emergência*: quanto maior a frequência, maior é a chance de não ser possível resolver o problema por falta de unidades móveis.
- *Mobilidade das unidades*: no caso de abordagem pró-ativa, quanto maior a mobilidade, maior é a necessidade de atualização de estado, aumentando o consumo de processamento e comunicação.
- *Frequência de atualização de estado*: no caso de abordagem pró-ativa, quanto mais frequente for a atualização, mais precisa é a solução do problema, ao custo de mais processamento e comunicação.

Portanto, a configuração que tende a propiciar a construção de um sistema com mais qualidade de serviço, logo de uso mais geral, é a de decisão descentralizada com alocação distribuída de brokers, especialmente no caso de cidades inteligentes, onde a disponibilidade e a escalabilidade do sistema são primordiais. O grande desafio está em selecionar e empregar ferramentas de construção de sistemas computacionais que assegurem as vantagens esperadas e, ao mesmo tempo, mitiguem as suas desvantagens.

## 5. Middleware baseado em Eventos

Em [Razzaque et al. 2016], é apresentada uma análise de 61 middlewares quanto à sua adequação para IoT. Os autores concluem que middlewares baseados em eventos são apropriados para aplicações móveis e reativas, pois permitem forte desacoplamento entre os componentes de uma aplicação distribuída, nesse caso, assinantes e publicadores de eventos. Mas, apontam, como desvantagens, o cumprimento insatisfatório de alguns requisitos, tais como interoperabilidade e segurança. Além disso, salientam que o paradigma de programação provido por esses middlewares não é suficientemente flexível, isto é, não se aplicaria para muitas aplicações relevantes.

A pouca flexibilidade do paradigma de programação dos middlewares baseados em eventos deve-se, basicamente, aos modelos de envio de eventos que implementam: o modelo publish-subscribe e o modelo ponto-a-ponto. No modelo publish-subscribe, os publicadores de eventos desconhecem os assinantes, logo, eventos são disseminados para todo o conjunto de assinantes, mas são entregues apenas aos assinantes interessados; é função do middleware fazer a entrega aos assinantes corretos. Para [Rheddane et al. 2014], o modelo publish-subscribe é usado como sinônimo do paradigma de comunicação *um-para-muitos*, onde, para cada mensagem, seu recebimento é garantido para todos os assinantes inscritos através de tópicos. No modelo ponto-a-ponto, produtores enviam mensagens para filas, enquanto recebedores podem estar registrados com algumas filas específicas para, assincronamente, recuperar as mensagens e, então, confirmá-las [Musolesi et al. 2004]. Cada mensagem é entregue somente uma vez, para somente um consumidor, isto é, somente um consumidor pode obter a mensagem. Essa forma de comunicação é também conhecida como *um-para-um*, na qual cada mensagem produzida é consumida apenas uma vez, sendo que isso é realizado pela fila de mensagens. Dessa forma, esse modelo de comunicação permite múltiplos consumidores conectarem-se à fila, mas somente um dos consumidores pode consumir a mensagem. Assim, mensagens são sempre entregues e podem ser armazenadas na fila até que um consumidor esteja pronto para recuperá-la. É função do middleware rotear as mensagens de subscrição e as mensagens com eventos até os assinantes.

Essa pouca flexibilidade foi confirmada no estudo dos middlewares publish-subscribe Hermes [Pietzuch and Bacon 2002], Steam [Meier and Cahill 2002], Siena [Carzaniga et al. 2003], EMMA [Musolesi et al. 2006], Mires [Souto et al. 2006], SensorBus [Ribeiro et al. 2005], RUNES [Costa et al. 2007], PSWare [Lai et al. 2009], PRISMA [Silva et al. 2014] e TinyDDS [Boonma and Suzuki 2010]. Buscou-se compreender arquitetura, modelos de assinatura, método de recebimento de mensagens, topologia dos servidores e, principalmente, como acontece o roteamento de eventos em cada trabalho. Os middlewares Hermes, EMMA, Siena, Mires, RUNES, PSWare e TinyDDS possuem arquitetura distribuída, o middleware Steam arquitetura parcialmente distribuída e os middlewares SensorBus e PRISMA arquitetura centralizada. Hermes utiliza para assinatura de eventos uma adaptação da assinatura baseada em tópico e conteúdo; STEAM utiliza assinatura baseada em tópico, conteúdo e localização; Siena, SensorBus, Runes e PSWare assinatura por conteúdo; enquanto todos os outros middlewares utilizam assinatura baseada em tópico. SensorBus e PRISMA possuem método de recebimento de mensagens pull, enquanto todos os outros middlewares push. Quanto à topologia dos servidores, Hermes e Mires possuem topologia peer-to-peer cíclica; Steam e Siena topologia peer-to-peer acíclica; Runes, PSWare e PRISMA topologia hierárquica. Em relação ao

roteamento de eventos, Hermes utilizou roteamento de eventos seletivo do tipo Rendez-vous, STEAM e Siena *flooding subscription*; entretanto, Siena considera o caminho mais curto para o envio dos eventos. EMMA utiliza o roteamento epidêmico, Mires multi-hop e TinyDDS *flooding event*. A maioria dos middlewares foram desenvolvidos para utilização em redes de sensores sem fio, exceto Hermes e Siena para redes com fio, Steam e EMMA para redes ad-hoc. Com exceção de EMMA que implementa os modelos de mensagens ponto a ponto e publish-subscribe, todos os outros middlewares analisados implementam o modelo de mensagens publish-subscribe.

Em nenhum desses middlewares é possível, por exemplo, criar uma aplicação na qual somente um (os  $n$  assinantes de maior prioridade ou mais adequados, segundo algum critério específico da aplicação) recebam um evento, como é o requisito para um sistema computacional para acionamento de unidades móveis, conforme discutido na Seção 4. Por essa razão, é proposto um novo middleware na Seção 6.

## 6. Middleware E2BS

O middleware *Event to Best Subscribers* (E2BS), aqui proposto, visa cobrir a lacuna identificada em plataformas para o desenvolvimento de aplicações baseadas em eventos, conforme discutido na Seção 5, tal que possa ser usado como plataforma para a implementação de um sistema computacional para o acionamento de serviços de emergência em cidades, segundo a abordagem de tomada de decisão descentralizada, descrita na Seção 4. O middleware E2BS encontra-se em fase de implementação e tem como base o trabalho anterior dos autores sobre Campos Magnéticos Virtuais [Calsavara and Lima Jr. 2010], inicialmente desenvolvido para o roteamento de mensagens em rede ad hoc, de forma a garantir o balanceamento de carga entre réplicas de servidores.

O princípio do E2BS consiste em criar um conjunto de redes overlay sobre a rede física que conecta brokers e unidades móveis, tal que haja uma rede overlay para cada combinação de ponto de referência e tipo de serviço. A função de uma rede overlay para um ponto de referência  $P$  e um serviço  $S$  é fazer com que um evento referente a uma situação de emergência associada a  $P$  seja notificado a um subconjunto de unidades móveis de  $S$ , selecionadas de acordo com a definição do problema na Seção 3; a entrega do evento a uma unidade móvel a aciona para atuar na situação de emergência. O conjunto de nós da respectiva rede overlay é composto pelo broker que gerencia a região da cidade que contém  $P$ , os brokers que possuem na sua região de gerenciamento alguma unidade móvel de  $S$  e as próprias unidades móveis de  $S$ .

A Figura 4 mostra um exemplo de como as redes overlay são configuradas para a cidade da Figura 3. Como a cidade possui 16 pontos de referência e dois serviços, devem ser configuradas 32 redes overlay. Na rede física, assume-se que todos os brokers estejam conectados à Internet, o que é válido para a abordagem de alocação de brokers distribuídos por regiões da cidade. Caso a alocação seja baseada em cluster, a comunicação entre brokers é normalmente feita por uma rede local de alta velocidade, mas também pode ser via Internet. A comunicação entre unidades móveis e brokers é, necessariamente, via rede sem fio, podendo ser completada com comunicação via rede cabeada, incluindo a própria Internet (não mostrado na figura). As topologias de rede overlay mostradas aplicam-se aos quatro pontos de referência da região gerenciada pelo broker  $B_1$ : devem

ser configuradas quatro redes overlay para o serviço  $S_1$  e quatro para  $S_2$ . Embora as quatro redes overlay para um serviço tenham a mesma topologia, o estado (informações sobre o nível de adequação das unidades móveis com respeito ao ponto de referência) que cada uma mantém é diferente. Os arcos representados numa rede overlay estão no sentido da propagação de informação sobre adequação das unidades móveis, e no sentido inverso da disseminação dos eventos. Um evento publicado por  $B_1$  com solicitação de duas unidades móveis do serviço  $S_1$  para um ponto  $\rho$  será encaminhado às duas unidades móveis de  $S_1$  mais adequadas a  $\rho$ , ou seja, duas entre  $v_{1,1}$ ,  $v_{1,2}$  e  $v_{1,3}$ , fazendo roteamento do evento através de  $B_2$  e  $B_3$ .

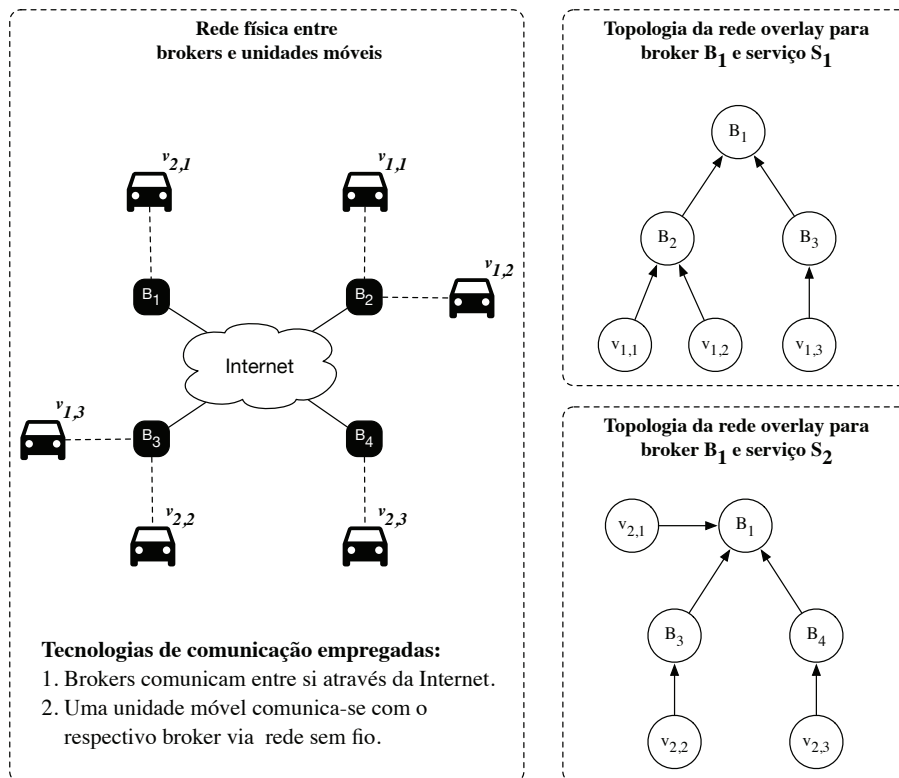


Figura 4. Exemplo de configuração de redes overlay

## 7. Conclusão

A compreensão das questões envolvendo os serviços de emergência em cidades foi fundamental para o isolamento do problema de acionamento de unidades móveis. A formalização desse problema e correspondente solução permitiu especificar os requisitos para a construção do respectivo sistema computacional que dispense a intervenção humana, em consonância com o conceito de Cidades Inteligentes. A pesquisa sobre plataformas para a implementação de tal sistema mostraram a ausência de um middleware apropriado, motivando a proposta de um novo tipo de middleware baseado em eventos. O middleware proposto, já em fase de desenvolvimento, permitirá avaliar as abordagens de projeto discutidas na Seção 4, bem como verificar quantitativamente as suas vantagens e desvantagens esperadas, descritas na Tabela 1, incluindo métricas de complexidade, eficácia e custo. Em especial, será possível avaliar se o custo (em termos e mensagens e de processamento) decorrente da atualização dinâmica das redes overlay é compensado pela melhor adequação das unidades móveis acionadas em situações de emergência.

## Referências

- Abu-Elkheir, M., Hassanein, H. S., and Oteafy, S. M. (2016). Enhancing emergency response systems through leveraging crowdsensing and heterogeneous data. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International*, pages 188–193. IEEE.
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376.
- Boonma, P. and Suzuki, J. (2010). TinyDDS: An interoperable and configurable publish/subscribe middleware for wireless sensor networks. In *Principles and Applications of Distributed Event-based Systems*, chapter 9. IGI Global.
- Calsavara, A. and Lima Jr., L. A. P. (2010). Routing based on message attraction. In *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops*, pages 189–194.
- Carzaniga, A., Rosenblum, D. S., and Wolf, A. L. (2003). Design and evaluation of a wide-area event notification service. In *Foundations of Intrusion Tolerant Systems, 2003 [Organically Assured and Survivable Information Systems]*, pages 283–334.
- Chen, M., Wan, J., and Li, F. (2012). Machine-to-machine communications: Architectures, standards and applications. *KSII Transaction on Internet and Information Systems*, 6(2):480–497.
- Chitumalla, P. K., Harris, D., Thuraisingham, B., and Khan, L. (2008). Emergency response applications: Dynamic plume modeling and real-time routing. *IEEE Internet Computing*, 12(1).
- Costa, P., Coulson, G., Gold, R., Lad, M., Mascolo, C., Mottola, L., Picco, G. P., Sivaharan, T., Weerasinghe, N., and Zachariadis, S. (2007). The RUNES middleware for networked embedded systems and its application in a disaster management scenario. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*, pages 69–78.
- Dragoicea, M., Patrascu, M., and Serea, G. A. (2014). Real time agent based simulation for smart city emergency protocols. In *System Theory, Control and Computing (ICSTCC), 2014 18th International Conference*, pages 187–192. IEEE.
- Gyrard, A. and Serrano, M. (2015). A unified semantic engine for Internet of Things and Smart Cities: From sensor data to end-users applications. In *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pages 718–725.
- Lai, S., Cao, J., and Zheng, Y. (2009). PSWare: A publish/subscribe middleware supporting composite event in wireless sensor network. In *2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–6.
- Meier, R. and Cahill, V. (2002). STEAM: event-based middleware for wireless ad hoc networks. In *Proceedings 22nd International Conference on Distributed Computing Systems Workshops*, pages 639–644.
- Milis, G., Kolios, P., Van Melick, G., Staykova, T., Helsloot, I., Ellinas, G., Panayiotou, C., and Polycarpou, M. (2016). Integrated modelling of medical emergency response



- process for improved coordination and decision support. *Healthcare Technology Letters*, 3(3):197–204.
- Musolesi, M., Mascolo, C., and Hailes, S. (2004). Adapting asynchronous messaging middleware to ad hoc networking. In *Proceedings of the 2nd Workshop on Middleware for Pervasive and Ad-hoc Computing*, MPAC '04, pages 121–126, New York, NY, USA. ACM.
- Musolesi, M., Mascolo, C., and Hailes, S. (2006). EMMA: Epidemic Messaging Middleware for Ad hoc networks. *Personal and Ubiquitous Computing*, 10(1):28–36.
- Patsakis, C., Papageorgiou, A., Falcone, F., and Solanas, A. (2015). s-health as a driver towards better emergency response systems in urban environments. In *Medical Measurements and Applications (MeMeA), 2015 IEEE International Symposium on*, pages 214–218. IEEE.
- Pietzuch, P. R. and Bacon, J. M. (2002). Hermes: A distributed event-based middleware architecture. In *Distributed Computing Systems Workshops, 2002. Proceedings. 22nd International Conference on*, pages 611–618. IEEE.
- Razzaque, M. A., Milojevic-Jevric, M., Palade, A., and Clarke, S. (2016). Middleware for Internet of Things: A survey. *IEEE Internet of Things Journal*, 3(1):70–95.
- Rheddane, A. E., Palma, N. D., Tchana, A., and Hagimont, D. (2014). Elastic message queues. In *2014 IEEE 7th International Conference on Cloud Computing*, pages 17–23.
- Ribeiro, A. R. L., Silva, F. C. S., Freitas, L. C., Costa, J. a. C., and Francês, C. R. (2005). SensorBus: A middleware model for wireless sensor networks. In *Proceedings of the 3rd International IFIP/ACM Latin American Conference on Networking*, LANC '05, pages 1–9, New York, NY, USA. ACM.
- Rose, K., Eldridge, S., and Chapin, L. (2015). The Internet of Things: an overview. *The Internet Society (ISOC)*, pages 1–50.
- Silva, J. R., Delicato, F. C., Pirmez, L., Pires, P. F., Portocarrero, J. M., Rodrigues, T. C., and Batista, T. V. (2014). PRISMA: A publish-subscribe and resource-oriented middleware for wireless sensor networks. In *Proceedings of the Tenth Advanced International Conference on Telecommunications, Paris, France*, pages 87–97.
- Souto, E., Guimarães, G., Vasconcelos, G., Vieira, M., Rosa, N., Ferraz, C., and Kelner, J. (2006). Mires: a publish/subscribe middleware for sensor networks. *Personal and Ubiquitous Computing*, 10(1):37–44.
- Verma, P. K., Verma, R., Prakash, A., Agrawal, A., Naik, K., Tripathi, R., Alsabaan, M., Khalifa, T., Abdelkader, T., and Abogharaf, A. (2016). Machine-to-machine (M2M) communications: A survey. *Journal of Network and Computer Applications*, 66:83–105.
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1):22–32.