

Universidade Federal de Santa Catarina  
Centro Tecnológico  
Departamento de Informática e Estatística  
INE5424 - Sistemas Operacionais II  
Professor: Rafael Luiz Cancian  
Estagiário de Docência: César Huegel Richa  
Alunos: Alisson Granemann Abreu  
Nathan Molinari  
Willian Santos de Souza

11100854  
13100769  
11100901

## **TSTP IoT Fog Interest Message Admission Control**

### **Motivação**

O atraso em troca de mensagens em sistemas embarcados com aplicações em tempo real é um fator prioritário e deve ser mensurável, para a predição do custo de todas as interações entre seus componentes, para agir ou não, dependendo do tipo de aplicação e seus objetivos. No caso específico deste trabalho, é de interesse obter o impacto ao adicionar um novo recurso ou dispositivo na fog, para avaliar se este é aceitável dentro do limite da rede, além de tomar devidas ações e/ou assegurar que certos requisitos de QoS serão atendidos.

### **Objetivos**

O objetivo deste projeto é prover um mecanismo que estime o impacto na rede ao aceitar novos interesses requeridos em um intervalo de tempo (Time-Triggered) em certa região ou novos dispositivos dentro da fog IoT. Para calcular a carga na rede ao admitir um novo interesse, é preciso descobrir quantos hops serão necessários para transmitir essa informação até o sink(gateway), dado um duty cycle global pré-definido (usualmente 1%), além de checar o impacto de interesses aceitos previamente.

### **Metodologia**

Após estudo e entendimento do problema, será feita a análise de requisitos em conjunto com o cliente (LISHA). O software será projetado utilizando os padrões de projetos apresentados em aula. Os testes serão realizados primeiramente em um simulador para então serem aplicados em ambiente real com sistemas embarcados EPOSMote III.

## Tarefas

1. Elaborar o plano detalhado do projeto
2. Estudo, entendimento do problema e modelagem da solução do problema
  1. Modelar uma abordagem para calcular quantos hop's são necessários para a entrega da mensagem
  2. Construir a função que calcula o tempo total da entrega da mensagem até o sink
  3. Modelar uma solução para calcular a carga total de um novo interesse, juntamente com interesses antigos
  4. Estudar possíveis cenários em que outros fatores podem influenciar o cálculo da carga na rede
  5. Modelar a solução para o impacto da adição de um novo dispositivo na rede
3. Implementação dos requisitos
  1. Adição de interesses (Interest)
  2. Adição de dispositivos IoT
4. Testes em ambiente real
5. Entrega final

## Entregáveis

1. Plano detalhado do projeto
2. Relatório com a modelagem da solução do problema
3. Implementação requisito “Adição de interesses (Interest)”
4. Implementação requisito “Adição de dispositivos IoT”
5. Resultados dos testes
6. Entrega final contendo o software e relatório completo

Entregáveis	Tarefas	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
1. Plano detalhado do projeto	1. Elaborar o plano detalhado do projeto.	E1									
2. Relatório com a modelagem da solução do problema.	2. Estudo, entendimento do problema e modelagem da solução do problema.  2.1. Modelar uma abordagem para calcular quantos hop's são necessários para a entrega da		x	x	x	E2					

	mensagem.  2.2. Construir a função que calcula o tempo total da entrega da mensagem até o sink.										
3.Implementação requisito “Adição de interesses”	3. Implementação dos requisitos:  3.1. Adição de interesses						x	x	x	x	E3
4.Implementação requisito “Adição de dispositivos IoT”.	3. Implementação dos requisitos:  3.2. Adição de dispositivos IoT							x	x	x	E4
5. Resultados dos testes	5.1 Testes em ambiente real  5.2 Análise dos resultados									x	E5
6. Entrega final contendo o software e relatório completo	6. Entrega final										E6

## Tasks

Em uma Wireless Sensor Network (WSN) operando em conjunto com o TSTP, no cenário em que são adicionados muitos interesses, há a possibilidade de uma sobrecarga na rede. Como no protocolo TSTP, apenas uma mensagem é transmitida por nodo, a cada período de tempo [2], se um nodo receber mais mensagens do que consegue enviar, ficará sobrecarregado. Esse comportamento pode refletir em outros nodos da rede causando mau funcionamento na mesma.

Para evitar a sobrecarga, é proposto um Controle de Admissão de Interesses (CAI), que conhece a carga da rede e é capaz de aceitar ou rejeitar um novo interesse. Este módulo será implementado no gateway, já que as requisições de novos interesses são recebidas por este. Para que o CAI tenha conhecimento da carga da rede, será implementado um módulo

chamado Estimador de Carga (EC), que terá um mapa da rede, os interesses admitidos e será capaz de estimar a carga atual da rede.

Cada nodo, ao inicializar, enviará uma mensagem de resposta para o gateway contendo suas coordenadas. Ao recebê-la, o gateway mandará para o EC a localização do nodo, que por fim conhecerá a localização de todos os sensores da rede. Com todas as coordenadas armazenadas, o EC construirá um grafo em que cada nodo é a localização global do sensor e as arestas irão representar a distância entre os nodos. Para geração do grafo será utilizado o parâmetro *RADIO\_RANGE*, que representa o alcance de cada sensor. Uma nova aresta é criada se a distância entre um nodo A e um nodo B é menor ou igual ao *RADIO\_RANGE*.

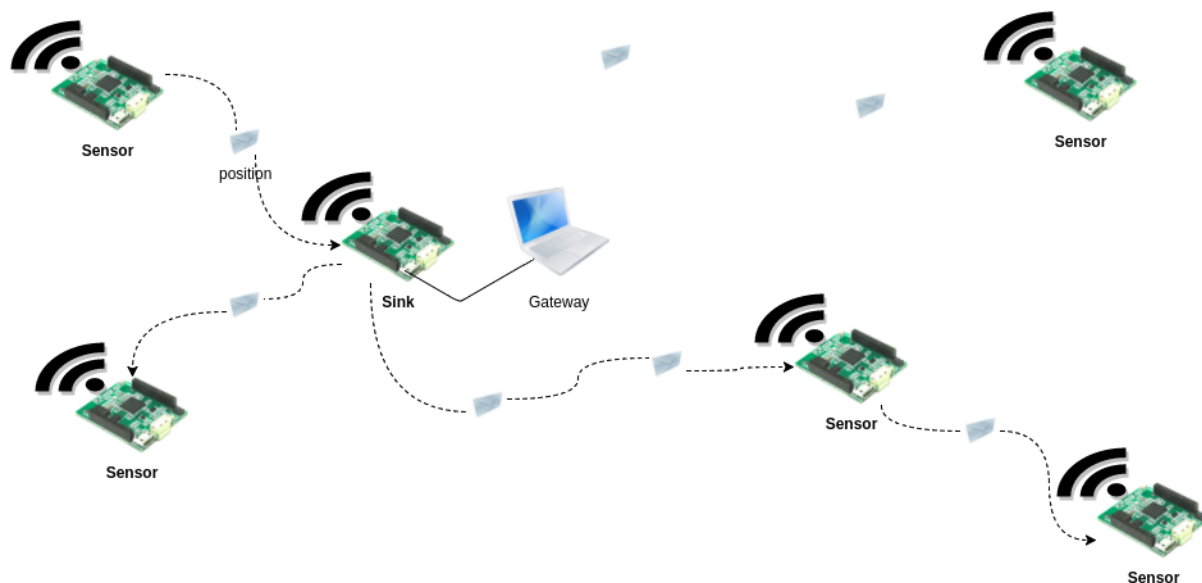
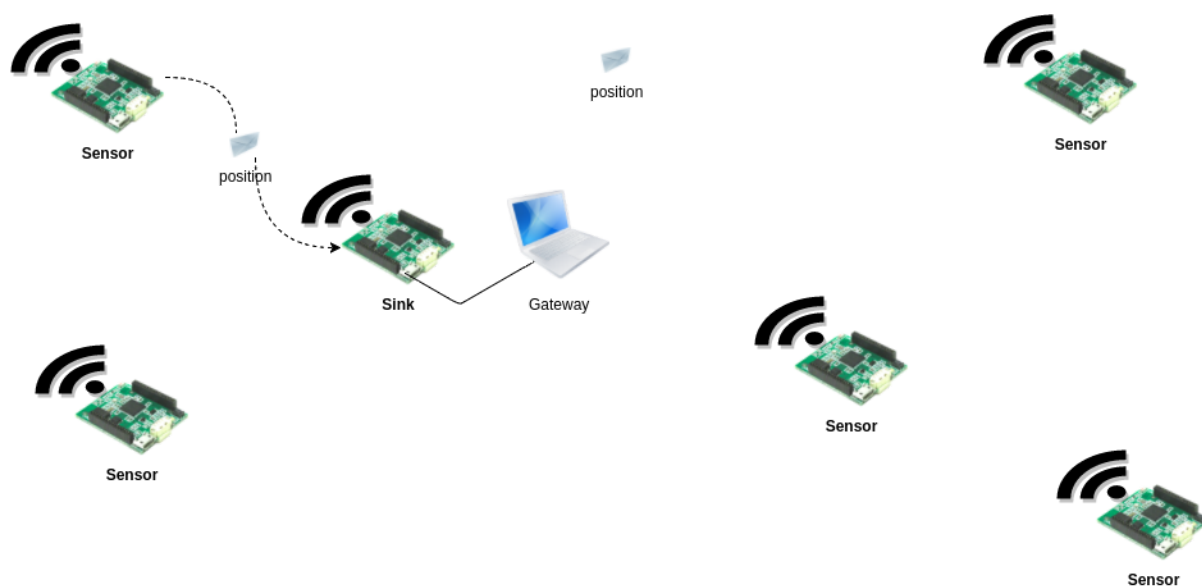


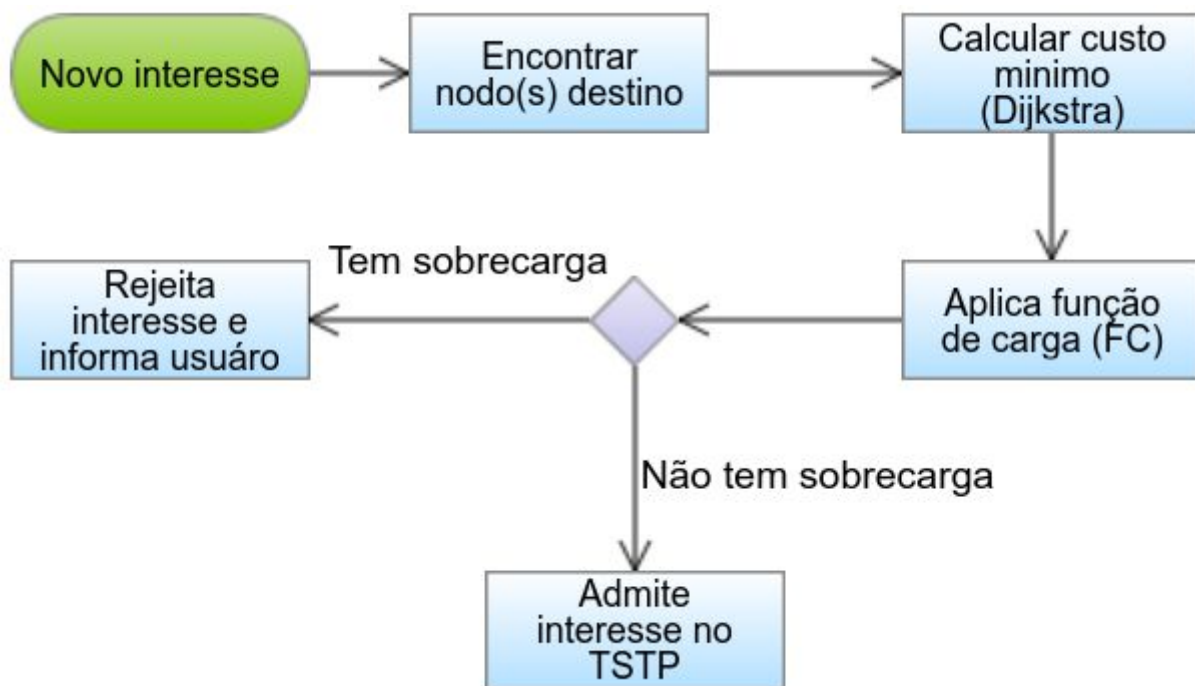
Figura 1 - Rede Iot



**Figura 2 - Boot de um nodo da rede**

A transmissão de mensagens no TSTP sempre é feita pelo nodo mais próximo do destino [1], ou seja, o caminho percorrido pela mensagem será mínimo. Ao receber uma nova mensagem de interesse, o EC irá encontrar o(s) nodo(s) contido(s) na região desejada pelo interesse. Com este(s) nodo(s) determinado(s), podemos calcular o caminho de custo mínimo do gateway até o destino. Esse cálculo será feito utilizando o algoritmo de Dijkstra.

Sabendo o caminho do gateway até o destino do interesse, é possível calcular a quantidade de hops, que será útil para o cálculo da carga da rede. Este cálculo, porém, necessita de mais informações, tais como, período do MAC e a lista de interesses já admitidos. A lista de interesses ficará no EC e como será considerado apenas interesses modo Time Triggered, sabemos a frequência do envio de cada resposta da região destino. Com esses parâmetros o EC será capaz de estimar a carga da rede. Baseado no resultado, o gateway admite ou não o interesse, dependendo da carga Figura 3. Ao rejeitá-la, o gateway enviará uma mensagem para quem o requisitou, informando a rejeição.



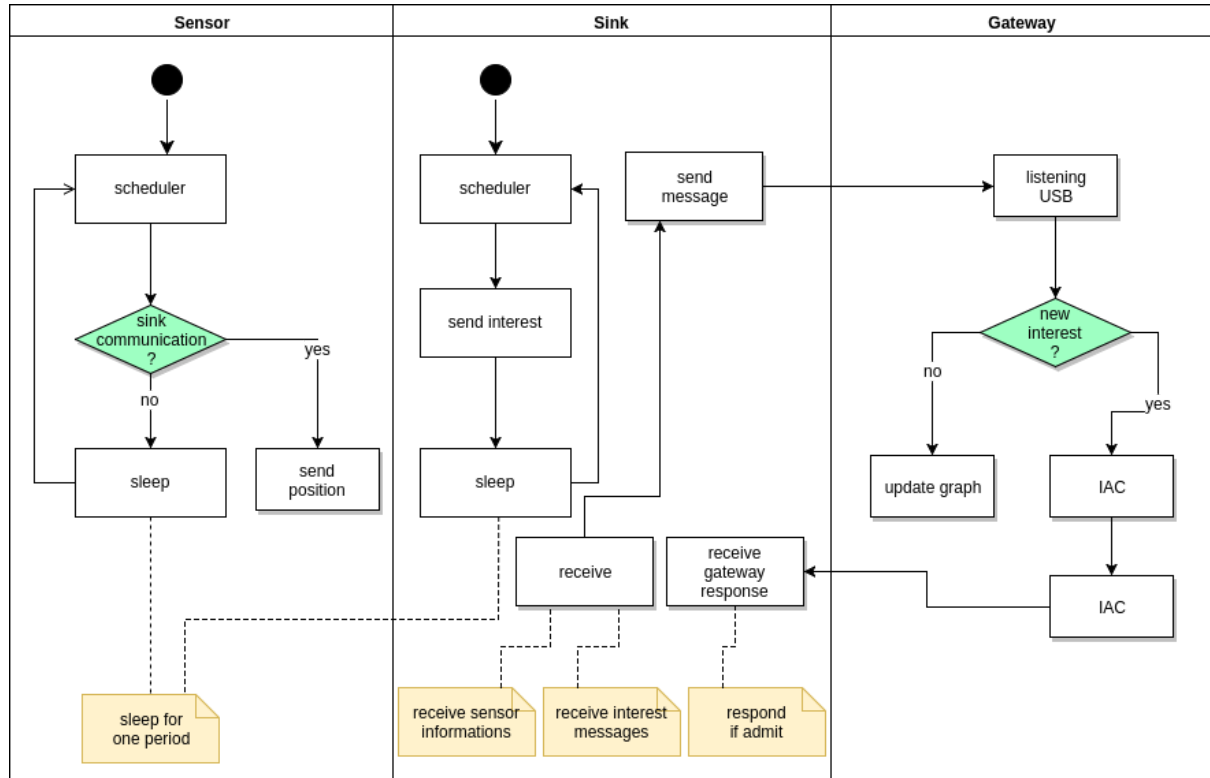
**Figura 3 - Fluxo da admissão de um novo interesse**

As seguintes premissas foram feitas para realização deste trabalho:

- A mensagem enviada ao inicializar os sensores, a qual tem como objetivo enviar sua localização para o gateway, sempre será recebida pelo gateway. Ou seja o gateway irá conhecer a localização de todos os sensores.
- Não será considerado perda de interesses pelos sensores e perdas do grafo e lista de interesses pelo gateway devido a quedas de energia ou desligamento de dispositivos.

- O caminho percorrido pela mensagem de resposta de um nodo até o gateway sempre será o mínimo, ou seja, a menor distância.
- Serão considerados na medição do CI apenas interesses Time Triggered
- Serão ignoradas falhas eventuais que podem causar retransmissões

## Desenvolvimento



**Figura 4 - Fluxo de comunicação**

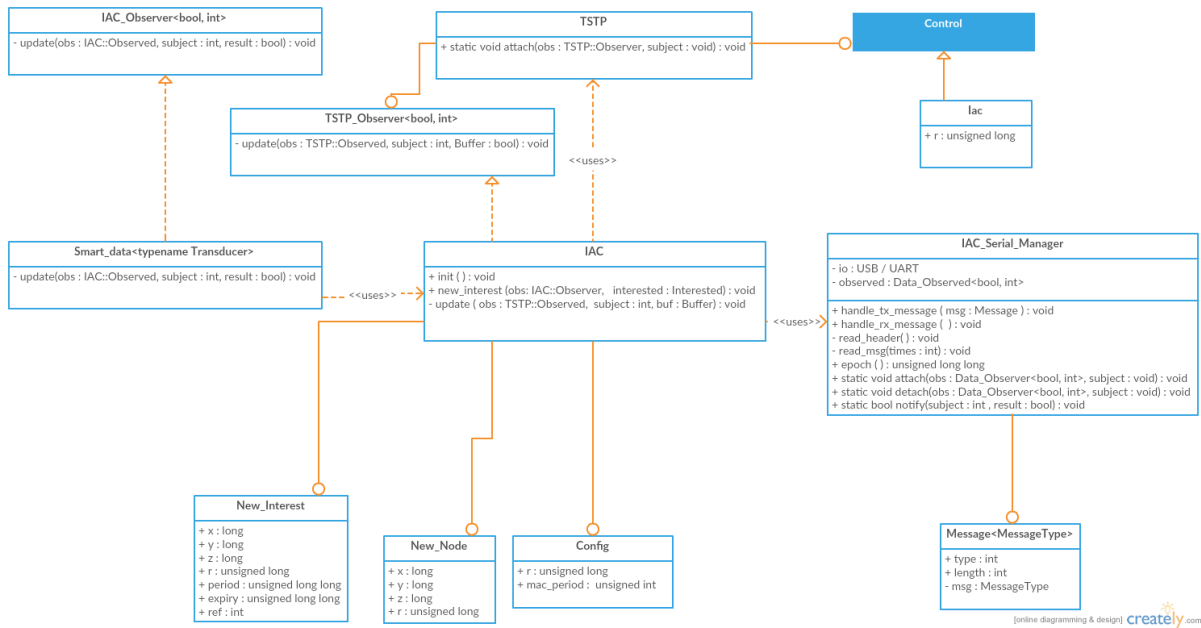


Figura 5 - Diagrama de classes no epos

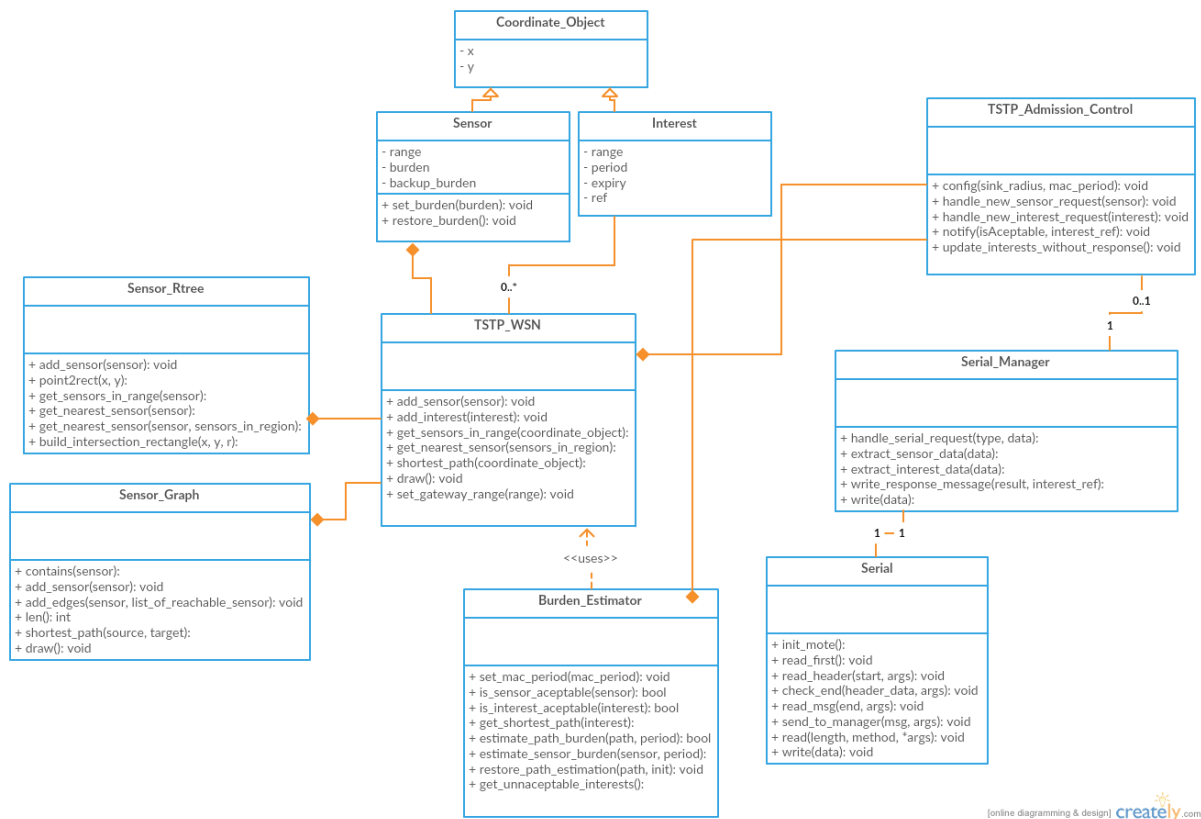


Figura 6 - Diagrama de classes no gateway

## Conclusões

## Bibliografia

1. Davi Resner and Antônio Augusto Fröhlich, TSTP MAC: a Cross-Layer, Geographic, Receiver-Based MAC Protocol for WSNs, In: Proceedings of the Brazilian Symposium on Computing Systems Engineering, Foz do Iguaçu, Brazil, November 2015.
2. Davi Resner and Antônio Augusto Fröhlich, TSTP MAC: A Foundation for the Trustful Space-Time Protocol, In: Proceedings of the 14th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC 2016)., Paris, France, September 2016.
3. C. Y. Chuang, Y. C. Chen and C. W. Hsueh, "Scheduling Low-Utilized Real-Time Systems with End-to-End Timing Constraints," 2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Daegu, 2016, pp. 98-98.