

GrADyS-SIM NextGen

Relatório Técnico

Objetivos Gerais

O software desenvolvido se enquadra na área de Redes e Sistemas Distribuídos. O programa possibilita que usuários implementem simulações populadas por nós capazes de comunicação e movimentação. Esses protocolos são controlados por um componente que nesse trabalho é chamado de “protocolo”. Trata-se de um componente que implementa a programação lógica de como o nó deve se comportar e interagir com o ambiente por meio de movimentação e comunicação com os outros nós. O foco da implementação é são algoritmos distribuídos pelos quais os nós atingem algum objetivo.

O programa se destina à pesquisadores da área Sistemas Distribuídos que estejam interessados em validar seus algoritmos em um ambiente simulado. Nós temos usuários garantidos dentro do próprio projeto GrADyS, dentro do qual esse software foi criado, mas o software foi projetado para uso geral e não apenas interno. A intenção é que ele seja usado no longo prazo para as simulações do LAC, por quem será mantido, e para usuários gerais externos ao laboratório.

As principais características que tornam esse software valioso são a definição de um modelo de implementação de algoritmos distribuídos figurado pelos chamados “protocolos” e a disponibilidade de diversos ambientes de execução onde esses protocolos podem rodar.

A ideia inicial de definir uma interface comum para criação de protocolos foi centralizar todo o trabalho do laboratório em uma linguagem e implementação. Isso nos permitiria construir todo um ferramental para facilitar a implementação desses algoritmos e facilitaria a leitura e aprendizado pelos usuários. Além da interface de simulação distribuímos também uma bibliotecas de “plugins” que são ferramentas que auxiliam na criação de protocolos.

Os protocolos executam em chamados “ambientes de execução”. Uma segunda implementação de valor desse software foi tornar os protocolos completamente independentes do ambiente de execução onde estão rodando. Eles não são expostos a nenhuma dependência do ambiente e, portanto, podem ser executados em qualquer um sem nenhuma mudança de código.

Os ambientes de execução em si também são implementações de valor trazidas por esse software. Fornecemos são dois, com a intenção de se adicionar um terceiro no futuro. O primeiro é chamado de “prototype-mode”, nesse modo de execução os protocolos serão rodados em um simulador escrito em python que foi criado especialmente para esse projeto. Trata-se de um ambiente que modela situações reais com um grau de simplificação grande. A vantagem dele é que ele é extremamente fácil de ser usado.

Simuladores da área, inclusive a nossa versão anterior do simulador, tendem a ter uma curva de aprendizado íngreme e serem difíceis de instalar. A intenção com esse primeiro ambiente foi remediar. O segundo ambiente de execução nada mais é do que nosso simulador antigo. Ele é implementado em um software de simulação de redes chamado OMNeT++. Ele produz um modelo bem realista de todos os aspectos de rede de uma simulação. Sua desvantagem é a dificuldade de uso e instalação. O simulador antigo foi integrado com a nova interface de

protocolos e agora deixou de ser um requisito para nosso framework de simulação, tornando-se uma dependência opcional para aqueles que buscam realismo.

Nossos objetivos futuros incluem um terceiro ambiente de execução: o mundo real. Fizemos nossa interface geral de criação de protocolos com a intenção de usá-la para programar veículos reais, a simulação é apenas uma etapa anterior de validação e verificação com custo muito inferior aos testes de campo. Essa integração com o mundo real ainda não foi feita, mas deve ser adicionada no curto/médio prazo.

Requisitos funcionais e não funcionais

Uma descrição mais natural desses requisitos pode ser encontrada no manual:

<https://thlamz.github.io/GrADyS-SIM-Nextgen/#what-should-i-expect-from-gradys-sim-nextgen>

Funcionais

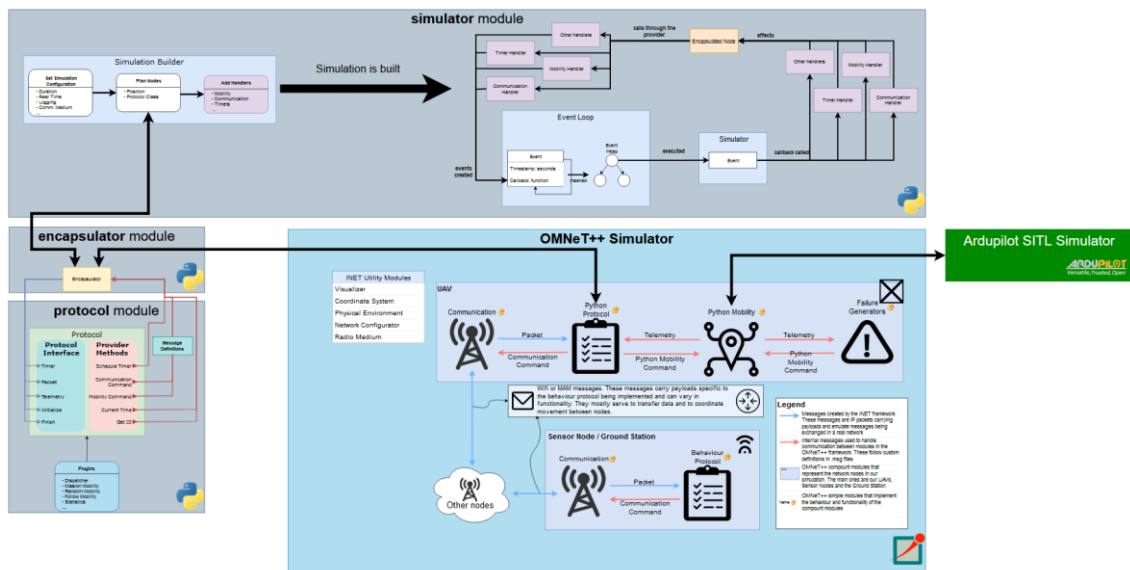
1. O sistema deve permitir a implementação de “protocolos”, componentes que descrevem uma lógica de comportamento de um nó com capacidades de comunicação e movimentação
2. Os protocolos devem poder ser executados em qualquer ambiente de execução, sem mudanças de código
3. Ferramentas para auxiliar na criação de protocolos (plugins) devem ser criadas
4. Um simulador Python deve ser fornecido para prototipação de protocolos
5. O simulador antigo escrito no OMNeT++ deve ser integrado com o novo framework de simulação

Não funcionais

1. O framework de simulação deve ter uma curva de aprendizado suave
2. O simulador Python e interface de protocolos deve ser fácil de utilizar
3. A interface de implementação de protocolos deve ser intuitiva para seu caso de uso
4. Os protocolos devem ser reutilizáveis
5. O framework, pelo menos a interface de protocolo e simulador em Python, deve ser leve em dependências
6. O framework, pelo menos a interface de protocolo e simulador em Python, deve ser fácil de instalar

Descrição da arquitetura do software

A arquitetura está disponível dentro do manual na seção <https://thlamz.github.io/GrADyS-SIM-Nextgen/framework/>. Ela é baseada no seguinte diagrama:



Descrição ou modelo funcional do software

Essa informação pode ser encontrada em diversas partes do manual. Segue algumas seções relevantes:

- <https://thlamz.github.io/GrADyS-SIM-Nextgen/Getting%20Started/execution/>
- <https://thlamz.github.io/GrADyS-SIM-Nextgen/Guides/creating/>
- <https://thlamz.github.io/GrADyS-SIM-Nextgen/Modules/Protocol/>
- <https://thlamz.github.io/GrADyS-SIM-Nextgen/Modules/Simulator/>

As páginas acima contêm explicações tanto naturais como mais técnicas, focadas no código da aplicação.

Manual e Utilização

O manual completo está disponível em <https://thlamz.github.io/GrADyS-SIM-Nextgen>. Ele contempla os casos de uso, tutorial de instalação, possui alguns guias e bastante material informativo sobre o framework.

Casos de uso de sucesso

Efeitos do ambiente em um algoritmo distribuído

Esse caso de uso descreve um pesquisador da área de redes e sistemas distribuídos que busca analisar os efeitos de diferentes aspectos de rede e de mobilidade no funcionamento do seu algoritmo. A sua análise será centralizada na análise do recebimento e envio de pacotes em diferentes condições.

Nesse contexto ele utilizará o GrADyS-SIM NextGen para avaliar um mesmo protocolo em diferentes condições. Ele variará o número de nós que compõe o protocolo e as condições as quais eles estão expostos. Para isso utilizará a interface de protocolo para criar seus protocolos e o simulador Python para suas simulações. Em específico utilizará diversas funcionalidades do simulador Python para variar esses aspectos de rede e validar seus resultados.

Após instalar o software seguindo [essa seção do manual](#) ele irá implementar sua solução. Seu passo-a-passo está descrito como um guia no manual [nesta seção](#). O guia não só descreve o

percurso que seguiu como dá informações importantes de como o simulador funciona, as noções lógicas por trás dos protocolos e serve como exibição das diferentes funcionalidades do programa.

Implementação de um cenário de coleta de dados

Um pesquisador do LAC da área de redes e sistemas distribuídos tem um cenário em mente: ele quer coletar dados de sensores dispersos em um campo. Esses sensores constantemente coletam informações sobre seu ambiente local que são de interesse para ele. O campo onde os sensores estão não possui nenhuma infraestrutura de rede e é de difícil acesso então ele irá empregar um enxame de drones para fazer a coleta e trazer os dados para uma estação de controle, onde o pesquisador ficará.

O pesquisador busca implementar a lógica que irá mover esses veículos assim como controlar os sensores e a estação de controle. Ele implementará toda essa lógica na forma de protocolos e, para validar sua ideia e a corretude dos protocolos, utilizará o simulador Python para simular o cenário.

Após instalar o software seguindo [essa seção do manual](#) ele irá implementar sua solução. O processo que ele passou para traduzir esse cenário em uma simulação e para implementar a lógica dos protocolos está disponível como guia [nesta seção do manual](#).

Casos de uso problemáticos

Simulação no mundo real

O mesmo pesquisador descrito [nesta seção anterior](#) agora busca usar os protocolos que criou no mundo real. Ele não conseguirá fazer isso no momento, visto que o código que permite que os protocolos interajam com o hardware presente no mundo real não existe ainda. É uma implementação que está projetada para o futuro próximo do projeto mas no momento não existe.

O pesquisador teria que por conta própria implementar o código de “cola” que serve para permitir que o protocolo rode em um ambiente novo, no caso o mundo real com hardware de verdade.

Simulação mais realista

Após implementar sua simulação em Python o pesquisador descrito [nesta seção anterior](#) busca expandir seu estudo com uma simulação de rede mais realista. Ele percebe que o GrADyS-SIM NextGen fornece um ambiente chamado “integrated-mode” onde ele pode executar seu protocolo em um ambiente de rede realista no simulador OMNeT++.

Um problema que ele terá é que o simulador do GrADyS implementado no OMNeT++ possui uma curva de aprendizado bem íngreme e o processo de instalação do OMNeT++ é bastante complicado. Para minimizar essas dificuldades implementamos a integração que permite que esse pesquisador reaproveite seu código python sem nenhuma modificação. Ele apenas terá que montar o ambiente de simulação dentro do OMNeT++, sem se preocupar com a lógica dos nós. Além disso melhoramos as instruções de instalação e uso do nosso ambiente de simulação no OMNeT++, a documentação está disponível em: <https://brunoolivieri.github.io/gradys-simulations/>.