

Guia de uso do Simulador Avançado TTN

1. Introdução

O Simulador Avançado TTN é uma poderosa aplicação de linha de comando em Python, desenhada para envolver e expandir a funcionalidade da Interface de Linha de Comando (CLI) oficial do The Things Stack (`ttn-lw-cli`). Ele fornece uma *shell* interativa para simplificar o processo de simulação de *uplinks* de dispositivos LoRaWAN® e outros eventos, tornando-se uma ferramenta inestimável para programadores que testam a lógica da aplicação, *payload formatters* e integrações sem a necessidade de *hardware* físico.

Principais Funcionalidades:

- **Shell Interativa:** Proporciona um ambiente amigável para configurar e executar simulações sem ter de correr o *script* novamente para cada comando.
- **Orientado por Ficheiros de Configuração:** Use ficheiros JSON para definir cenários de simulação complexos, incluindo o tipo de simulação, o alvo (*target*), *flags* e regras de geração de *payload*.
- **Geração Avançada de *Payload*:**
 - Gere *payloads* aleatórios simples ou fixos.
 - Use um *template* JSON estruturado para definir formatos de *payload* binários complexos com dados aleatórios para cada campo, incluindo o empacotamento de bytes (*byte packing*) e *endianness* corretos.
- **Múltiplos Modos de Simulação:**
 - **Modo Manual:** Execute simulações únicas (*one-shot*) para testar cenários específicos.
 - **Modo Periódico:** Execute simulações em *background* a um intervalo configurável para imitar um dispositivo real a enviar dados ao longo do tempo.
- **Flexível e Extensível:** Embora focado em `application-uplink`, a estrutura foi desenhada para executar qualquer subcomando `ttn-lw-cli simulate`, bastando defini-lo num ficheiro de configuração.

2. Pré-requisitos

Antes de começar, certifique-se de que tem o seguinte instalado e configurado no seu sistema (ex: Ubuntu, macOS, WSL no Windows):

1. **Python 3:** O *script* foi escrito em Python 3.

2. **The Things Stack CLI (`ttn-lw-cli`)**: O simulador depende totalmente desta ferramenta. Deve estar instalada e acessível através do `PATH` do seu sistema.
 - [Guia Oficial de Instalação](#)
3. **Conta TTN com Login Ativo**: Deve ter sessão iniciada na sua conta The Things Stack através da CLI. Se não estiver, execute `ttn-lw-cli login` e siga as instruções antes de usar este simulador.

3. Estrutura de Arquivos Explicada

Irá trabalhar com três arquivos principais:

Nome do Ficheiro	Propósito
<code>ttn_simulator.py</code>	O Núcleo da Aplicação. Este é o <i>script</i> Python que irá executar para lançar a <i>shell</i> interativa. Contém toda a lógica para processar comandos, interagir com o <code>ttn-lw-cli</code> e gerir o estado da simulação.
<code>simulation_config.json</code>	O Arquivo de Definição da Simulação. Este é um arquivo de configuração de alto nível que cria. Define que tipo de simulação executar, o seu alvo, as suas <i>flags</i> e como o seu <i>payload</i> deve ser gerado. Pode ter múltiplos ficheiros de configuração para diferentes cenários.
<code>payload_template.json</code>	O Arquivo de Estrutura do Payload. Este ficheiro opcional é referenciado pelo seu <code>simulation_config.json</code> . Fornece um projeto detalhado e de baixo nível para criar um <i>payload</i> binário estruturado com múltiplos campos, tipos de dados e regras de aleatorização.

4. Como Começar: Um Tutorial Passo a Passo

Este tutorial irá guiá-lo através da sua primeira simulação periódica.

Passo 1: Criar o Template do Payload (`payload_template.json`)

Este ficheiro define a estrutura do pacote de dados do seu sensor. Crie um ficheiro chamado `payload_template.json` com o seguinte conteúdo.

```
{
  "_comment": "Define a estrutura dos dados do nosso sensor.",
  "_field_order": ["temperature", "humidity", "status", "battery_voltage"],
  "fields": {
    "temperature": {
      "type": "float",
      "min": -10.0,
```

```

    "max": 40.0,
    "precision": 1,
    "packer": "f",
    "byte_order": "big"
  },
  "humidity": {
    "type": "uint",
    "min": 0,
    "max": 100,
    "packer": "B"
  },
  "status": {
    "type": "choice",
    "values": {"active": 0, "sleep": 1, "error": 2},
    "packer": "B"
  },
  "battery_voltage": {
    "type": "uint",
    "min": 2800,
    "max": 4200,
    "packer": "H",
    "byte_order": "big"
  }
}
}
}

```

Passo 2: Criar a Configuração da Simulação (`my_periodic_sim.json`)

Este ficheiro orquestra a simulação. Crie um ficheiro chamado `my_periodic_sim.json`.

```

{
  "simulation_type": "application-uplink",
  "description": "Uma simulação periódica a enviar dados de sensor a cada 15 segundos.",
  "periodic_settings": {
    "interval": 15,
    "enabled_on_load": false
  },
  "common_args": ["your-app-id", "your-device-id"],
  "flags": {
    "f-port": 15,
    "confirmed": false,

```

```
"settings.data-rate-index": 3,  
"settings.frequency": "868300000"  
},  
"payload_source": {  
  "type": "json_template",  
  "file": "payload_template.json"  
}  
}
```

IMPORTANTE: Substitua `"your-app-id"` e `"your-device-id"` pelo seu Application ID e Device ID reais do The Things Stack.

Passo 3: Executar o Simulador e Carregar a Configuração

1. Abra o seu terminal no diretório onde os seus ficheiros estão localizados.
2. Execute o *script* Python:

```
python ttn_simulator.py
```

3. Verá a mensagem de boas-vindas e o *prompt*: `(adv-ttn-sim)` .
4. Carregue a sua configuração de simulação:

```
(adv-ttn-sim) load_sim_config my_periodic_sim.json
```

A aplicação irá confirmar que o ficheiro foi carregado e que o intervalo periódico foi definido para 15 segundos.

Passo 4: Iniciar e Parar a Simulação Periódica

1. Para começar a enviar dados a cada 15 segundos, escreva:

```
(adv-ttn-sim) start_periodic_sim
```

O simulador irá agora correr em *background*, imprimindo os detalhes de cada *uplink* simulado à medida que acontece. Pode continuar a escrever outros comandos enquanto ele corre.

2. Para parar a simulação, escreva:

```
(adv-ttn-sim) stop_periodic_sim
```

5. Detalhes de Configuração Aprofundados

Os detalhes de configuração dos arquivos JSON estão disponíveis no outro pdf chamado “Guia: Como Criar JSONs de Configuração do Simulador”

6. Referência de Comandos

Comando	Descrição
<code>load_sim_config <file></code>	Carrega um cenário de simulação a partir do ficheiro JSON especificado.
<code>simulate</code>	Executa uma única simulação com base na configuração atual.
<code>start_periodic_sim</code>	Começa a enviar <i>uplinks</i> em <i>background</i> no intervalo configurado.
<code>stop_periodic_sim</code>	Para a simulação periódica em execução.
<code>config_periodic ...</code>	Configura interativamente o intervalo periódico (ex: <code>config_periodic interval=10</code>).
<code>config_payload ...</code>	Configura interativamente o método de geração de <i>payload</i> .
<code>config_sim_flags ...</code>	Configura interativamente as <i>flags</i> de simulação (usado principalmente se nenhum ficheiro de configuração for carregado).
<code>view_config</code>	Exibe todas as configurações atuais (simulação carregada, <i>flags</i> interativas, <i>payload</i> e configurações periódicas).
<code>list_apps</code>	Lista todas as aplicações a que o seu utilizador tem acesso.
<code>list_devices [app_id]</code>	Lista todos os dispositivos numa aplicação específica.
<code>quick_setup</code>	Um assistente interativo para selecionar a primeira aplicação e dispositivo disponíveis.
<code>set_target <app> <dev></code>	Um assistente interativo para definir manualmente a aplicação e o dispositivo alvo.
<code>set_cli_path <path></code>	Define o caminho para o executável <code>ttn-lw-cli</code> se não estiver no <code>PATH</code> do seu sistema.
<code>exit</code> / <code>Ctrl+D</code>	Sai da <i>shell</i> do simulador.

7. Informações Adicionais

Esta ferramenta fornece uma estrutura robusta para testes. Para simulações avançadas (ex: `gateway-forward` , `lorawan-uplink`), precisará de compreender as *flags* específicas exigidas pelo `ttn-lw-cli` , que podem ser encontradas na documentação oficial. A secção de *flags*

do seu ficheiro de configuração é poderosa o suficiente para lidar com qualquer *flag* que o `ttn-lw-cli` aceite.

Links Úteis:

- **Documentação da CLI do The Things Stack:**
<https://www.thethingsindustries.com/docs/concepts/features/cli/>
- **Referência do Comando `ttn-lw-cli simulate` :**
https://www.thethingsindustries.com/docs/ttn-lw-cli/ttn-lw-cli_simulate/
- **Módulo `struct` do Python (para formatos de *packer*):**
<https://docs.python.org/3/library/struct.html>