

C 言語の通信ライブラリを呼び出す Python ラッパーの提案

渡邊 憲士^{†*}, 清水 一輝[†], 鈴木 秀和[†], 内藤 克浩[‡], 渡邊 晃[†]

([†] 名城大学, [‡] 愛知工業大学)

Proposal for Python Wrapper that calls C-language Communication Library

Kenshi Watanabe[†], Kazuki Shimizu[†], Hidekazu Suzuki[†], Kastsuhiro Naito[‡], Akira Watanabe[†]

([†]Meijo University, [‡]Aichi Institute of Technology)

1 はじめに

大規模なプログラミングにおいて、ライブラリは処理速度や移植性、多言語との連携といった観点から、C 言語で実装されることが多い。それに対して、アプリケーションは Python などの抽象度の高い高級言語にて開発されることが一般的である。そのため、アプリケーションが C 言語ライブラリを使用する場合は、一般的に呼び出し元の言語に応じたラッパーを作成して、ラッパーを経由して C 言語ライブラリを使用する。このとき、アプリケーション側はライブラリの C 言語を意識する必要があるという課題がある。しかし、通信ライブラリの場合はこれを不要にできると考えた。本稿では通信ライブラリを対象として、開発者が C 言語を意識しないで済む Python ラッパーの実現方法について提案する。

2 既存のラッパー

ラッパーを介して C 言語ライブラリを使用する際には、呼び出し元に新たな機能を提供することが多いため、アプリケーションは C 言語ライブラリの API を意識する必要がある。すなわち、C 言語ライブラリを使用したことのない開発者が C 言語ライブラリを意識する必要があるため、開発負荷が大きくなる。そのため、通信 API のように呼び出し元の標準 API に基本機能がすでに用意されている場合は、呼び出し元の標準 API と同じ方法でライブラリを使用することが望ましい。

3 通信ライブラリ用 Python ラッパー

通信ライブラリについて、呼び出し元のプログラミング言語標準の API と同じ方法で利用できるようにする方法を検討した。検討対象は Python アプリケーションから NTMobile フレームワークライブラリを呼び出す場合である。

<3・1>NTMobile フレームワークライブラリ NTMobile フレームワークライブラリ（以下 NTMfw）は NTMobile をアプリケーションで実装する C 言語の通信ライブラリである。NTMobile[1] はネットワーク環境に関わらず通信を開始することができる通信接続性とネットワークが切り替わった際にも通信を継続できる移動透過性を実現する技術である。NTMfw を呼び出すための API はカーネルの通信ライブラリを呼び出す API と機能互換を持ち、名称が異なるという関係がある。

<3・2>Python ラッパーの構成 Fig. 1 に Python ラッパーのモジュール構成図を示す。提案方式では既存方式による API に加え、NTM ソケットクラスを追加した。提案方式では NTM ソケットクラスを経由して通信を行う。

NTM API インプリクラスは一般のラッパーでも必要となる機能で、C 言語と Python の違いの除去を行う。両者の違いは型名の違いであったり、同等の機能をもつ API の引数に関する

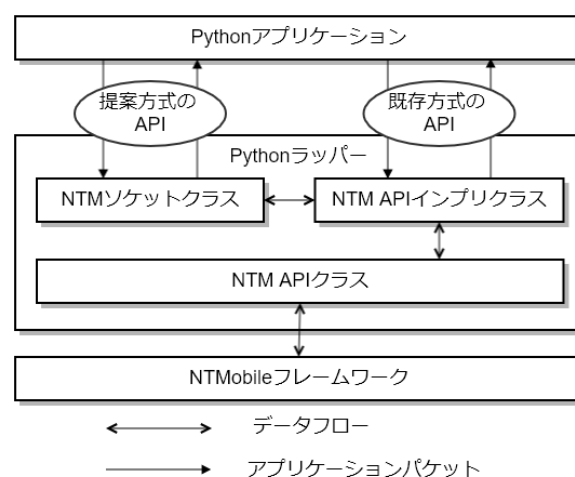


Fig. 1 Structure of Python Wrapper

違いである。型の違いは、型のサイズに応じて型名を変更することで対応可能である。また、API の引数の違いは、引数にて得られたデータを C 言語の API の引数に合うように細分化し、渡すことで対応可能である。マッピングには Python 標準の外部関数ライブラリである ctypes を利用する。

NTM ソケットクラスは Python 標準ソケットを継承して、NTMobile 用に再定義した Python ソケット API 互換のソケット API である。Python 標準 API を使用して自動的に NTMobile 通信を実現できるように、Python 標準の通信クラスのメソッドをオーバーライドする。これにより Python アプリケーションが Python の通信に関するメソッドを呼び出した際に、Python ラッパーによって NTMfw の関数を自動的に呼び出すようになる。

NTMfw 起動時には初期化処理が必要である。初期化処理は NTMfw 特有の処理であるため、Python 標準 API をオーバーライドすることができない。そこで、初期化処理のメソッドは NTM API インプリクラスに定義する。

4 まとめ

本稿では、通信ライブラリ用 Python ラッパーを実現する方法を検討した。今後は、検討方式の実装および性能評価を行う予定である。

文 献

[1] 上醉尾. 他: IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価 情報処理学会論文誌 Vol.54, No.10, pp.2288–2299, 2013.

C言語の通信ライブラリを呼び出す Pythonラッパーの提案

渡邊憲士† 清水一輝† 鈴木秀和† 内藤克浩‡ 渡邊晃†
†名城大学 理工学部
‡愛知工業大学 情報学部



研究背景

- ▶ 大規模な開発においてライブラリはC言語で実装され、アプリケーションは抽象度の高い高級言語で実装される
- ▶ ラッパー
 - あるプログラミング言語にて実装された機能を他の言語から利用できるようにするもの

研究背景

- ▶ C言語によるライブラリの実装
 - 実行時における処理速度
 - ハードウェアを直接制御できる
 - クロスプラットフォーム
 - 標準化されたインターフェースがある

研究背景

▶ プログラミング言語ランキング

順位	2018年	2017年	2016年
1位	Python	Python	C
2位	C++	C	Java
3位	C	Java	Python
4位	Java	C++	C++
5位	C#	C#	PHP

研究背景

▶ Pythonが人気の理由

- 豊富なライブラリ
 - 専門的なライブラリを利用した効率的な開発
- シンプルでわかりやすい文法
 - 読みやすく効率のよいコードを簡単に書くことができる

既存技術

▶ ctypes

- PythonからC言語を利用するための外部関数ライブラリ
- C言語と互換性のあるデータ型を提供し、共有ライブラリ内の関数呼び出しを可能にする
- コードを共有ライブラリに用意する必要がある

既存技術

- ▶ SWIG(Simplified Wrapper Interface Generator)
 - C/C++で書かれたプログラムをPythonを含む様々な言語と組み合わせることができる
 - 自動的にライブラリ全体をラップする
 - ファイル構成が煩雑になる

既存技術

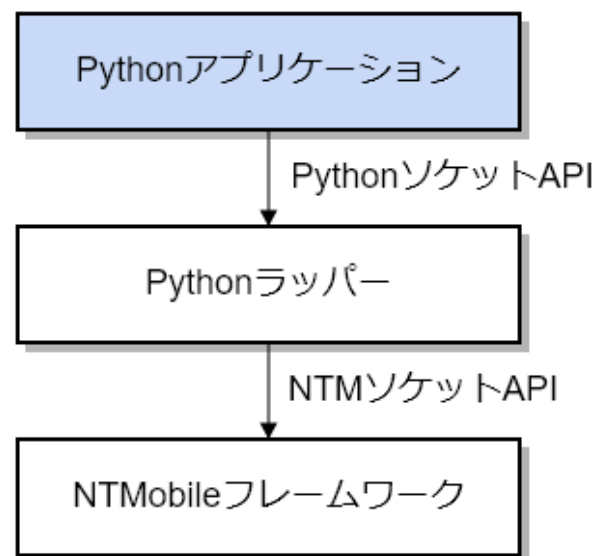
- ▶ 両者に共通する課題
 - アプリケーション側がライブラリのC言語を意識する必要がある
- ▶ 提案内容
 - 通信ライブラリではPython標準のAPIと同じ方法で利用できるようにする

NTMobileフレームワーク

- ▶ NTMobileをアプリケーションで実装するC言語の通信ライブラリ
- ▶ NTMobile
 - 通信接続性
 - ネットワーク環境に関わらず通信を開始することができる
 - 移動透過性
 - ネットワークが切り替わった際も通信を継続することができる

Pythonラッパーの動作

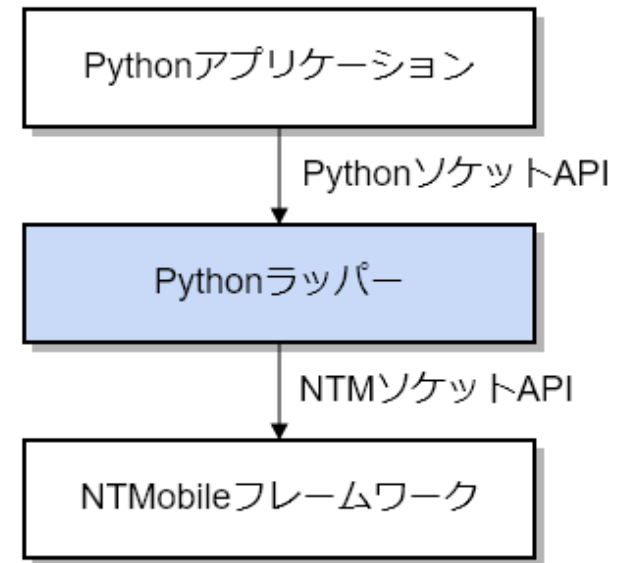
- ▶ Pythonアプリケーション
 - 提案方式
 - PythonソケットAPIでデータを送信
 - 既存方式
 - NTMソケットAPIでデータを送信



Pythonラッパーの動作

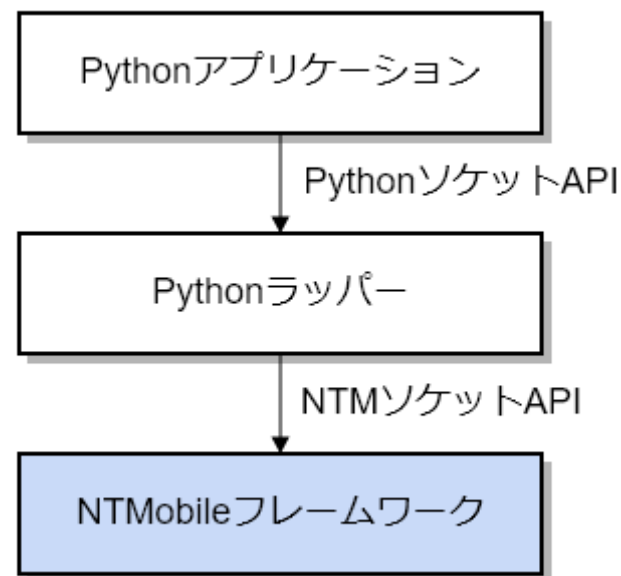
▶ Pythonラッパー

- 提案方式
 - NTMソケットAPIをマッピング
 - 送信するデータの型変換
 - NTMソケットAPIでデータを送信
- 既存方式
 - NTMソケットAPIをマッピング



Pythonラッパーの動作

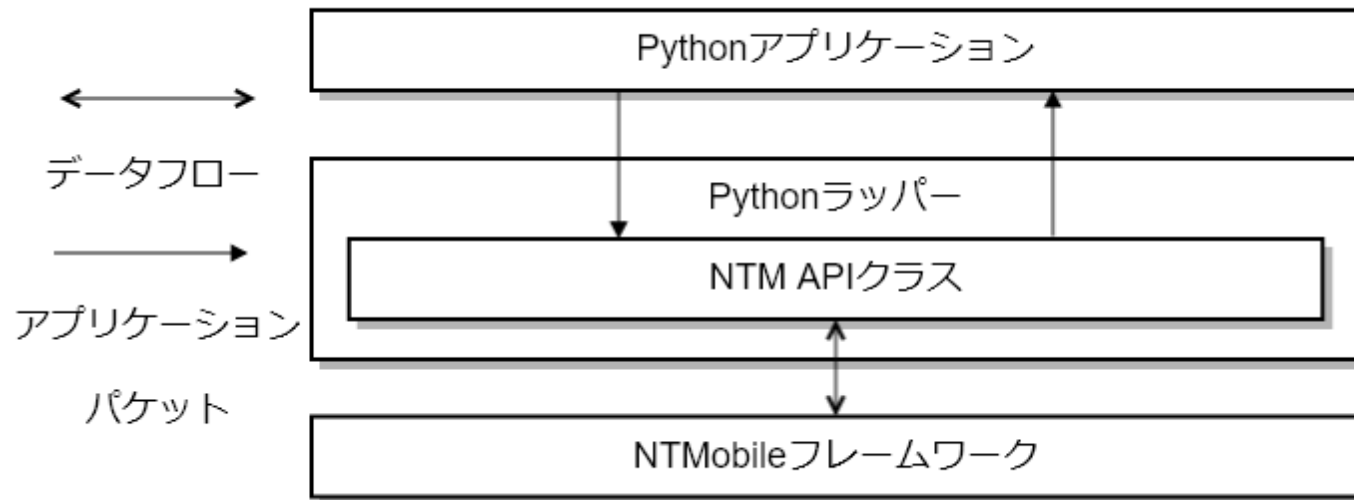
- ▶ NTMobileフレームワーク
 - 仮想IPアドレスを用いてパケットを生成
 - パケットの暗号化やMACの付与
 - BSDソケットAPIで送信



Pythonラッパーの構成

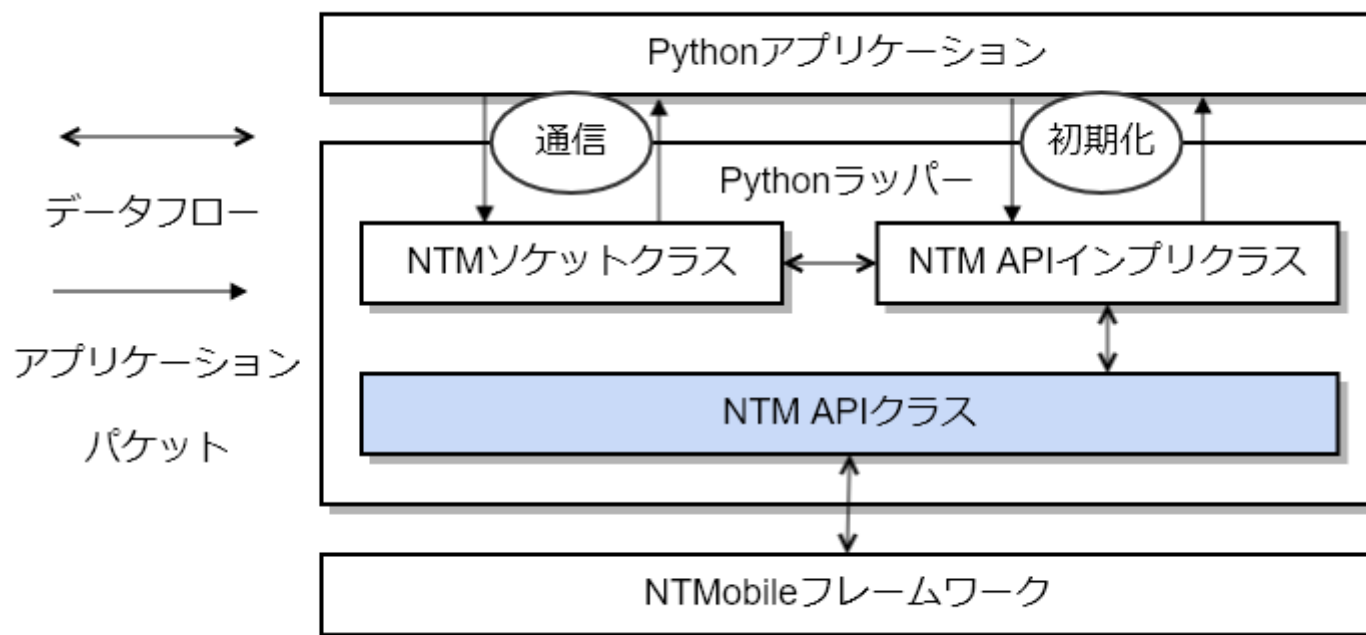
▶ 既存方式

- マッピングされたNTMソケットAPIをラップ



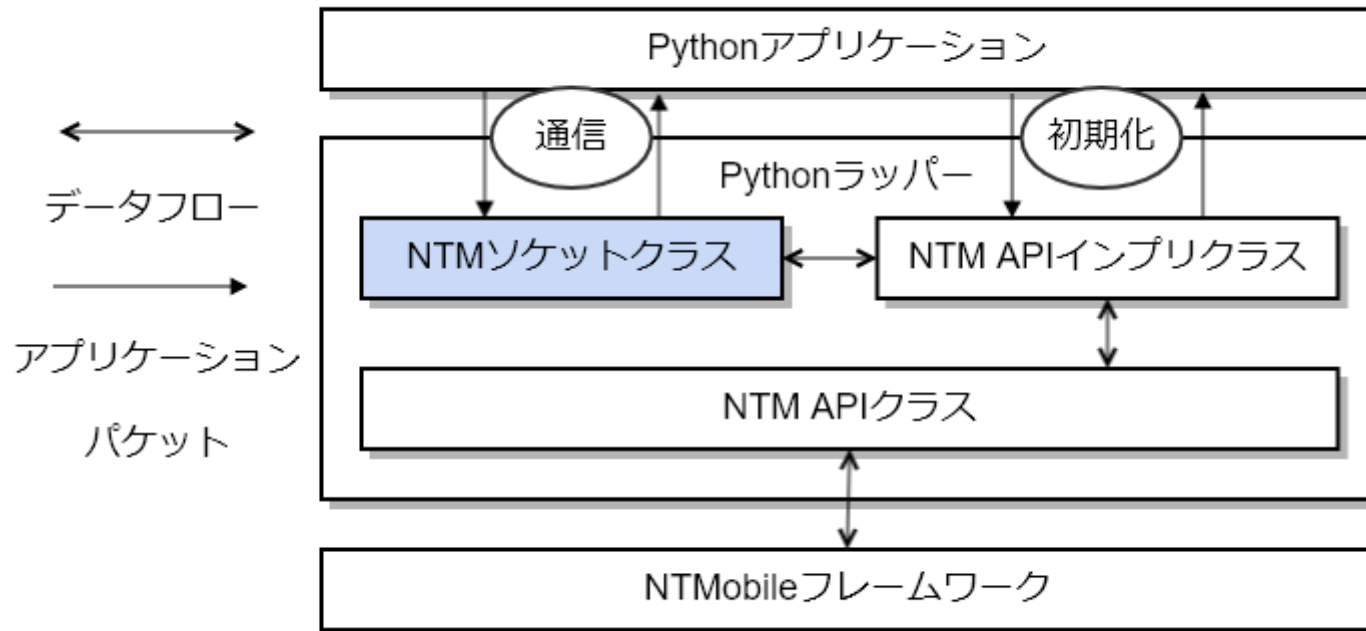
Pythonラッパーの構成

- ▶ マッピングされたNTMソケットAPIをラップ



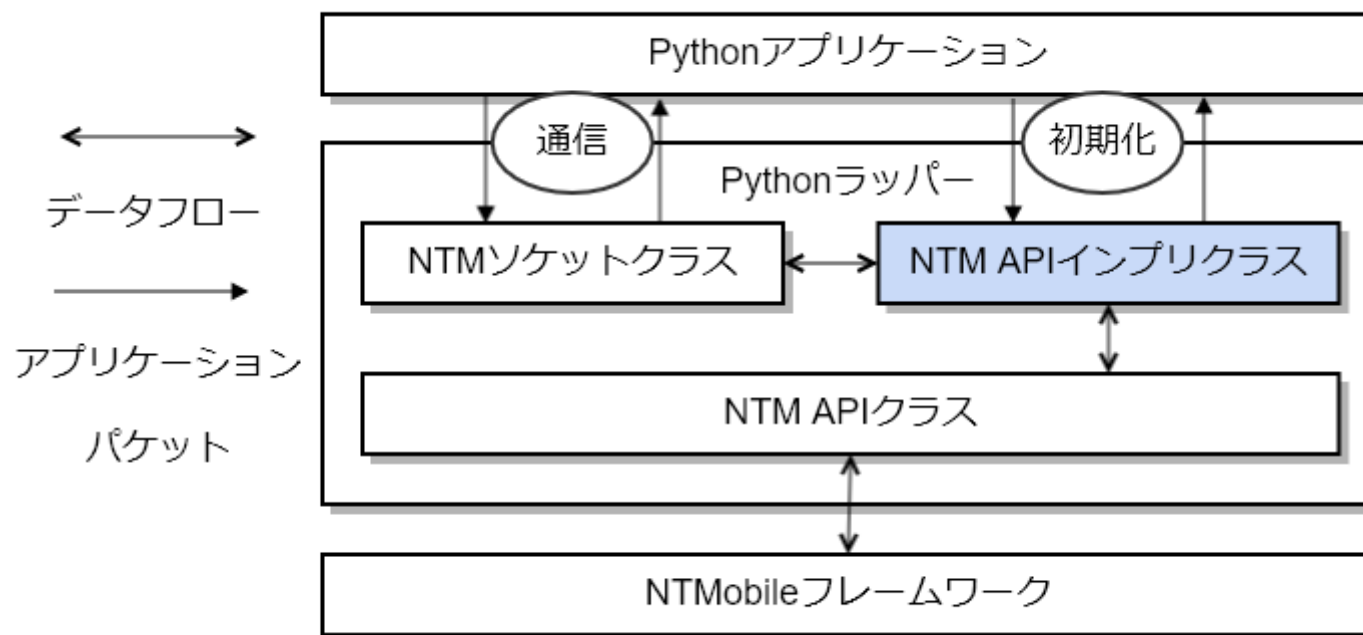
Pythonラッパーの構成

- ▶ Pythonソケットクラスを継承
- ▶ PythonソケットAPI互換のソケットAPI



Pythonラッパーの構成

- ▶ C言語とPythonの違いの除去
- ▶ 起動時の初期化处理



Pythonラッパー

- ▶ 通信機能はPython標準APIと同じ方法で使用
- ▶ Pythonの標準APIにない機能はCライブラリのAPIと同じ方法で使用

まとめ

- ▶ Pythonラッパーの提案
 - 通信機能をPython標準のAPIと同じ方法で利用できるようにすることで、開発負荷を軽減
- ▶ 今後の予定
 - 提案方式の実装、性能評価

NTMobile

- ▶ 仮想的にエンドツーエンド通信を実現
 - AS:NTM端末のアカウント管理
 - DS:仮想アドレスの配布と通信経路を指示
 - RS:必要に応じてパケットを中継する

