

# ユーザ主導型通信を実現するセッション層アーキテクチャ

金子 晋丈

## 1 はじめに

現在インターネット上に様々なデバイスやアプリケーションプログラムが相互に通信できる通信環境が構築されている。しかしながら、現在のインターネットの利用形態は、ユーザがそれぞれのデバイス上で動作するアプリケーションプログラムを直接制御しなければならないため、ネットワーク上に利用可能な資源が遍在しても自由自在に使いこなすことは困難である。

そこで筆者らは、通信デバイスや通信アプリケーションと、アプリケーションプログラム間を繋ぐ通信チャンネルを明確に分離し、ユーザが通信チャンネルを直接管理し制御できるユーザ主導型ネットワークアーキテクチャを構築した。本報告書ではユーザによる通信チャンネル管理・制御機構の開発を中心に、ユーザ主導のネットワークアーキテクチャを実現する基本的機能、およびそのプロトタイプを実装について述べる。

## 2 ユーザ主導型ネットワークアーキテクチャ

### 2.1 ユーザ主導型通信

ユーザ主導型通信は、これまでの端末を主体とした通信ではなくユーザを主体とした通信を可能にするものである。これにより、我々の日常的なコミュニケーションと同様に、誰と話しているかが分かっている安心感や、時間や場所、メディアを変えてコミュニケーションが「継続」する連続性を通信機構として実現することが可能となる。

ユーザ主導型通信における「ユーザ」とは、通信サービスを楽しむユーザであり、提供するプロバイダである。1人のユーザが複数の通信サービスを楽しむ場合には、同一のアイデンティティですべ

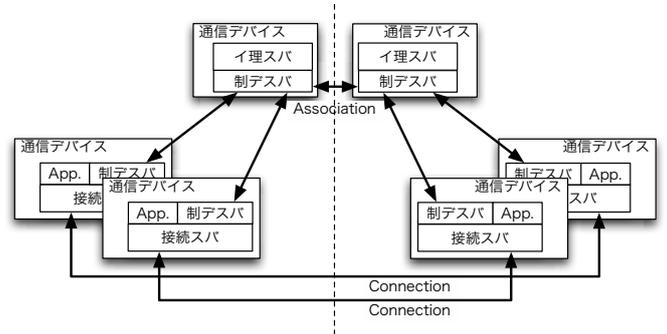


図 1: ユーザ主導型通信のアーキテクチャ  
でのサービスを楽しむ場合もあれば、サービスをそれぞれ異なるアイデンティティで楽しむ場合もある。ユーザ主導型通信は、これらのアイデンティティを暗号的な識別情報を用いて区別する。したがって、通信相手の識別は、暗号的に通信相手のアイデンティティを認証することで可能となる。以下では、アイデンティティの認証が行われた通信者間をアソシエーションと呼ぶ。

### 2.2 ユーザ主導型ネットワークアーキテクチャ

ユーザ主導型通信のアーキテクチャを図1に示す。ユーザ主導型通信は、アソシエーションと実通信の対応付けを管理する機構、管理機構の指示に従い、実際にアプリケーションデータのやりとりを実現する実通信を設定しアプリケーションプログラムと接続する接続機構、管理機構と接続機構間、管理機構間を繋ぐ制御機構によって構築されている。それぞれの機構が保持する機能を以下に示す。

#### (1) 管理機構

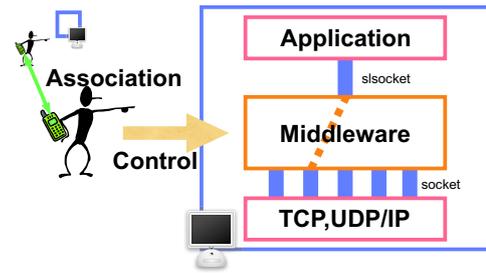
- 実通信環境の把握
- アソシエーションの定義
- アソシエーションと実通信の対応付け

## (2) 接続機構

- アプリケーションプログラムと実通信路の接続
- アプリケーションプログラムが感知できない実通信に関する情報の扱い

## (3) 制御機構

- 通信デバイス間の通信状態情報の一貫性維持



## 3 セッション層アーキテクチャ

### 3.1 セッション層アーキテクチャの概要

セッション層アーキテクチャは、2.で述べたアーキテクチャをセッション層において構築したシステムである。本システムでは、現在、広範に利用されているインターネットをネットワーク基盤に用いることとし、識別情報にIPアドレスとポート番号、トランスポート層のプロトコルを用いている。また、アソシエーションには暗号学的な識別情報を用いることで、安全性を確保する。

セッション層アーキテクチャは、セッションマネージャとソケットマネージャによって構成されている。セッションマネージャは、2.で述べた管理機構と制御機構を保持しており制御端末で動作する。ソケットマネージャは、接続機構と制御機構を保持しており各通信デバイス（実通信端末）で動作する。

### 3.2 セッションマネージャ

セッションマネージャが動作する制御端末に関して、現在、ほとんどのユーザが携帯電話端末を日常的に持ち歩いていることから、これを制御端末もしくは制御端末のユーザインタフェースとして用いる。携帯電話端末はユーザのアイデンティティとして深く根付いており、その意味においても自然であろう。

### 3.3 ソケットマネージャ

一方、実通信端末で動作するソケットマネージャの実現形態として、アプリケーションプログラムへの組み込み、既存のネットワークフレームワークの変更、既存のネットワークフレームワーク上にミド

図 2: 実通信端末の実現形態

ルウェアとして実現の3つのアプローチが考えられる。アプリケーションプログラムへの組み込みは、アプリケーションプログラムの設計に制約を与え、開発コストが高くなる。既存のネットワークフレームワークの変更は、インターネットにおけるフロー識別を根本から問い直すことになり本質的な解決となるが、既存のアプリケーションプログラムをすべて変更しなければならない。既存のネットワークフレームワーク上にミドルウェアを構築するアプローチは、アプリケーションプログラムへの影響を最小限にとどめるとともに、既存のアプリケーションプログラムとの共存も可能である。本稿ではソケットマネージャとしてミドルウェア方式を採用した（図2）。

### 3.4 セッション層アーキテクチャの実装

上記のセッション層アーキテクチャをWindows XP上、およびFreeBSD上に実装を行い、基本的な動作の確認を行った。

## 4 おわりに

本稿では、ユーザが通信相手との通信チャンネルを通信アプリケーションとは独立に管理するユーザ主導型のネットワークアーキテクチャについて述べ、その実現形態であるセッション層アーキテクチャを示した。