

# IPv6/IPv4相互接続技術の概要、実装、運用

角川 宗近

(株)日立製作所 / KAMEプロジェクト  
sumikawa@sumikawa.jp

# IPv4からIPv6への移行技術は必須

---

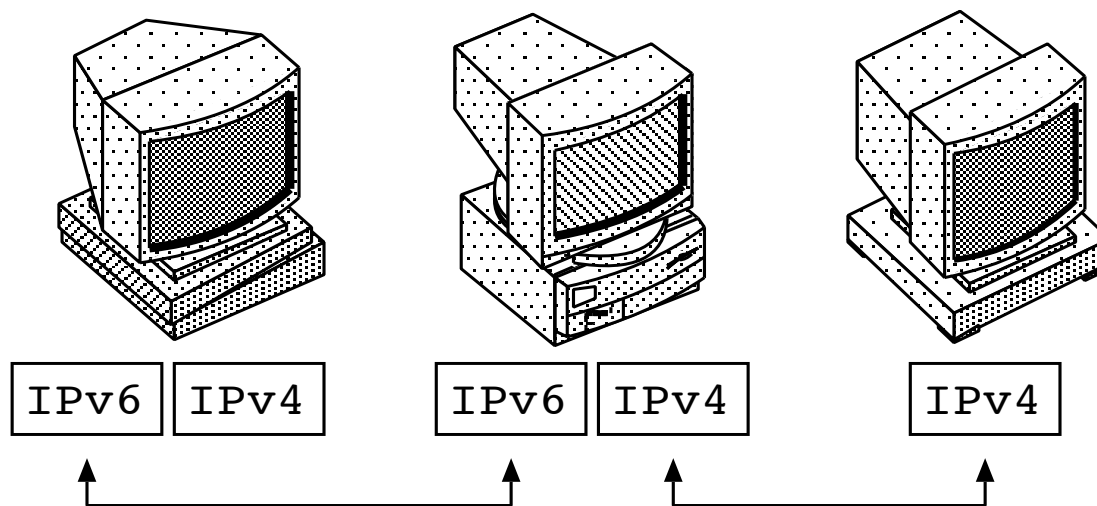
- インターネット中のマシンを一度にIPv6対応にはできない
- IPv4とIPv6を通信させる技術確立する必要がある
- 二つのアプローチ
  - エンド-エンドが直接通信するアプローチ
    - デュアルスタック
    - トンネル
  - トランスレータ経由で通信するアプローチ
    - ヘッダ変換
    - トランスポート中継
    - 代理アプリケーション

# デュアルスタックとトンネル(1/2)

---

## ■ デュアルスタック

- 一つの機器にIPv4とIPv6の両方を実装する
  - IPv4しか対応していない機器とはIPv4で通信
  - IPv6に対応している機器とはIPv6で通信

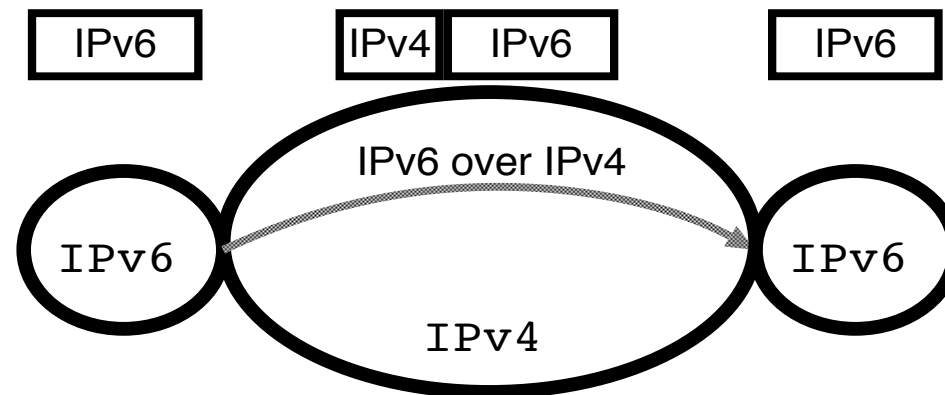


# デュアルスタックとトンネル(2/2)

---

## ■ トンネル

- 離れたIPv6の島を結ぶ技術
- プロバイダによる実験トンネル提供サービス
  - OCN、IIJ
  - <http://www.freenet6.net/>



# トランスレータ

---

## ■ IPv4機器とIPv6機器が通信する必要性

- デュアルスタックでもアドレスが足りなければIPv4アドレスをつけない
- これから登場するIPv6専用端末

## ■ 完璧なトランスレータは存在しない

- 限定条件で使えるトランスレータは幾つかある
  - 状況、移行時期に応じて使い分けるしかない
- トランスレータを使わないにこしたことはない
  - IPv6の本来の目的であるエンド-エンドのフラットな通信を壊す

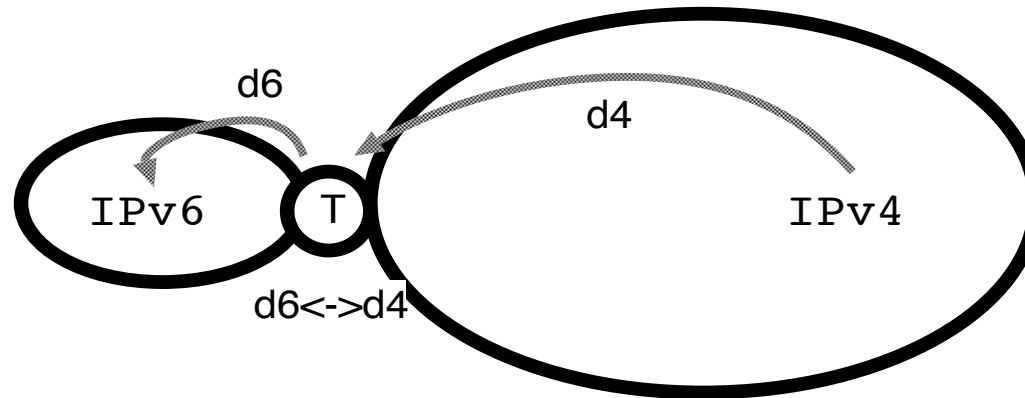
## ■ トランスレータを整理、体系化する必要性

- プロトコル変換の技術によるパターン分け

# トランスレータが不得意な状況

---

- サイト外からのアクセスは難しい
  - サイト内のアドレスが足りないため動的なマッピングが必要
    - DNSキャッシュ
    - マッピング用のアドレス枯渇
    - マッピング開放のタイミング



- IPv6 -> IPv4 はNATと同程度の使い方が適している
- IPv6間は直接通信

# トランスレータの分類

---

## ■ プロトコル変換の技術

- ヘッダ変換
- トランスポート中継
- 代理アプリケーション

# ヘッダ変換

---

- ヘッダを入れ替えて転送
- チェックサムをヘッダの分だけ補正
- ftp問題
- ICMPのセマンティクスが違う
- v4->v6変換時に20バイト増えるので分割しないといけない
- UDPゼロチェックサム問題



# トランスポート中継

---

- コネクションを横取りして、張り直す。
  - ファイアウォール、WWW横取りキャッシュ
- ftp問題はあるが、比較的簡単
- 2本のコネクションを扱う時(rsh等)にやや大変
  - プロトコルごとにケアしないといけない

# 代理アプリケーション

---

- 運用で実現したトランスレータ
- トランザクション・サービスを転送
- プロトコ尔的に代理サービスをサポートしている必要がある
- ftp問題なし

# 既存の実装(1/2)

---

## ■ NAT-PT(KAMEプロジェクト、日立)

- ヘッダ変換による通信
- NATの自然な拡張

## ■ FAITH(KAMEプロジェクト)

- トランスポート中継
- コネクション管理にPCBを用いている
- 実装が容易

## ■ SOCKS(富士通研、NEC)

- SOCKSを用いている組織に導入しやすい

## ■ Toolnet6(日立)

- Bump in the stack方式
- セルフ・トランスレータによるデュアルスタックの実現
  - IPv4アプリケーションが自動的にIPv6対応になる
- モデル的にはトランスレータではない
- ヘッダ変換

## 既存の実装(2/2)

---

- apache、 sendmail、 bind ...
  - 代理アプリケーション

# まとめ

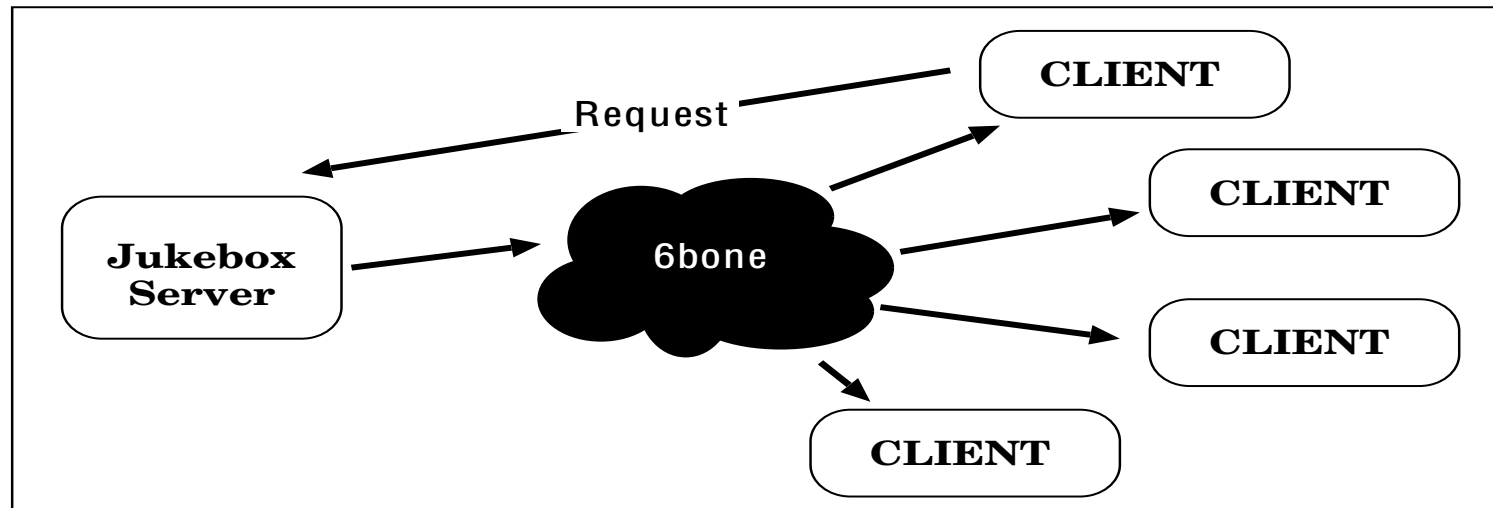
---

- IPv4からIPv6への移行技術は必要
- デュアルスタックとトンネルが最も望ましい
- 一部の状況下ではトランスレータが利用できる
  - ヘッダ変換
  - トランスポート中継
  - 代理アプリケーション
- 状況に応じて使いわける必要がある

# MP3 ジュークボックス

---

- MP3データをUDP over IPv6マルチキャストで配信
  - IPsecで暗号化
- 曲リクエストも可能
  - 曲名一覧取り寄せプロトコル
  - 曲リクエスト・プロトコル
  - リクエスト・キュー確認プロトコル
- CGIも用意、Webベースによるリクエスト



# なぜMP3か？

---

- 圧縮率が高いため、広域に放送できる
  - 128kbpsでCDクオリティ
- サーバ、クライアントが構築しやすい
  - エンコーダ、デコーダが豊富にある
  - サウンド機能さえついていれば気軽に楽しめる
- ユーザが導入しやすいため、広域なマルチキャスト実験ができる

# 実装

---

## ■ サーバ

- icecastを改造

- <http://www.icecast.org/>
- 元はIPv4のHTTPを用いてMP3を配信
- KAMEプロジェクトでIPv6化 + UDP化 + マルチキャスト化
- <ftp://ftp.kame.net/pub/kame/misc/>

## ■ クライアント

- mpg123(コマンドライン)
- wmp3(GUI)
- mcastread (UDP -> stdout)

## ■ KAMEプロジェクトで運用テスト

- KAME FM
  - J-POP中心
- Hard Rock KAME
  - 洋物メタル系



# ■ KAME FM

---

Thank you for accessing KAME-FM.



TCP on unicast: <http://kamefm.kame.net:8000/>  
UDP on multicast: <mcast://kamefm.kame.net:8000>

Get Library List: [shoutreq -L kamefm.kame.net](http://shoutreq -L kamefm.kame.net)  
Request music (e.g. request #116): [shoutreq -M 116 kamefm.kame.net](http://shoutreq -M 116 kamefm.kame.net)  
Confirmation requests: [shoutreq -Q kamefm.kame.net](http://shoutreq -Q kamefm.kame.net)

### Contents


- [information about currently playing music \(Password required\)](#)
- [request music \(multi/single\) \(Password required\)](#)
- [show the current \(queue list\) \(Password required\)](#)
- [KAME-FM photo gallery](#)

### Links


- [Hardwork KAME](#)



# ■ Hard Rock KAME




---



おまえがハードロックを好きな事は知ってるぜ！なぜならここに来る奴は皆そうだからさ！奴らはビートに引かれて勝手に集まって来るんだ。ここでハードロックが流れてるなんて、一言も喋ってないのにな。おまえもビートにつられてやって来たんだろ？隠さなくって自分かってる。さあ、もっとこっちに来いよ。いかれた奴らがおまえを待ってるぜ。

<妙哉求む、おりーず>



[Click me to request](#)

How to listen to me? There is the [instruction guide](#).

You can show the [current queuing list](#), and [playlist](#) which is the list when there is no queuing.

---

hardrock's [kame.net:8000](#), **HARD ROCK KAME**, NOW ON AIR.

If you have some trouble, O.K. send me by [E-Mail](#).

Also look [here](#) is the KAME-N'M official site.

There are other way to enjoy me.

[hardrock.kame.net:8001](#), you can watch the current sound.

[hardrock.kame.net:8003](#), waiting your requests.

[http://hardrock.kame.net:8000/](#) for unicast.