

8T-07 アクティブネットワーク実行環境の試作

伊東 大輔[†] 寺田 松昭[†]

東京農工大学工学部

1 はじめに

従来のネットワークはパケットのデータ部を処理せず「Store & Forward」のみ行うので、「パッシブネットワーク」と言われる。これに対して、ネットワークでパケットのデータ部に処理を施す、つまり「Store & Compute & Forward」を行うネットワークは「アクティブネットワーク」と呼ばれ、数年前から研究が活発化している [1]。

本研究ではアクティブネットワークの実行環境を対象に、様々な実行環境間の相互接続性を改善する方法を考案し、Java¹を用いて実装を試みた。

2 アクティブネットワーク

「Store & Compute & Forward」を行うノードをアクティブノード (以下 AN) は、一般に、ノード OS、実行環境部、アプリケーションの3階層から構成される。

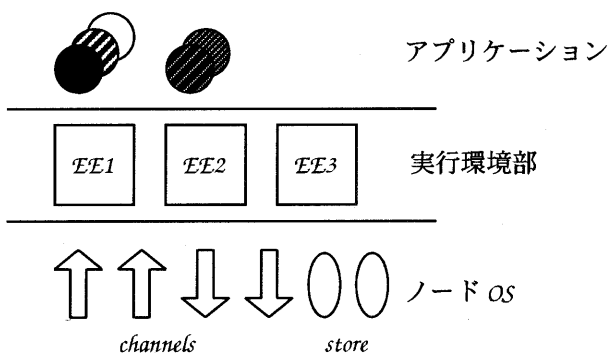


図 1: アクティブノード構成図

ノード OS はノードの資源を管理し、転送、演算処理、メモリなどを含む資源を実行環境に提供する。実行環境はアプリケーションに対する API の提供や、仮

A Study on Active Network Execution Environment
Daisuke ITO[†], Matsuaki TERADA[†]
Faculty of Technology, Tokyo University of Agriculture and Technology
184-8588, Tokyo, Japan
{dai, m-tera}@tela.cs.tuat.ac.jp

¹Java は米国およびその他の国における Sun microsystems, Inc. の登録商標です

想マシンとして動作する。アプリケーションは、実行環境から提供される API を利用し動作するプログラムのことを指す [2][3]。

3 本研究の目標

様々な実行環境が研究されているが、それらは独自のアーキテクチャ、パケットフォーマットを採用し、互換性がない。そこで、実行環境にパケットを渡す共通のアーキテクチャを規定し、すべての AN で同じ処理ができるようにすることで相互接続性を実現する。

4 設計

実行環境にパケットを渡すためには、実行環境を管理する機構が必要になる。これを実行環境プラットフォーム (以下プラットフォーム) と呼ぶことにする。プラットフォームはノード OS から渡されたパケットを最初に処理する必要があるため、実行環境部とノード OS の間に位置する (図 2)。実行環境には、プラットフォームの資源を変更できるもの (以下 PEE) と、できないものの (図 2 中 EEn) 2 種類を設定する。PEE は実行環境の登録、ルート決め方の変更などができる。プラットフォームはアプリケーションを実行環境に渡すか、渡す実行環境がなかった場合は、PEE にその後の処理を任せる。

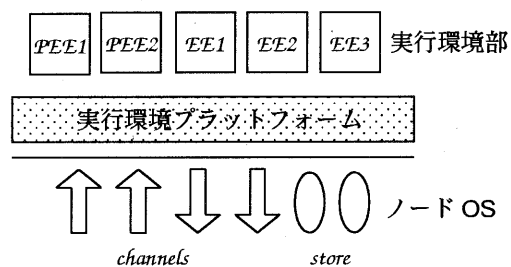


図 2: 実行環境プラットフォームの実装

目的の実行環境が存在しない場合、プラットフォームから処理をまかされた PEE は、パケットヘッダに

示される実行環境の存在する AN までダウンロード要求を出す。ダウンロード要求は、目的の AN までの間に要求に答える AN が存在した場合、そこからダウンロードする。一度ダウンロードされた実行環境は実行環境の更新要求、削除要求が来ない限り AN 上にキャッシュされる。また、ダウンロード要求に答えた AN から要求を出した AN までの中間ノードにも同じくキャッシュされる。(図 3)。

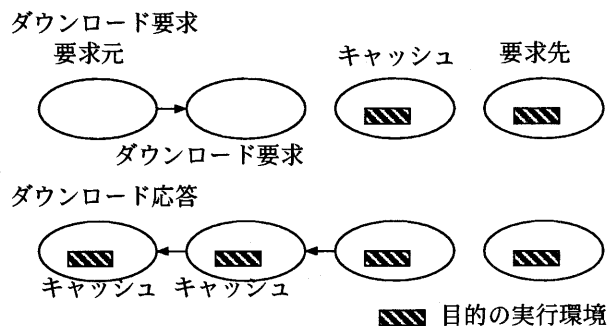


図 3: 実行環境のダウンロード

5 実装

本研究では、上記のアーキテクチャを Java により試作した。Java を使う利点は (i)safety、(ii) シリアライズ化が簡単、ということである。(i) については、厳格な型チェックや例外処理の機能があげられる。(ii) については、実行環境やアプリケーションを作成するユーザに AN の資源を利用する機能を提供しやすく、ユーザ側からは利用しやすいことがあげられる。ただし、Java のみを用いてはネットワーク層に直接アクセスすることができないという問題がある。このため、ルーティング用に、送り先と送り元のみから構成される疑似 IP パケットを UDP パケットにカプセル化した。

疑似 IP パケットには実行環境を識別する実行環境識別パケットがカプセル化される。プラットフォームは実行環境識別ヘッダから目的の実行環境を識別し、実行環境識別パケットのデータ部を渡す。実行環境により処理が終わったパケットは再度疑似 IP パケットにカプセル化され PEE であるルーティング実行環境 (以下 REE) に渡される。REE は次のノードをルーティングテーブルから検索しパケットを転送する。実行環境識別ヘッダは実行環境識別子と実行環境所有ノードの 2 つのフィールドから構成され、プラットフォームは目的の実行環境が存在しない場合 PEE であるダウンロード実行環境 (図 4 中 DEE) に実行環境所有ノードから

ダウンロードするように要求する。

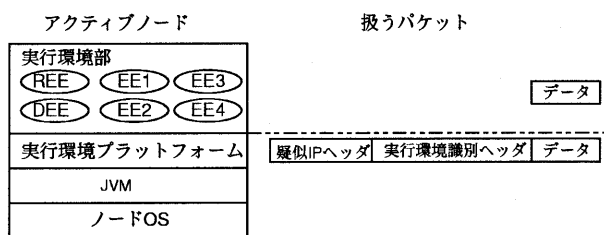


図 4: アクティブノード

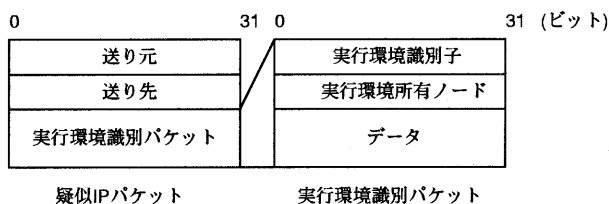


図 5: パケットフォーマット

6 結言

実行環境間の相互接続性を改善する方式を考案し、Java を用いて実装を試みた。

本研究で試作した実行環境は Java のみで作成したため、ネットワーク層以下の資源を利用できず、仮想的な転送処理機構の上に実現した。

セキュリティについては今後の課題である。

参考文献

- [1] D L. Tennenhouse et al., "A Survey of Active Network Research", IEEE Communications Magazine. January 1997.
- [2] K. L. Calvert, ed. "Architectural Framework for Active Networks Version 1.0", University of Kentucky. July 27, 1999.
<http://www.dcs.uky.edu/~calvert/arch-docs.html>
- [3] J M. Smith et al., "Activenetwork A Progress Report", IEEE Communications Magazine. April 1999.