B-028

ユーザレベル OS のためのユーザレベルネットワーク機能
User-level Network Function for User-level Operating Systems

榮樂 英樹 * 新城 靖 † 加藤 和彦 †
Hideki Eiraku Yasushi Shinjo Kazuhiko Kato

1. はじめに

ユーザレベル OS とは，実機上で動作する OS を，他の OS 上でユーザプロセスとして動作可能にしたものである．現在，部分的なハードウェアのエミュレーションを行う軽量 VM (Virtual Machine) と，機械語命令の静的な書き換えによって，ユーザレベル OS を実現している [2] ．現在，ユーザレベル OS として NetBSD, Linux, および FreeBSD が，Linux および NetBSD 上で動作している．その軽量 VM は，イーサネットのデバイスをエミュレートしていなかったので，ネットワーク機能が使えなかった．

本論文では，新たに開発した軽量 VM のネットワーク機能について述べる．その特徴は，ホスト OS の機能を利用して高度な機能を付加している点にある．たとえば，ホスト OS 上の SSL ライブラリを利用することで，ユーザレベル OS 上で動作する HTTP サーバを，SSL 対応にすることができる．また，このような機能を特権利用者の権限を用いずに実現している．

2. 設計と実現

2.1 ユーザレベル OS の外部のネットワークへの接続方法

ユーザレベル OS を，外部のネットワークに接続する代表的な方法は，次の 2 つである．

(1) ホスト OS のネットワークインタフェースを使う方法．ユーザレベル OS が入出力するイーサネットのフレームを，たとえば TUN/TAP 等を経由して，ホスト OS カーネルでイーサネットに対して入出力する．
(2) ソケットインタフェースを使う方法．ホスト OS のソケットインタフェースを介してユーザレベルで入出力する．代表的なものとして Slirp[3] がある．

後者は，前者と比べて，安全である．前者の方法では，ユーザレベル OS は，不正なパケットを送ることが可能である．また，後者は特権が不要である．

本研究では，後者の方法を採用した．その理由は，高度な機能をユーザレベルで安全に付加することができるからである．

2.2 軽量 VM によるユーザレベルネットワーク機能の実現

ネットワーク機能を実現する方法として，ユーザレベル OS 内に組み込む方法と軽量 VM で行う方法が考えられる．本研究では，軽量 VM で実現することにした．その理由は，複数のユーザレベル OS を支援することが容易になるからである．

2.1 節で述べたソケットインタフェースを使う方法では，軽量 VM は，TCP/IP のヘッダを取り除いたり，付加する必要がある (図 1)．それには，まず既存の TCP/IP スタックを利用する方法が考えられる．Slirp は，4.4BSD の TCP/IP コードをベースとして，この方法で実現されている．この方法の利点は，コードを再利用できる点である．しかし，既存の TCP/IP スタックは VM 用に設計されておらず，機能拡張が難しい．そこで，今回は軽量 VM のための簡易スタックを開発することにした．

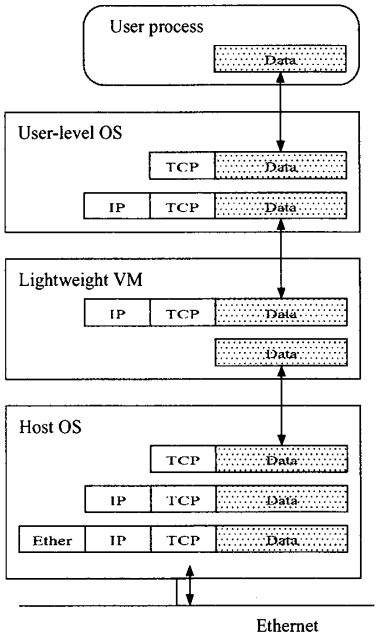

図 1: 軽量 VM によるユーザレベルネットワーク機能の実現

簡易スタックは，図 1 に示すように，ユーザレベル OS が出力する IP パケットから，IP 層および TCP 層のヘッダを取り除き，データを取り出す．次に，そのデータを通常のソケットインタフェースを用いてホスト OS に渡す．逆に，軽量 VM はホスト OS のソケットを通じてデータを受信すると，そのデータに対して，TCP 層および IP 層のヘッダを付加し，ユーザレベル OS へ渡す．

2.3 ユーザレベル OS のための特化

通常の TCP/IP スタックには，以下のような機能がある．

- 基本データ転送
- 信頼性
- フロー制御
- 多重化
- コネクション
- 優先度とセキュリティ

本軽量 VM ではユーザレベル OS のために特化し，これらの機能のうち，必要なものだけを実現する．必要な

*筑波大学第三学群情報学類
†筑波大学大学院システム情報工学研究科コンピュータサイエンス専攻／ JST CREST

機能とは，基本データ転送，フロー制御，多重化，コネクションである．ユーザレベル OS と軽量 VM の間は，信頼性のある通信路で結ばれているので，メッセージの破損，消失，重複が起こらない．また，メッセージの書き換えのような，セキュリティについても考慮する必要がない．

2.4 外部からの接続の受け付け

ユーザレベル OS でサーバを動作させ，外部から接続する場合，軽量 VM にあらかじめポート番号などを与えておく必要がある．なぜなら，サーバを動作させても，クライアントから接続要求がない限りサーバからパケットが出力されることはなく，軽量 VM は，サーバが動作していることがわからないためである．また，特権ポートを用いるサーバを，ユーザレベル OS 上で安全に実行するために，ホスト OS のポート番号マッピング機能を用いている．

2.5 デバイスドライバ

ユーザレベル OS が出力する IP パケットを，軽量 VM で取得するには，次の 3 つの方法が考えられる．

(1) イーサネットデバイスのエミュレーションを行う方法
(2) シリアルポートを使う方法
(3) 軽量 VM と通信するためのデバイスドライバをユーザレベル OS に追加する方法

ここでは，(3) の方法を使うことにした．(1) の方法は，イーサネットデバイスのエミュレーションがあまり容易ではなく，またオーバヘッドも大きい．(2) の方法は，シリアルに変換することによるオーバヘッドが避けられない．なお，デバイスドライバと軽量 VM の間の通信には，Unix ドメインのソケットを用いている．

3. ユーザレベル OS のための機能拡張

ユーザレベルネットワーク機能では，パケットフィルタリングのように，従来特権が必要だった機能を，特権なしで利用できるようにする．このような機能をあらかじめ軽量 VM で提供することも検討したが，独自のカスタマイズを容易にするため，モジュールを使って実現する．以下のような機能をモジュールにより提供する．

- パケットフィルタリング: 普通のスタックと同じである．
- コネクショントラッキング: 簡易スタックを活用することでコネクショントラッキングを実現する．
- プロトコル変換: 生の通信を，SSL に対応させたり，IPv4・IPv6 の変換を行う．これにより，たとえば HTTP や POP3 のサーバまたはクライアントを，それぞれ HTTPS や POP3S のサーバまたはクライアントとして利用することができる．

4. 実験

現在，軽量 VM の簡易スタックの基本部分は動作しており，機能拡張部分を開発している．基本部分の大きさは 1,700 行であり，Slirp の対応する部分の 6,000 行の 1/3 の大きさである．

実現した簡易スタックの性能を測定した．環境は，CPU が Pentium 4 3.00GHz，ネットワークは 1000BASE-T で，NIC は Intel PRO/1000 である．ホスト OS は Linux 2.4.24，ユーザレベル OS は NetBSD 1.6.1 で，User-mode Linux[1] はバージョン 2.4.26-1um である．実験では，TCP/IP のサーバを実機上で動かし，クライアントをユーザレベル OS 上で動かし，その転送速度をサーバ側で測定した．結果を表 1 に示す．このように，実機の 1/100 で，User-mode Linux の Slirp より遅いため，チューニングが必要であることがわかる．

表 1: 性能 (単位: MB/s)

OS/環境	性能
Linux/VM なし	117.3
User-mode Linux/TUN/TAP	33.9
User-mode Linux/Slirp	9.1
本ネットワーク機能 (NetBSD/軽量 VM)	1.2

5. 関連研究

User-mode Linux[1] は，Linux を，Linux 上でユーザプロセスとして実行できるように移植したユーザレベル OS である．User-mode Linux のネットワーク機能では，本論文で述べた方法と同じように，デバイスドライバを組み込む方法を用いている．外部のネットワークへ接続するには，次の 3 つの方法がある．

(1) TUN/TAP のように，イーサネットのフレームをそのままやり取りする方法
(2) PPP や SLIP を用いる方法
(3) Slirp を用いる方法 (2.1 節)

(1) と (2) はホスト OS の特権が必要となる．

6. まとめ

この論文では，ユーザレベル OS のためのユーザレベルネットワーク機能の実現について述べた．これは，VM において，IP パケットを，ホスト OS のソケットインタフェースを介してユーザレベルで入出力する．モジュールによる機能拡張が可能である．

今後の課題は，性能を改善すること，および，機能拡張部分を実現することである．

謝辞

この軽量 VM の開発の一部は，萩谷プロジェクトマネージャのもと，情報処理振興事業協会（IPA）平成 15 年度未踏ソフトウェア創造事業の援助を受けて行った．

参考文献

[1] Dike, J.: A user-mode port of the Linux kernel, *the 4th Annual Linux Showcase & Conference* (2000).
[2] Eiraku, H. and Shinjo, Y.: Running BSD Kernels as User Processes by Partial Emulation and Rewriting of Machine Instructions, *USENIX BSDCon 2003 Conference (BSDCon'03)* (September 2003).
[3] Gasparovski, D.: *Slirp Manual* (1995).