5W-7

# ネットワークプロセッサを用いたVPNの実装と評価

黒羽 秀一[††]　金田 健太郎[†]　隠岐 徹[†]　初谷 良輔[††]　齋藤 孝道[†]

† 明治大学

## 1 はじめに

IP パケットの高速処理が求められるルータやレイヤ 3 スイッチなどのネットワーク機器の実現において，ASIC(Application Specific Integrated Circuit) と比較して柔軟で高機能なデバイスであるネットワークプロセッサ (以下，NP と呼ぶ) が注目されている．高速化の手段として，NP では，専用モジュールを用いて，パケット処理や暗号処理を行っている．

暗号モジュールを持つ NP を用いて，IPSec-VPN における暗号処理をオフロードし評価したものとして [2][3] があるが，SSL(Secure Socket Layer)[4] や TLS(Transport Layer Security)[5] を用いた VPN の実装例は少ない．そこで，本論文では，Intel 社の NP である IXP425 [6] を搭載する評価ボード (以下，評価ボードと呼ぶ) 上に SSL-VPN を実装し，パフォーマンス計測と考察を行う．

## 2 ネットワークプロセッサ

### 2.1 概要

NP は，一般に，汎用プロセッサ，パケット処理専用モジュール (以下，NPE[1] と呼ぶ)，メモリ，外部接続のためのインタフェースから構成されている．NP 用の基本ソフトウェアは，NPE 上で動作するパケット処理用のプログラムと汎用プロセッサ上で動作する制御プログラムの 2 種類のプログラムから構成されている．

### 2.2 IXP425 アーキテクチャ

本論文で用いる IXP425 は，主に，XScale アーキテクチャを基にした汎用プロセッサ (以下，XScale コアと呼ぶ) と 2 つの NPE から構成されている．

XScale コアは，NPE と周辺装置などの一般的な処理を行い，NPE は主に IP パケットの処理を行う．

片方の NPE は暗号モジュールを内蔵している．この暗号モジュールは，共通鍵暗号化方式の DES, 3DES と AES[2] に対応しており，その利用モードとして，CBC(Cipher Block Chaining) と ECB(Electronic Code Book) が利用可能である．また，ハッシュアルゴリズムとして SHA-1 と MD5 に対応している．

## 3 開発環境

### 3.1 IXP425 搭載評価ボード

評価ボードは，主に，IXP425，プログラムメモリである Flash ROM，メインメモリである SDRAM と 2 つの NIC から構成されている．また，今回，OS は $\mu$Clinux を利用する．

### 3.2 暗号モジュールの利用法

IXP425 の NPE が提供する暗号処理機能を利用するための一手法として，IXP4*xx* シリーズ向けのデバイスドライバ開発用の API である AccessLibrary がある．アプリケーションから暗号処理機能を利用するためには，まず，専用のデバイスドライバを実装し，そのデバイスドライバにアクセスする仕組みが必要となる．しかしながら，この手法は開発コストが高くつき，さらに，低レベルの暗号処理を独自に実装することは非常にリスクが高いため，本論文では，OpenSSL と OCF(OpenBSD Cryptographic Framework)[7] を組み合わせた開発基盤を構築し，暗号モジュールを抽象化する OCF 経由でユーザ空間から AccessLibrary を介して暗号モジュールにアクセスする (図 1)．

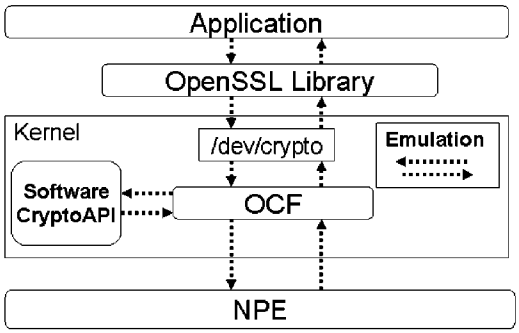


図 1: 利用の概要

## 4 SSL-VPN の実装

実装した SSL-VPN の概要について，ホスト 1 が異なるネットワーク上のホスト 2 にパケットを送信する場合を例にして，図 2 を用いて説明する：

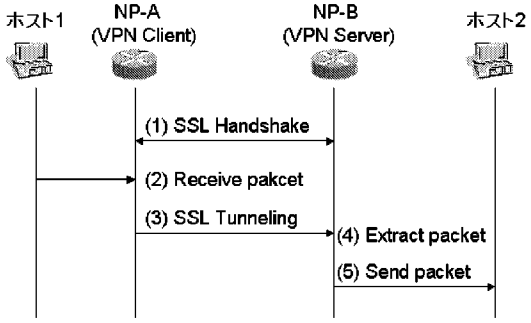


図 2: VPN の概要

まず，VPN クライアントとして動作する NP-A と VPN サーバとして動作する NP-B 間で SSL ハンドシェイクによって VPN 接続 (SSL セッション) を確立する (図 2 (1))．次に，NP-A がホスト 1 からのイーサネットフレームを受信し (図 2 (2))，宛先の確認をして IP パケットを SSL レコードでカプセル化し，NP-B へ送信する (図 2 (3))．この時の暗号化は NPE で行う．NP-B がそれを NPE 上で復号して元のパケットを抽出し (図 2 (4))，ホスト 2 へ送信する (図 2 (5))．

## 5 評価

評価ボードの性能評価として，パケットの転送，TCP 通信，暗号処理，ハッシュ処理のスループットを計測し，評価ボード間の SSL 通信のスループットを計測した．その上で，実装した SSL-VPN のスループットを計測した．

### 5.1 パケットの転送速度

図 2 の環境で，Iperf [8] を利用してホスト 1 とホスト 2 間の TCP 通信速度を計測した．ただし，TCP コネク

†† Shuichi KUROBA, Ryosuke HATSUGAI
† Kentaro KANEDA, Toru OKI, Takamichi SAITO
{kuroba, kaneda, oki, hatsugai, saito}@cs.meiji.ac.jp
Meiji University(†), Graduate School of Meiji University(††)
1-1-1 Higashimita, Tama-ku, Kawasaki-shi, Kanagawa, 214-8571, Japan(†)(††)

[1] NPE は Network Processor Engine の略称である．

[2] 鍵長は 128bit のみに対応している．

ションが確立されているのはホスト1とホスト2間であり，NP-AとNP-Bは，パケットの転送処理のみを行う．また，ホスト1とホスト2はPentium4(3.00GHz)と1024MbyteのDDR-SDRAMメモリを搭載するマシンであり，NP-AとNP-BはIP forwardを有効とした．計測結果は，平均して94.1Mbpsであった．

### 5.2 評価ボード間のTCP通信速度

図2のNP-AとNP-Bの2者間でのTCP通信のスループットを計測した(表1)．送信データサイズが1024byteから8192byteの間は，サイズが大きくなるにつれTCP通信のスループットが向上したが，8192byte以降は約89Mbpsで一定で，これが最大値と推定できる．

### 5.3 暗号処理の速度

DES，3DES，AES-128をCBCモードで利用して，暗号モジュールとソフトウェアのスループットを計測した．3種とも，ソフトウェア処理の場合，データサイズによるスループットへの影響は少なかったが，暗号モジュールを利用した場合，データサイズが大きくなにつれてスループットが向上した．AES-128の結果を表2に示す．表中のXScaleはソフトウェア，Engineは暗号モジュールによる処理をそれぞれ示す．

### 5.4 ハッシュ処理の速度

SHA-1とMD5のスループットを計測した．ただし，OCFがSHA-1とMD5に対応していないため，ソフトウェアについてのみ計測した．両方ともスループットは他の通信や処理を上回り，データサイズが大きくなるにつれて向上した．SHA-1の結果を表3に示す．

### 5.5 評価ボード間のSSL通信速度

図2のNP-AとNP-Bの2者間でのSSL通信速度を計測し，暗号処理を暗号モジュールで行った場合とソフトウェアで行った場合のスループットを比較した．計測は，DES，3DESとAES-128の場合について行い，ハッシュ処理にはSHA-1のみを使用した．スループットは，暗号モジュール，ソフトウェア共にデータサイズが大きくなるにつれて向上したが，データサイズが小さいときはソフトウェアの方が高く，1500byteからは暗号モジュールの方が高くなった．AES-128の場合の結果を表4に示す．

### 5.6 SSL-VPNのスループット

図2において，NP-AとNP-B間でSSL-VPNを張り，ホスト1とホスト2間でIperfによるTCP通信を行った際のスループットを計測した．NP-AとNP-B間の暗号スイートにはAES-128-CBC, SHA-1を利用した．結果は表5に示すとおり，パケット送受信のバッファサイズに関係なく約6Mbpsとなった．

## 6 考察

ここでは，SSLを構成する部位であるTCP通信，暗号化/復号，ハッシュ処理の各スループット(5.2, 5.3, 5.4副節)から想定されるSSL-VPNの理論上のスループットと実測したSSL-VPNのスループット(5.6副節)を比較し，IXP425，および，実装したSSL-VPNのパフォーマンスとそのボトルネックについて考察する．

実装したSSL-VPNは，受信したパケット毎に暗号化/復号，ハッシュ処理，そしてTCP通信を行うが，これらの各処理に入力されるデータサイズは，EthernetのMTU(Maximum Transmission Unit)の最大値である1500byte以下である．この範囲で最もスループットが低いのは，計測結果から，暗号化の65.8Mbpsであり，これがIXP425上でのSSL通信全体の理論上のスループットの最大値と推定できる．

しかしながら，SSL通信と実装したSSL-VPNのスループットは，それぞれ約10Mbps，約6Mbpsで，理論値と比べて大きく下回ることがわかる．その原因の一つとして，XScaleコアへの負荷の集中が考えられ，

表1: TCP通信速度

| Data size (byte) | 1024 | 1500 | 4069 | 8192 | 10240 |
|---|---|---|---|---|---|
| Throughput (Mbps) | 83.5 | 84.2 | 88.0 | 89.9 | 89.1 |

表2: AES-128_CBC

| Data size (byte) | 256 | 512 | 1024 | 1500 | 2048 |
|---|---|---|---|---|---|
| XScale (Mbps) | 31.1 | 31.5 | 31.6 | 31.6 | 31.6 |
| Engine (Mbps) | 37.2 | 54.3 | 69.3 | 65.8 | 82.4 |

表3: SHA1

| Data size (byte) | 256 | 512 | 1024 | 1500 | 2048 |
|---|---|---|---|---|---|
| Throughput (Mbps) | 122.7 | 131.2 | 135.8 | 137.1 | 139.8 |

表4: SSL通信：AES-128-CBC-SHA1

| Data size (byte) | 256 | 512 | 1024 | 1500 | 2048 |
|---|---|---|---|---|---|
| XScale (Mbps) | 4.5 | 6.9 | 9.5 | 10.5 | 11.9 |
| Engine (Mbps) | 2.8 | 5.2 | 8.9 | 10.9 | 13.7 |

表5: SSL-VPN：AES-128-CBC-SHA1

| Buffer length (byte) | 512 | 1K | 10K | 100K | 1M |
|---|---|---|---|---|---|
| Throughput (Mbps) | 6.05 | 6.02 | 6.06 | 6.07 | 6.06 |

OCFや暗号エンジンに対する様々な制御に実行遅延が発生し，これがパフォーマンスを低下させていると推定できる．これを検証するために，実際にXScaleコアに対して高い負荷をかけた状態で暗号エンジンのスループットを計測したところ，そのスループットが大幅に低下することを確認した．

## 7 まとめと今後の課題

本論文では，Intel社のNPであるIXP425を搭載する評価ボード上に実装したSSL-VPNのパフォーマンス計測を行い，パフォーマンス上のボトルネックについて考察した．

今後の課題として，ルーティングの実装や管理用インタフェースの実装，そして，実装したSSL-VPNのパフォーマンスの向上が挙げられる．

### 参考文献


[1] R. Atkinson, "Security architecture for the Internet protocol", RFC1825, IETF Networking Group, August 1995.

[2] Yi-Neng Lin, Chiuan-Hung Lin, Ying-Dar Lin, and Yuan-Chen Lai, "VPN Gateways over Network Processors: Implementation and Evaluation", RTAS2005.

[3] 下國治，河合純，陣崎明，山澤昌夫，中村修，村井純:Security Network Processor による低消費電力 IPSec ESP の実装と評価，インターネットコンファレンス 2003 論文集,pp.51-58，2003.

[4] Alan O. Freier, Philip Kocher, and Paul C. Kaltorn, “The SSL Protocol Version 3.0 draft”, March 1996

[5] Tim Dierks and Christopher Allen, “RFC2246: The TLS Protocol Version 1.0”, Jan 1999.

[6] http://www.intel.com/design/network/products/npfamily/index.htm

[7] A. D. Keromytis, J. L. Wright, and T. de Raadt, "The Design of the OpenBSD Cryptographic Framework", In Proceedings of the USENIX Annual Technical Conference, June 2003.

[8] http://dast.nlanr.net/Projects/Iperf/