

情報の発信・共有・検索・受信に積極的に関わる ルータクラウドに関する研究

研究代表者 鯉淵 道紘 国立情報学研究所・アーキテクチャ科学研究系・准教授
研究分担者 西 宏章 慶應義塾大学大学院・理工学研究科・准教授
連携研究者 川島 英之 筑波大学大学院・システム情報工学研究科・講師

1 研究の概要

研究目的 本研究では、情報爆発時代における人類の全情報の収集、整理を効率的に行うために、インターネット・ルータがパケット転送するのみならず、情報の発信・共有・検索・受信に積極的に関わるルータクラウド・インフラストラクチャを提案、探求する。ネットワークルータは、エンドデバイスでは知り得ない広範囲な情報（例えば詳細な行動履歴）を獲得できる。このルータの立地の良さを利用して、本研究では、まず（１）パケットストリームを内部のメモリに取り込み、解析するサービス指向ルータ・アーキテクチャを提案する。次に、（２）パケットストリーム解析を支援するストリーム処理エンジンを提案する。そして、（３）サービス指向ルータにより実現される代表的なアプリケーション例として詳細な行動履歴解析や膨大なサイエンスデータ処理などを検討する。本研究と既存のインターネットとの比較を図 1 に示す。

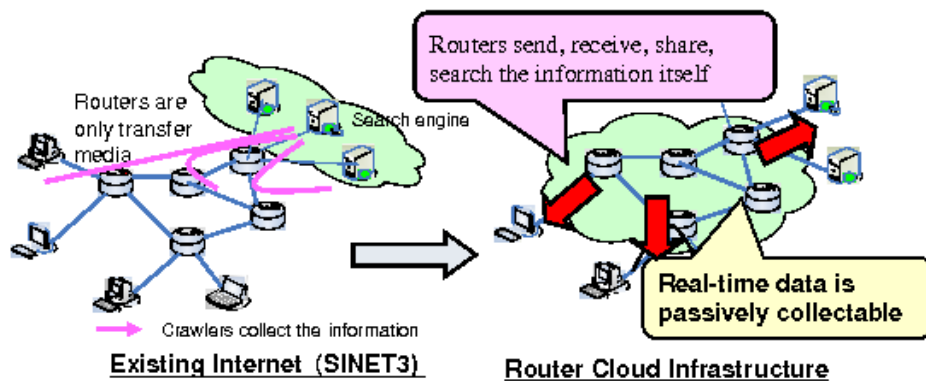


図 1: 既存のインターネットとルータクラウド・インフラストラクチャ

本年度の研究成果の概要 本年度は、目的を達成するために下記の 2 つの課題に大きく分けて取り組んだ。

- サービス指向ルータの要素技術の研究
- サービス指向ルータのシミュレータの研究

サービス指向ルータの要素技術の研究として、本年度、(1) 正規表現プロセッサ、(2) 高速データベースインサージョン、(3) パケット単位の文字列検索などの課題に取り組み、6Gbps でのストリームの取り組みの実現性を示すなどの成果をあげた [1, 2, 5, 7, 6]。

サービス指向ルータのシミュレータの研究として、まず、インターネット・ルータが情報の発信・共有・検索・受信の処理に積極的に関わることで、リアルタイムでの端末で得られない豊富な情報収集によるアプリケーション高度化を支援するネットワークモデルの提案を進めた。さらに、そのルータ、ゲートウェイなどのインターネット・インフラストラクチャのシミュレータを開発するなどの成果をあげた [3, 4]。

以下、順に説明する。

2 サービス指向ルータの要素技術の研究

2.1 サービス指向ルータ

近年、インターネット上に存在するあらゆる情報は、マッシュアップなど複数発信源からの情報を組み合わせる手法により、多角的な価値とコンテンツとしての意義を有するようになった。例えば Google や Amazon が次々と生み出す新技術が Web アプリケーション・サービスの高度化に寄与し、新たなビジネスを掘り起こし広げている。現在、Web アプリケーション・サービスのさらなる高度化のための 1 つの方法として、インターネット・インフラストラクチャであるルータやゲートウェイが取得可能な情報を積極的に活用する、あるいはルータが担うスイッチングをサービスに追加する研究が注目されている。

我々は、トラフィックからアプリケーションが必要とする情報を正規表現により抽出するサービス支援型ルータを提案している [8]。このルータは、図 2 に示した通り、既存のルータに高速な情報抽出を行うための正規表現プロセッサとオンメモリデータベースへの高速なデータインサクションを行うハードウェアを加えた構成を持つ。また、対象となる表現はユーザが SQL を基にした処理系を用いて記述することでルータに登録する [3]。

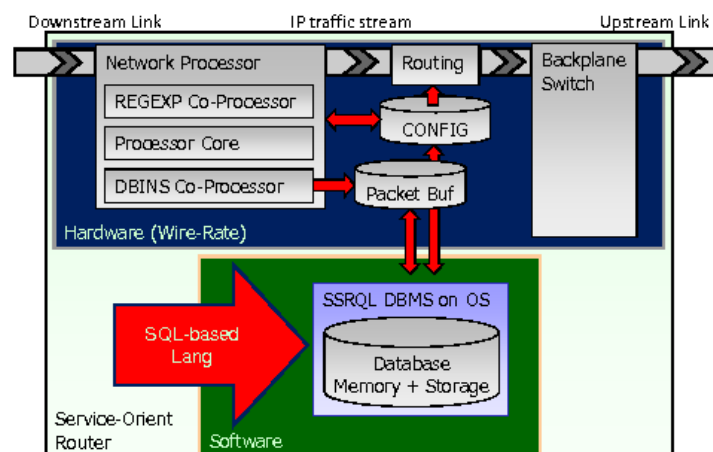


図 2: サービス指向ルータの概要

2.2 正規表現プロセッサ

正規表現処理は、ネットワーク侵入検知システム (NIDS) などをはじめ様々なネットワークアプリケーションで広く使われている。代表的な NIDS である Snort や Bro もパターンファイルに正規表現を用いている。しかしこれらはソフトウェアベースのため処理速度が遅くネットワークのサービス品質に影響を及ぼしている。高速な処理を妨げる要因の一つは処理時間の大半を占めるパターンマッチングである。NIDS のパターンマッチングに用いられるパターン数は膨大であり、かつ複雑な正規表現も含まれている。このような膨大かつ複雑な正規表現を高速で処理できるエンジンのニーズは高い。これまで決定性有限オートマトン (DFA) や非決定性有限オートマトン (NFA) ベースのリコンフィギュラブルなデバイスを使うことにより

正規表現処理を加速する多くの研究が広く行われてきた。しかし DFA や NFA ベースの再構築可能なシステムは、正規表現処理のパターンを頻繁に更新するごとに回路を書き変えるため、サービス指向ルータが目指しているユーザからのリクエストに応じて短期間でパターンを更新する場合には、必ずしも最適とは限らない。

そこで、我々は、その他のアプローチとして迅速なパターン更新を可能とするプロセッサタイプの正規表現プロセッサを提案、検討した。

正規表現プロセッサは、ネットワークプロセッサにおける複数のプロセッシングユニットアレイ (PU) に付随するコプロセッサとして搭載される。パケットは途中で PU と、正規表現プロセッサの両方に送られる。

正規表現プロセッサコアは一般的な RISC プロセッサが備えるうち基本的な命令セットに加え、正規表現処理と抽出処理に特化した命令を拡張したプロセッサである。正規表現プロセッサは正規表現におけるマッチング処理を専用に行う複数のマルチファンクションコンパレータを集中管理し、バレルシフタ内のある範囲の文字列を割り当て比較する。

パケットはストリームと呼ばれるまとまった単位で順次正規表現プロセッサのバレルシフタに情報が送られる。バレルシフタは、16 文字までの文字列を任意の範囲で取り出すことが可能である。また並列取り出しも可能である。

取り出した文字列はマルチファンクションコンパレータ内に複数あるビットマップコンパレータアレイとバッファに送られる。抽出された文字列は、ポストプロセッサに送られる。そして、ポストプロセッサは抽出すべき文字列の最終選択を行う。その結果はハードウェアベースのデータベースインサージョンエンジンがデータベースへ書き込む。

1つのビットマップコンパレータアレイは、128 個のフリップフロップのアレイで構成され、各フリップフロップを1つの文字に対応させ、その値 (0 or 1) によりその文字が抽出対象文字かどうかを判定する。これにより、パイプ (||) を用いた 7-bit ASCII 文字の比較を1ステップで行うなどの高速化が可能である。

我々は、Verilog HDL を用いてビットマップコンパレータを設計した。そして、これらを 45nm の CMOS プロセスを用いて Design Compiler を用いて論理合成した。その結果を表 1 に示す。MCPS は Mega Character per Second の略である。これらの結果より、使い方を限定することにより 1Gbps 程度のスループットがあれば許容されるエッジルータなどで用途を限定することで本正規表現プロセッサアーキテクチャを導入することは可能であるといえる。

表 1: 論理規模及び動作遅延

回路面積 (μm^2)	動作遅延 (ns)	スループット
4060.8	1.34	7.46MCPS, 5.97Gbps

2.3 データベースインサージョンエンジン

2.3.1 概要

一般にデータベースシステムでは信頼性を獲得するためにハードディスクへの書き込みが原則とされるが、全てのデータをメモリ上に格納し、高速なデータ処理が可能である In-Memory (IM)DB が新しいアプリケーションを開拓している。しかしながら、IMDB はディスク型データベースと比較し容量が小さい。ストレージ容量の制限を考えれば、すべてのトラフィックを書き込むのではなく、ユーザにより選択的に絞り込んだ情報を書き込むことが望ましい。また、高速な書き込み処理の実現と記憶容量の確保・障害によるデータ消失の回避を両立するため、サービス指向型ルータにおけるデータベースは IMDB とハードディスクストレージ DB の階層的なアーキテクチャが必要である。

我々が提案する図 3 に示した DB インサージョンエンジンは、データベースへの書き込みを行うハードウェアであり、高速なメモリ書き込み処理を実現する Stream Input Adopter と、永続化処理を実現する Archiving Engine などで構成される。ホスト PC からの要求により、IMDB もしくは Disk-Based DB の内容を求められるデータ形式に変換後、ホスト PC のメモリへ DMA 転送する。

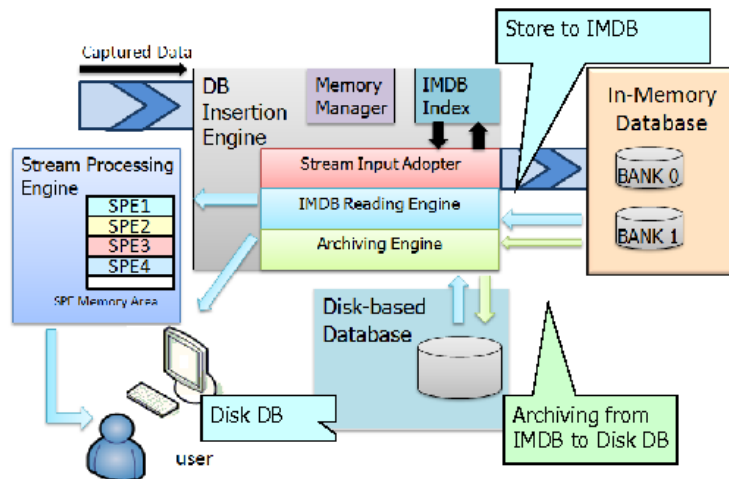


図 3: データベースインサージョンエンジン

2.3.2 IMDB 領域への書き込み手法

トラフィック情報を IMDB へ書き込む際、セグメントという単位で管理され、そのインデックス情報を作成・管理しなければならない。インデックスの作成方法を以下の 2 種類検討した。

インデックス構造 もっとも単純な方法は、図 4 に示した逐次アクセス方式である。インデックスは、データの到着時間のみを利用して作成する。構成が単純でありメモリの RAS-to-CAS 遅延を隠蔽できるため、書き込み処理が高速に実行できるが、一つのセグメントに複数クエリのデータが含まれるため、検索コストが大きくなる。

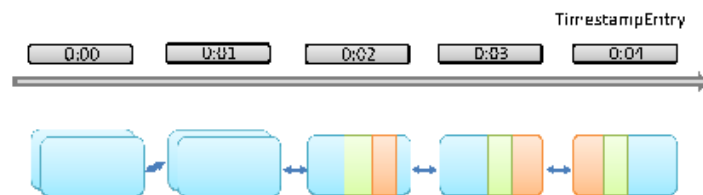


図 4: 逐次アクセス

第 2 の方法は、図 5 で示した Leaf 別逐次アクセス方式である。データを Leaf-ID と時間の両方で管理する。高速な検索が可能であるが、インデックスメモリで管理できるエントリの構成が固定である場合各セグメントは多くて 1 つのトラフィックしか含めることができない。セグメントのサイズを小さくできない場合はメモリ利用効率が低下する。

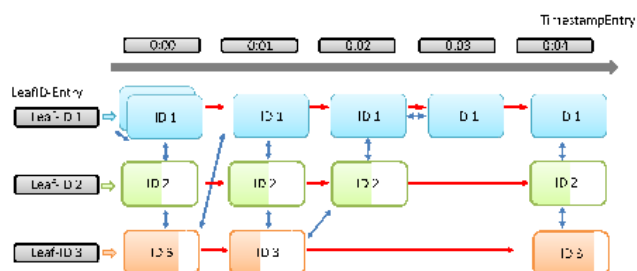


図 5: Leaf-ID 別逐次アクセス

2.3.3 評価

前節で延べた 2 種類の基本的なインデックス構造を評価した。インデックス作成部分を Verilog により設計し、論理規模及び動作遅延を評価した。評価には 5.1 節で述べた 3 方式について専用命令を含む独自マイクロコードで記述し動作遅延を求めた。IMDB メモリへの書き込み処理は専用命令により並列処理されるため、メモリスループットがネットワークスループットよりも十分高いと仮定した。結果、ボトルネックとなるのはインデックス作成部位となる (評価環境の詳細は文献 [2])。また設計環境は正規表現プロセッサと同様である。

表 2 に論理規模及び動作遅延の結果を示す。また、インデックスメモリとしてオフチップメモリを使用する場合、以下の式を用いてインデックス作成に要する所要時間を求めた。スループットと命令サイズの比較を表 3 に示す。

$$\text{所要時間} = (\text{所要サイクル数} + WL \times \text{メモリ書き込み回数} + RL \times \text{メモリ読み出し回数}) \times \text{動作周波数} \quad (1)$$

表 2: 論理規模及び動作遅延

回路面積 (μm^2)	回路面積 (NAND 換算)	動作遅延 (ns)
79872	100091	2.80

表 3: ネットワークスループットの比較

	スループット (Gbps)				命令サイズ (KB)
	オンチップメモリ		オンチップメモリ		
	case1	case2	case1	case2	
逐次アクセス	267	10.2	53.3	0.95	0.92
Leaf-ID 別逐次アクセス	104	3.97	25.2	0.83	1.11

表 3 において、取得を想定している内容を含むパケットでほぼ最小に近いサイズと考えられる 50B のパケットが連続して届いた場合の性能を case1 に、WIDE MAWI プロジェクトで取得された 2009 年 7 月 12 日 14:00-14:15 のトラフィックにおける HTML パケットの平均サイズを利用した場合の性能を case2 に示

す。これらの結果から、1Gbps 以上の性能を獲得するためには、オンチップメモリよりインデックスメモリを構成することが必要である。

本年度は正規表現エンジン、高速なデータベースインサージョンの他にも、高速に流れる混ざり合った複数の TCP コネクションから必要なコネクションのみを抽出するため、コネクションが終了するのを待たずにパケット到着ごとにコンテキストスイッチを利用して抽出の高速化手法 [6] や過去に通過したパケットの処理をキャッシュに登録することで、極めて似たヘッダを持つ後続のパケットを高速に処理するネットワークプロセッサの高速化 [5, 7] などを行った。

3 サービス指向ルータのシミュレータの研究

トラフィック情報は、「ある URL にアクセスしたユーザは、他のどんな URL にアクセスするのか」、「ユーザがある URL にどのくらいの時間滞在していたのか」、「その情報がいつ、どこからネットワーク上に現れたのか」といった、検索サービスや人気調査にとって重要な情報を含んでいる。これらの情報は従来のネットワークシステムでは利用が困難であった。サービス指向ルータではリアルタイムでサービスに必要な情報をネットワークトラフィックから抽出することができる [8]。ただし、前節で述べたハードウェアに関する研究においてはスループットなどの定量的な評価が得られた一方、「抽出した情報量がどの程度になるのか?」「ユーザの行動履歴解析などのアプリケーションの補助をどの程度効果的にサポートできるのか?」等の評価は難しかった。

そこで、我々が提案しているサービス支援型ルータにおける動作検証と有効性を調べるためにシミュレータソフトウェアを開発した [4]。

3.1 シミュレータの設計

シミュレータは、イーサネットのポートを監視し、様々な TCP ストリームの中から正規表現により、アプリケーションが必要とするデータを抽出する。

高速なインターネット回線においてすべての情報を保存するのは非現実的であり、中には提供するサービスに必要な情報も多く存在する。そこで本シミュレータはパケットから取り出したコンテンツのうち必要な部分のみを抽出し保存する。具体的には正規表現における条件抽出を利用して有用な情報を振り分け、部分抽出を行っている。これにより保存データベースの肥大化を最小限に抑えたとともに、サービスの提供時に必要な保存データの解析コストを軽減する。

3.1.1 設計

本シミュレータは Linux のルーティング機能を利用して動作し、その構成を図 6 に示す。

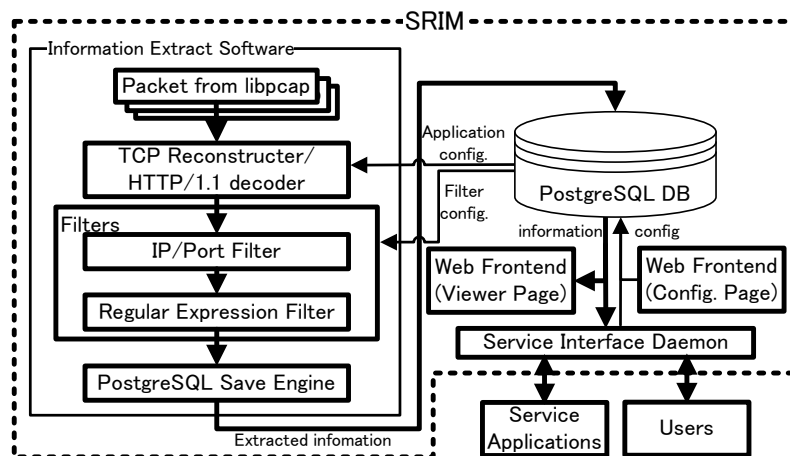


図 6: シミュレータの構成

ルータを通過するパケットは図 6 における Information Extract Software によりコンテンツに変換される。本シミュレータにおいて、まず TCP ストリームの再構築及び HTTP/1.1 のデコードを行う。次に送り元/宛先の IP アドレス (ネットマスク) とポート番号のペア及び正規表現により指定された抽出条件を元に、抽出が必要な場合は正規表現の文字列抽出スペシャルキャラクター () により指定された抽出文字列情報に相当する内容の取得し、PostgreSQL データベースに保存される。また Information Extract Software における抽出条件とソフトウェアの設定も PostgreSQL データベースに保存されている。

サービスアプリケーションやユーザは本シミュレータが動作するマシンの特定のポートで待ち受けている、図 6 中における Service Interface Daemon を経由して保存された抽出情報を取得し、また抽出条件やソフトウェアの設定を変更する。また PostgreSQL に直接接続する Web インタフェースからも抽出情報の参照と抽出条件・ソフトウェアの設定変更が可能である。

3.1.2 Information Extract Software の実装

シミュレータは Linux の eth デバイスを通過するイーサネットパケットを解析し、可読情報の取得及び抽出を行う。本章では具体的な実装方法について述べる。Information Extract Software は C++ において実装を行った。パケットの取得と TCP ストリームの再構築には Libnids ライブラリ [9] を利用した。Linux の eth デバイスをリアルタイムに監視すること、並びに pcap ファイルから対象パケットを取得することが可能である。また TCP の再構築を行う・行わないの指定も可能である。HTTP/1.1 のデコードは RFC2616[10] を参考に chunk encoding ならびに gzip encoding について対応を行った。正規表現による情報抽出には Libboost regex ライブラリ [11] を利用した。データベースには PostgreSQL を利用した。Web フロントエンドは PHP により実装した。

3.1.3 保存情報と設定情報のフォーマット

Information Extract Software により TCP 再構築と HTTP/1.1 のデコード処理をし、抽出された情報は PostgreSQL のテーブルに保存される。抽出データを保存するテーブルの構成を表などの詳細は文献 [4] にまとめた。

情報抽出のための条件も PostgreSQL のテーブルに保存される。このテーブルはサービスアプリケーション及びユーザ指定抽出条件のリクエストを受け付けるアプリケーションと Web フロントエンドにより変更され、Information Extract Software が読み出して利用する。テーブルの構成は文献 [4] にまとめられている。Information Extract Software は TCP ストリームの最大保存サイズ等ソフトウェアの動作に必要な設定情報を持つ。この設定情報も PostgreSQL のテーブルに保存されている。

3.2 評価

本シミュレータはすでに、慶應義塾大学大学院理工学研究科西研究室のゲートウェイマシンにおいて半年以上の間、継続して動作している。本評価では、抽出した情報量を評価するために、通過するパケットから Web ページのタイトルを取得し、それをリストとして出力するアプリケーションを作成した。これは Web ページの人気調査などのサービス支援補助の一例である。

ここでは、有用な情報のみを抽出した場合による容量削減の評価をルータを通過する全トラフィックを取得する場合と比較する。評価において想定したサービスアプリケーションは HTTP 通信のリクエストからアクセスしたホームページの URL を抽出するものである。このサービスを実現するために必要な情報を抽出する正規表現を以下のように作成した。

```
GET (.*) HTTP/1.\d\r\n.*Host: (.*)\r\n
```

学術情報ネットワークにおける 2008 年 5 月平日の日中 (1 秒間) におけるトラフィックを用いた結果を図 7 に示す。

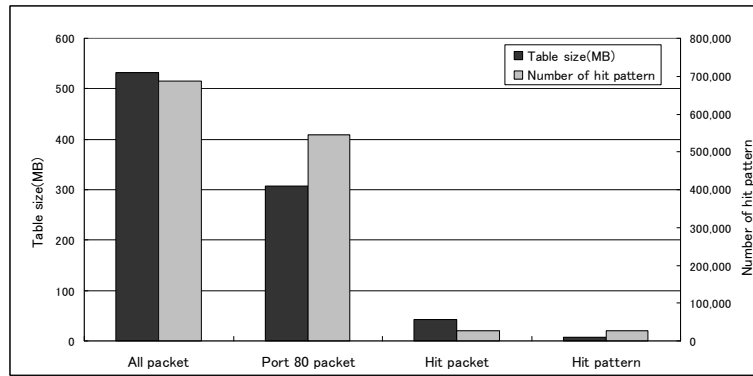


図 7: 学術情報ネットワークにおける保存容量の削減効果

図 7 において Table size(MB) はデータベース内の抽出情報保存に利用しているテーブルの容量を示している。Number of hit pattern は保存された情報の件数を示す。各評価条件において送信元/宛先の IP アドレス及びポート番号、パケットサイズは必ず保存される。All packet は全パケットの内容を加工せずに保存した場合である。この時のテーブル容量は 532MB、件数は 685,970 件であった。Port 80 packet は送信元/宛先ポート番号のどちらかが 80 番であった場合のパケットを内容を加工せずに保存した場合である。この時のテーブルの容量は 307MB、件数は 544,266 件であった。Hit packet は抽出する正規表現を指定し、正規表現にマッチしたパケットの内容を加工せずに保存した場合である。この時のテーブル容量は 42MB、件数は 25,802 件であった。最後に Hit pattern は抽出する正規表現を指定し、正規表現の文字列抽出スペシャルキャラクター () で囲まれた抽出指定部分のみを保存した場合である。この時のテーブル容量は 7MB、件数は 25,802 件であった。

これらの評価結果より、抽出条件にマッチする部分のみを保存した場合、保存に必要なテーブルの容量は全パケットを保存した場合と比較し 1.4%、80 番ポートに絞った場合と比較して 2.4%、抽出条件は指定するがパケット全体を保存した場合と比較して 17.4%の保存容量で想定サービスが提供出来ることが分かった。

4 今後の展望

今年度は、研究計画に従い、ここで報告したように 2 つの課題について研究を行った。今後は、(1) 引き続き、サービス指向ルータの核となる要素技術の開発と評価、(2) パケットストリーム解析を支援する、SQL に基づく宣言的言語処理系を提案し、(3) サービス指向ルータにより実現される代表的なアプリケーション例として行動履歴解析、膨大なサイエンスデータ処理などを実インターネットトラフィックを用いた定量的な評価から示す予定である。そして、最終的に、情報爆発時代における人類の全情報の収集、整理を効率的に行うために、情報の発信・共有・検索・受信に積極的に関わるルータクラウド・インフラストラチャの姿をより具体形に示す予定である。

参考文献

- [1] 永富 泰次, 鯉淵 道紘, 川島 英之, 西 宏章 “パケットストリームの正規表現処理を可能とするネットワークプロセッサ”, 情報処理学会研究報告 Vol.2009-ARC-184 No.21, Aug 2009
- [2] 牧野 友昭, 辻 良繁, 川島 英之, 鯉淵 道紘, 西 宏章, “サービス指向型ルータにおける高速な書き込み機構の提案”, 電子情報通信学会技術研究報告 CPSY2009-23, Vol. 109, No.168, pp.79-84, Aug. 2009
- [3] 川島 英之, 鯉淵 道紘, 西 宏章, パケットストリーム処理における正規表現選択演算を含む問合せ最適化, 電子情報通信学会技術研究報告 CPSY2010, Mar 2010 (to appear)

- [4] 石田 慎一, 原島 真悟, 川島 英之, 鯉渕 道紘, 西 宏章, パケットデータ管理基盤における抽出処理の効率化技法, 電子情報通信学会技術研究報告 CPSY2010, Mar 2010 (to appear))
- [5] Airi Akimura, Hiroaki Nishi, Multi-Context-Aware Cache Accelerating Processing on Network Processors for Future Internet Traffic, Proceedings of the 12th International Conference on Advanced Communication Technology, Feb 2010 (to appear)
- [6] 原島 真悟, 石田 慎一, 西 宏章, “サービス指向型ルータのためのパケット単位の文字列探索”, 情報処理学会創立 50 周年記念 (第 72 回) 全国大会, Mar 2010 (to appear)
- [7] 明村 愛里, 西 宏章, “時間的局所性の傾向を利用したネットワークプロセッサ・アーキテクチャ”, 情報処理学会創立 50 周年記念 (第 72 回) 全国大会, Mar 2010 (to appear)
- [8] Koichi Inoue, Dai Akashi, Michihiro Koibuchi, Hideyuki Kawashima, Hiroaki Nishi, Semantic router using data stream to enrich services, 3rd International Conference on Future Internet (CFI), pp.20-23, Jun 2008
- [9] Libnids, <http://libnids.sourceforge.net/>
- [10] Hypertext Transfer Protocol – HTTP/1.1, <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [11] Boost C++ Libraries, <http://www.boost.org/>

研究成果リスト

著書, 論文

1. Airi Akimura, Hiroaki Nishi: “Multi-Context-Aware Cache Accelerating Processing on Network Processors for Future Internet Traffic”, Proceedings of the 12th International Conference on Advanced Communication Technology, 2010(to appear).
2. 永富 泰次, 鯉渕 道紘, 川島 英之, 西 宏章: “パケットストリームの正規表現処理を可能とするネットワークプロセッサ”, 情報処理学会研究報告 Vol.2009-ARC-184 No.21, Aug 2009, 2009.
3. 牧野 友昭, 辻 良繁, 川島 英之, 鯉渕 道紘, 西 宏章: “サービス指向型ルータにおける高速な書き込み機構の提案”, 電子情報通信学会技術研究報告 CPSY2009-23, Vol. 109, No.168, pp.79-84, 2009.
4. 原島 真悟, 石田 慎一, 西 宏章: “サービス指向型ルータのためのパケット単位の文字列探索”, 情報処理学会創立 50 周年記念 (第 72 回) 全国大会, Mar 2010 (to appear).
5. 明村愛里, 西 宏章: “時間的局所性の傾向を利用したネットワークプロセッサ・アーキテクチャ”, 情報処理学会創立 50 周年記念 (第 72 回) 全国大会, Mar 2010 (to appear).
6. 石田 慎一, 原島 真悟, 川島 英之, 鯉渕 道紘, 西 宏章: “パケットデータ管理基盤における抽出処理の効率化技法”, 電子情報通信学会技術研究報告 CPSY2010, Mar 2010 (to appear).
7. 川島 英之, 鯉渕 道紘, 西 宏章: “パケットストリーム処理における正規表現選択演算を含む問合せ最適化”, 電子情報通信学会技術研究報告 CPSY2010, Mar 2010 (to appear).