

連想メモリを用いた高速なコンテンツセントリックネットワークルータ のハードウェア設計と評価

大岡 睦[†] 阿多 信吾^{††} 井上 一成^{†,†††} 村田 正幸[†]

[†] 大阪大学 大学院情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5

^{††} 大阪市立大学 大学院工学研究科 〒 558-8585 大阪府大阪市住吉区杉本 3-3-138

^{†††} 奈良工業高等専門学校 〒 639-1058 奈良県大和郡山市矢田町 22

E-mail: [†]{a-ooka,inoue.kazunari,murata}@ist.osaka-u.ac.jp, ^{††}ata@info.eng.osaka-cu.ac.jp

あらまし インターネットに代わる革新的なネットワークアーキテクチャである CCN ではキャッシングやマルチキャストなどのコンテンツ中心の通信を活かした画期的な技術が提案されているが、その実現には IP ルータを遥かに上回るパケット処理能力を持ったルータが必要となる。しかし、ルータの全体的な設計に関する議論は乏しく、したがって本稿では CCN ルータのアーキテクチャについて設計・評価を行う。先ずテーブルごとの名前検索とマッチング手法について議論し、確実な通信を実現するためのアルゴリズムを明確にする。次に低遅延で高スループットなパケット処理を実現する手法を提案する。最後に、ルータ全体のハードウェア設計を示し、その性能とコストの評価を行う。

キーワード 新世代ネットワーク, コンテンツセントリックネットワーク (CCN), ネットワークアーキテクチャ, ルータハードウェア, CAM, ブルームフィルター

Hardware Design and Evaluation of a High-speed CCN Router Using CAM

Atsushi OOKA[†], Shingo ATA^{††}, Kazunari INOUE^{†,†††}, and Masayuki MURATA[†]

[†] Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka, 565-0871, Japan

^{††} Graduate School of Engineering, Osaka City University
3-3-138 Sugimoto, Sumiyoshi-ku, Osaka-shi, Osaka 558-8585, Japan

^{†††} Nara National College of Technology, 22 Yata-cho, Yamatokoriyama, Nara 639-1058, Japan

E-mail: [†]{a-ooka,inoue.kazunari,murata}@ist.osaka-u.ac.jp, ^{††}ata@info.eng.osaka-cu.ac.jp

Abstract Content-centric networking (CCN) that is an innovative network architecture requires routers with performance far superior to that offered by today's Internet routers. There are, however, few router-level designs incorporating all the necessary components. The design and evaluation of a complete router is the primary contribution of this paper. We provide a concrete hardware design for a router model incorporating two entities that we propose. Our contributions are (1) presenting a proper algorithm for looking up and matching name addresses in CCN communication, (2) proposing a method to process CCN packets in a way that achieves high throughput and very low latency, and (3) demonstrating performance and cost on the basis of a concrete hardware design.

Key words Future Networks, Content-Centric Networking, Architecture, Router Hardware, CAM, Bloom Filter

1. はじめに

近年、コンテンツセントリックネットワーク (Content-Centric Networking; CCN) が注目され、[1], CCNx, NDN, PURSUIT, SAIL などのプロジェクトを始めとして世界中で研究が活発に行われている [2]. しかし、CCN の実現には多くの課題を解決する必要がある。先ず、CCN で用いられるコンテンツアド

レスの運用には新たな名前解決・ルーティング機構が必要となる。次に、Interest パケットと Data パケットによるパン屑型ルーティングはネットワークがマルチキャストを備えるための特殊なプロトコルを不要とする一方で、従来よりも遥かに高いテーブル更新頻度を要求する。更に、コンテンツ配布型のアプリケーションに適したキャッシングの仕組みは特に大きな議論の対象となっており、どのように少ないリソースで効果的にコ

コンテンツをキャッシュするかについて多くの研究が成されている [3], [4]. 他にも, セキュリティやモビリティ, 現在の IP ネットワークから CCN への移行に関する問題など, CCN の実現方法が具体的に検討されつつある.

特に重要な課題は CCN ルータの実現であるが, その実現方法や具体的な性能は明らかではない. CCN の通信を実現するためには, CCN ルータの具体的な実装方法を示すことが必要不可欠である. また, 現実的なルータの能力もネットワーク性能や様々な提案手法の現実性を評価する上で重要である. しかしながら, 既存研究のほとんどが CCN の要素技術に関するものばかりで, CCN ルータに関する包括的な研究は極めて少ない. Caesar [5] は FIB, DiPIT [6] や NameFilter [7] は PIT に焦点を当てており, 高速で低コストなフォワーディング手法を提案しているが, 偽陽性を取り除けていない. 偽陽性の無いトライ木を用いた名前検索機構には NCE [8], ENPT [9], MATA [10] がある. これらはパイプライン処理によって高いスループットを実現しているが, レイテンシを小さくすることが困難であるという欠点がある. 包括的な研究には [4] や [11] があるが, 必要なリソースの理論的な分析しか行われておらず, 現実的なルータの能力を明確にするには不十分である.

本研究の目的は CCN ルータの全体像を明確にして, その実現可能性と具体的な性能を示すことである. 2 章では CCN の仕様を確認し, 実装する上で議論が不十分な名前検索アルゴリズムについて考察する. 3 章では CAM ベースの名前検索機構 (NLE) と要求カウント機構 (ICE) を持つルータアーキテクチャを提案し, 続く 4 章でそのハードウェア設計を示す. 5 章ではハードウェアのメモリ構成に基づいて, 主要である NLE のコストやスループットに関する評価と考察を行う.

2. CCN

2.1 CCN における通信

我々は NDN [3] や CCNx [12] の通信モデルおよびルータモデルを採用する. CCN では, すべてのコンテンツのデータに *name* と呼ばれるアドレスが割り当てられ, コンテンツを要求する *Interest* パケットと, コンテンツを供給する *Data* パケットを交換する, 要求・供給型通信が行われる. 通信において, 要求の時に指定される *name* は, 供給の時に指定される *name* の部分名となっても構わない. 例えばテキストデータを取得したいコンテンツ消費者が “/txt/A” を要求したとして, 供給者はバージョン情報やセグメント情報を付加した “/txt/A/v1/s1” という *name* の *Data* パケットを返送するかもしれない. 以下ではこのような動的な名前付けを *active naming* と呼ぶ. また, このとき “/txt/A” と “/txt/A/v1/s1” が, 双方別々のコンテンツを指すことも考えられる. 以下では, 何らかのコンテンツを指し示す *name* のプレフィックスも別のコンテンツを指し示すことを *name sibling* と呼ぶ. ただし, *name sibling* が許容されるかどうかは明確にはされていない.

CCN の通信を実現するためのルータは, フォワーディング情報を持つ FIB, *Data* 返送待ちの *Interest* の情報を持つ PIT, キャッシュを持つ CS の 3 つの基本的なテーブルを持つ [1], [3].

これらのテーブルは *name* を検索キーとして対応するエントリの情報を返す, 連想配列と同様のデータ構造を持つ.

2.2 名前検索アルゴリズム

CCN が保持する各テーブルは, 単一のキーと単一のエントリが対応するような単純な連想配列ではなく, *prefix match* や *active naming* のために 1 つの検索キーが複数のエントリにマッチしうる複雑な検索テーブルである. よって, テーブルの実装においてはそのテーブルが実現する検索アルゴリズムが実際のネットワークにおいて機能するかどうかの考察が必須である. しかし, 検索対象とするパケットおよびテーブルごとにどのようなマッチングが可能かについては考察がなされたことがない. したがって, 各テーブルおよび各パケットごとに, どのような検索・マッチングアルゴリズムを採用すべきかについて議論する. 以下では簡単のために *Interest* パケットと *Data* パケットを単に “*Interest*” と “*Data*” と表記する.

2.2.1 マッチ手法と選択手法

各組み合わせごとの考察に先立ち, 実行可能な検索アルゴリズムについて考察する. テーブルの検索アルゴリズムは, テーブル内のエントリがどのような検索キーとマッチさせるかを決定する *マッチ手法* と, そのマッチした複数のエントリからどれを選択すべきかを決定する *選択手法* から構成される. 以下では, 検索キーを K_S , エントリキーを K_E と表記し, それらのキー K のプレフィックスを $P(K)$ と表す.

マッチ手法は (1) $K_S=K_E$ でマッチとする *Exact Match (EM)*, (2) $P(K_S)=K_E$ でマッチとする *Search-Key Prefix Match (SPM)*, (3) $K_S=P(K_E)$ でマッチとする *Entry-Key Prefix Match (EPM)*, (4) K_S と K_E が, 完全に一致するか, 一方が他方のプレフィックスと一致した場合にマッチとする *Both-Keys Prefix Match (BPM)* の, 4 つの手法が候補として考えられる.

EM 以外のマッチ手法は複数エントリにマッチする可能性があるため, 選択手法を考察する必要がある. 従来の IP と同様の SPM を除いて, 最長一致検索 (LPM) だけでは不十分であり, EPM や BPM では “/txt/A” に対して “/txt/A/v1/s1” でも “/txt/A/v4/s6” でもマッチしうるため, LPM 以外の優先度による選択手法を設ける必要がある. 例えばエントリの登録時間や要求回数などはその優先度の候補となりうる. また, マッチしたエントリすべてを選択する手法も考えられる.

2.2.2 各テーブルごとの検索アルゴリズム

各テーブルがどのマッチ手法・選択手法を採用すべきかに関して, エントリの集約の観点から考察を行う. FIB はエントリの集約を行うために従来の IP ルータと同様の SPM と LPM の組み合わせが適している. PIT と CS に関しては, *name sibling* があるか否かで場合分けする必要がある. *name sibling* が禁止されている場合, PIT はプレフィックス関係にある複数の *Interest* を集約することができる. そのため *Interest* のマッチ手法には長い *name* も短い *name* も集約可能な BPM が有用である. *Data* のマッチ手法には *active naming* のため SPM を採用しなければならない. 次に CS は, 到着した *Interest* が *active naming* によるものである場合に, CS 中の *Data* の *name* とプレフィックス関係にある *Interest* をヒットさせることでキャッシュヒット率

の向上が可能である。よって、Interest のマッチ手法には Data より短い name を持つ Interest でもマッチさせる EPM が良い。Data のマッチ手法は完全一致である必要があるため EM を採用する必要がある。name sibling が許容されている場合、曖昧な name による要求が name sibling と active naming のどちらによるものか判別できないため、PIT に対する Data のマッチ手法を除いて EM でなければならない。いずれの場合も、複数のエントリにマッチしうるマッチ手法には適切な選択手法が必要となるが、余白の都合上議論を省略する。以上の議論から、PIT と CS に対して実現しうるマッチ手法および選択手法の候補を表 1 に示す。ただし、FIB に対する Interest のマッチ手法および選択手法は LPM であるため省略している。この内、name sibling に対応可能な手法は (I) のみである。

表 1 名前検索アルゴリズムのまとめ (マッチ手法/選択手法)

	CS		PIT	
	Data	Interest	Data	Interest
(I)	EM/-	EM/-	SPM/LPM など ¹	EM/-
(II)	EM/-	EM/-	SPM/-	BPM/優先度選択
(III)	EM/-	EPM/優先度選択	SPM/LPM など ¹	EM/-
(IV)	EM/-	EPM/優先度選択	SPM/-	BPM/優先度選択

¹ FIFO (first in, first out) やすべてを選択する手法も利用可能。

3. ルータアーキテクチャ

固定長の IP アドレスの検索と異なり、可変長の名前検索処理は負荷の大きな処理である。したがって、本稿では CAM とブルームフィルタを導入した名前検索テーブルとして、Name Lookup Entity (NLE) を提案する。CAM を用いることで偽陽性の無い高速検索を実現しつつ、ブルームフィルタで CAM の検索負荷を削減する。また、ルータの資源消費が特に大きいとして懸念されるキャッシュに対応するために、キャッシュする価値のあるコンテンツを選別するための Interest Counting Entity (ICE) を提案する。

検索アルゴリズムには、ブルームフィルタと CAM の組合わせに適した検索アルゴリズムとして、表 1 の (I) を採用する。ここで、name sibling を無視すれば、Interest を PIT で検索する場合を除くすべてのマッチ手法を SPM に揃えることができ、実装を単純化できる。name の管理の複雑性も考慮して、以降では name sibling は存在しないものと見なす。また、CAM は 3 値ではなく 2 値で良いため、更なるコスト削減が期待できる。

提案する CCN ルータのアーキテクチャを図 1 に示す。パケットは図 1 上部の Face の内どれかに到着する。Face に到着したパケットはまず Packet Parser に送られる。ここで、Interest パケットの場合は name のみ、Data パケットの場合は name と data 部分に分けて、NLE と ICE に送られる。NLE におけるヒット情報から読み込むストレージを決定し、パケットのフォワーディングなどの適切な処理が行われる。ICE は要求回数の少ない Data のキャッシュを避けるために、name ごとに Interest による要求の回数をカウントし、要求回数が一定数以上の場合のみキャッシングを行う。転送すべきパケットがある場合は、そのパケットを適切な Face 宛に転送する。

3.1 NLE

名前検索処理の実装は、従来のハッシュテーブルによる手法

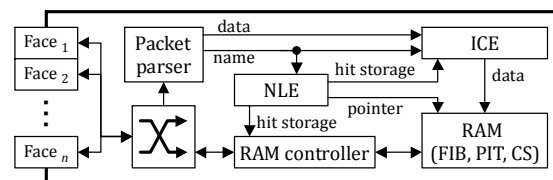


図 1 CCN ルータのアーキテクチャ

ではなく、CAM をメインとして用いた手法を考える。従来のハッシュテーブルを用いた手法では偽陽性を取り除けず、CCN ルータが正しくパケットをフォワーディングできない恐れがある。ハッシュテーブルを用いた手法で偽陽性をなくすためには、コンポーネント階層ごとに文字列が正しく格納されているかを確認しなければならないため、何度もハッシュテーブルの検索を行う必要があり、処理遅延が大きくなる。一方、CAM をメインとして利用する手法は、1 クロックで偽陽性の無い検索を行うことができるが、そのコストが非現実的となるだろうという推測から敬遠されている。そこで、本研究では 1 つの大きな CAM を用いるのではなく、複数の小容量 CAM に分割することによってコストを抑えつつ、DLB-BF を組み合わせて高速な検索機構を実装する。DLB-BF は、[13] で提案されている手法である。1 つの巨大なブルームフィルタを数学的に等価な方式で複数のメモリに分割して、階層化された name に対してもパイプラインによる高速なプレフィックスマッチング処理を実現することができる。

CAM のエントリ幅は固定長だが、name は可変長であるため、CAM エントリの幅より長い name への対処が必要である。したがって、長い name を登録する場合はエントリ幅で分割してそれぞれ別々のエントリとして CAM に格納する。これは図 2 のように、Short Name(SN), Partitioned Name (PN), Partitioned Prefix(PP) の 3 種類で定義する。SN は分割する必要が無い name に対して用いられるエントリであり、PN と PP は分割する必要がある長い name に対して用いられるエントリである。PN と PP によって疑似的なツリー構造の検索を可能にしており、ツリー構造で言えば PN は葉、PP は節として機能する。ただし、CAM エントリの幅を $W[\text{bit}]$ 、CAM エントリのアドレス値のビット長を $L[\text{bit}]$ とする。

CAM のエントリは RAM の領域と対応しており、RAM には格納されたデータや制御用の変数などが格納される。SN は単純なエントリであり、CAM には長さが $(W - 1)[\text{bit}]$ を超えない完全な name が格納されており、RAM にはそのエントリと対応するデータが格納される。 $(W - 1)[\text{bit}]$ を超える長さの name は、最後の部分名だけが PN として格納され、その PN がデータを格納する。それ以外のプレフィックス部分は PP として格納されるが、各 PP または PN は自分の直前の PP の CAM におけるアドレスを Address フィールドに保持しておくことで、部分名的一致によるエントリの衝突を防ぐ。また、同じプレフィックスを共有する name がある場合、それらはプレフィックスが一致する最後の PP で集約される。PP は集約された name の数をカウントした値を RAM に保持しておき、エントリ追加時に集約が行われた場合にはカウントを 1 増やし、エントリ削除時

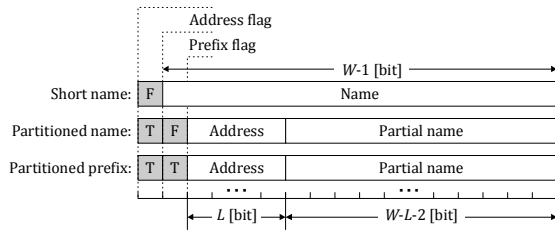


図2 CAM エントリの定義

にはそのカウントを1だけ減らす。もしカウントが0の場合はそのエントリを削除する。

3.2 Interest Count Entity (ICE)

ICE は要求回数の少ないコンテンツのキャッシュを避けるために用いる。コンテンツの人気度は Zipf 則に従っているとされる。例えば [14] によると、キャッシュの保持時間を数時間から数日とすれば、6, 7 割を超えるトラフィックが1度か2度しか要求されないコンテンツに対するものである。一方、何度も要求があるようなコンテンツは、コンテンツ全体の1~2割以下であり、それらの一部のコンテンツによってトラフィックの3~4割が占められている。そこで、name ごとに要求の回数を記憶しておき、要求が閾値を超えたときに初めてキャッシュを行うことで、ワнтаイマーのコンテンツを無視して、キャッシュする意味のあるコンテンツだけをキャッシュする。ICE は、その実現のために name ごとの要求回数を記録する。Data が PIT でヒットした場合でも、ICE で要求回数が閾値を超えていなければキャッシュされない。この機構により、キャッシュ容量を節約しつつ高いキャッシュヒット率を実現できる。

4. ハードウェア設計

4.1 NLE

図3にNLEのハードウェアアーキテクチャを示す。図中の記号において、 W はCAMのエントリ幅、 W_1, W_2 は W に応じて決まる一度に検索可能なビット幅、 l は長さやインデックス値を表すために必要なビット幅、 s, p はRAMのアドレス値を表すために必要なビット幅、 M はPartial Name Bufferの個数、 N はPartial Prefix Bufferの個数、 D はCAMの個数である。

partial name とは、CAMのエントリ幅より長いnameを分割した部分名を意味する。partial prefix とは、partial nameを区切り文字に基づいて分割した部分プレフィックスを意味する。基本的には、まず前処理として、Partial Name ParserでCAMのエントリ幅より長いnameを分割する。ただし、この処理は十分に長いエントリ幅を設定しておくことで大部分のnameに対して省略できる。次に、それを更にPartial Prefix Parserで区切り文字に基づいて分割する。その後、各プレフィックスに対してDLB-BFでブルームフィルタの検索を行い、そこでヒットしたものだけCAMで検索を行う、という手順で検索を行う。分割の段階でバッファ溢れが起きたとしても、その時点でバッファに格納されているプレフィックスのみで一度検索を行い、ヒットしなかった場合は続きから処理を再開する。CAMでの検索には、図2で定義したように、直前のプレフィックスとなるpartial nameのアドレスを付与して検索するため、nameの分割によるエントリの衝突は発生しない。CAMの検索結果アドレ

スは、検索処理機構外部のRAMへの参照(pointer)を保持するメモリRAM-Pointer Memoryのアドレス値である。最終的な返り値は、RAM-Pointer Memoryに格納されたpointer値となる。

重要な変数として、CAMのエントリ幅によって決まる W, W_1, W_2 を考察する。partial nameの分割処理を省略して処理を高速化するためには、nameが可能な限り W_1 以下となるように W の値を選択するのが望ましい。しかし、 W の値が大きすぎるとCAMにnameを格納するときに無駄なスペースが出てくるため、そのトレードオフについて考える必要がある。

このアーキテクチャの利点は、大容量のCAMを用いるのではなく、小容量のCAMを複数個用いる点である。大容量のCAMは複数の小容量CAMで同じだけの容量にしたものと比較して市場価格が高く、検索時の消費電力も大きい。よって、複数の小容量CAMを組み合わせてコストおよび消費電力を低減しつつ、並列処理を可能にすることでスループット増加を図る。加えて、DLB-BFを用いてnameを階層ごとに分割することでBCAMによるLPMを可能にし、一般的なTCAMよりも必要なコストを下げるができる。

4.2 ICE

ICEのハードウェア設計を図4に示す。ICEはnameをキーとしてInterestによる要求回数を保持する単純なハッシュテーブルとして機能する。更に、ハッシュ衝突に備えて完全なnameも保持する。

ICEのパケットごとのアルゴリズムは次のようになる。Interest到着時には、対応するnameのエントリに保持されているカウント値Counterを増加させる。対応するエントリが存在しない場合、またはnameが異なる場合は新しいnameのエントリで上書きする。既存のエントリの要求回数を上書きすることを許容することで、疑似的にキャッシュのタイムアウトを実現することができる。一方でData到着時にはカウント値が閾値を超える場合にのみそのDataをCSでキャッシュする。

ICEの最も大きな利点は、キャッシュする意味のないキャッシュを無視できる点であるが、更に閾値を動的に変更することによってキャッシュ容量に対するキャッシュヒット率を調整しうる。閾値を固定してキャッシュする場合、トラフィックが多い環境下では必然的にキャッシュされるパケット数は増加する。すると、比較的人気の無いコンテンツによってキャッシュが占有され、人気の高いコンテンツはキャッシュの置き換えの多発によって溢れてしまう。これは、必要となるキャッシュ容量の増加またはキャッシュヒット率の低下を招く。しかし、閾値を可変にすれば、トラフィックに応じて閾値を変化させることで、適切なキャッシュヒット率を保てる。しかし、閾値はその数だけ重複したInterestの受信を強要するため、大きな閾値はキャッシュの効果を低下させる要因となる。一方、閾値を極端に小さい値にした場合、人気のあるコンテンツよりも人気の無いコンテンツの方が数が多いため、キャッシュの置き換えが多発してキャッシュヒット率が下がってしまい、結果的に冗長なトラフィックが増加する可能性が考えられる。したがって、冗長なトラフィックの削減量を最大化できるような適切な閾値を選択する方法を考えなければならない。

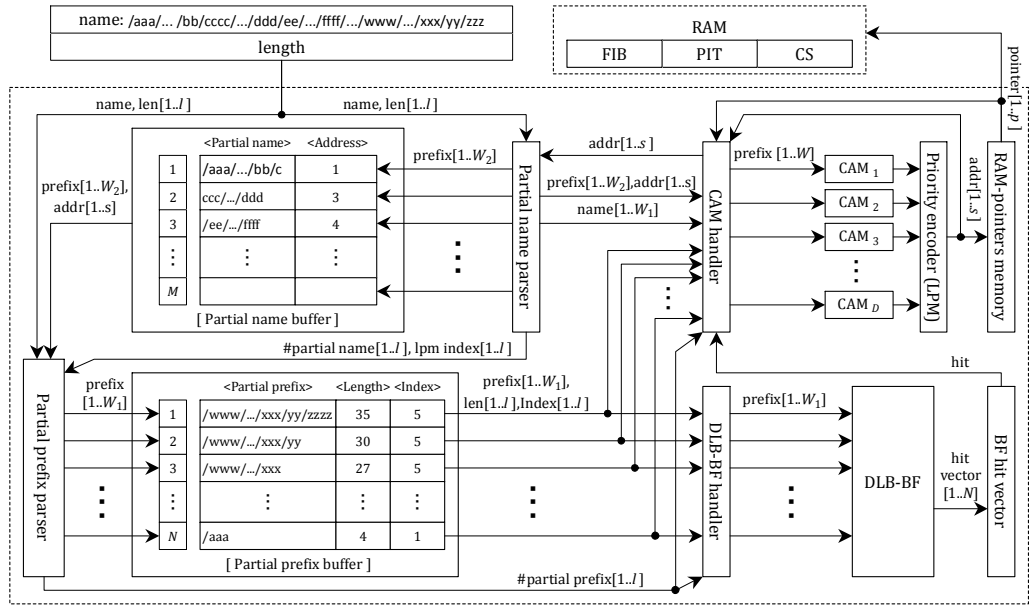


図3 NLEのハードウェア設計

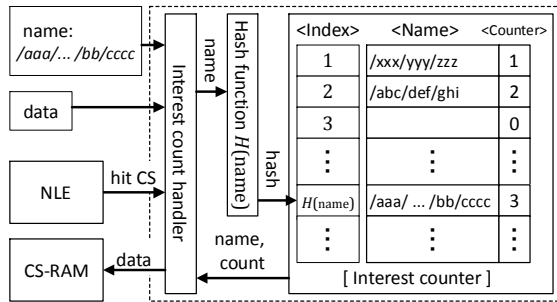


図4 Hardware Design of ICE

5. 評価と考察

CAMを用いたCCNルータの実現可能性を明らかにするために、必要なメモリ容量とスループットの評価と考察を行う。メモリ容量の拡張性のボトルネックは明らかにCAMであり、CAMの保有可能なエントリ数に応じてRAMの容量も決まるため、評価はNLEを中心に行う。比較のために、[4],[10]と同等の仮定として、平均パケットサイズを256B、平均Interestサイズを40B、平均Dataサイズを1500B、テーブルの保有するエントリ数を10Mとする。また、[8]の調査に基づき、99%のnameは40Bを超えない、かつ6階層以下であると想定する。

5.1 メモリ容量の評価

図3に示すように、NLEは2つのバッファPartial Name Buffer, Partial Prefix BufferとDLB-BF、およびCAMによって構成されている。まず最も重要なのはCAMのエントリ幅 W であり、最初の仮定から $W = 40[\text{Byte}]$ と置く。このとき、最大1500B長のnameを格納するために必要なPartial Name Bufferのバッファ数は38個以上となるが、現実にはそのような長すぎるnameは存在しないと想定してバッファ数 $M = 32$ とする。また、Partial Prefix Bufferのバッファ数 N は階層数によって決まり、現在のURLが最大70階層であることを考慮して、 $N = 64$ とする。このとき、Partial Name BufferおよびPartial prefix Buuferのバッファ容量は、それぞれ10[Kbit], 22[Kbit]となり、バッファを

SRAMで実装するとしてもその容量は妥当であると言える。今後は更にパイプライン処理に伴う回線コストを考慮に入れて実現可能性を模索していく必要がある。

DLB-BFは実現したい誤り率 α によって必要な容量が変化する。ここで、DLB-BF中に格納されているエントリ数を n 、DLB-BFの合計ビット数を m 、ハッシュ関数の数を k とすると、DLB-BFは通常のBloom Filterと数学的に等価であるため、 $\alpha = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \simeq \left(1 - e^{-\frac{kn}{m}}\right)^k$ と計算できる。 α を最小にする k の値は $k = \frac{m}{n} \log 2$ の時であり、 $\frac{m}{n} \simeq 4.8 \times x$ となる。これは、誤り率を10分の1にするためには、保有するエントリあたり約4.8[bit]を割り当てるべきであることを意味する。ここで、 $\alpha = 10^{-6}$ 、 $n = 10M$ とすると、DLB-BFに必要なメモリ容量は288MBとなる。要素の削除のためにカウンティングフィルタの方式を導入して各エントリに16bitを割り当てると、更に4.6GBの容量が必要となる。これをSRAM上で実装する場合、SRAMのコストは1USD/MB[15]であるため、合計で4,600USD必要となる。

CAMは $W = 40[B]$ 幅のエントリを10M個保有するため、必要なメモリ容量は3.2Gbitとなる。CAMのコストは1Mbitあたり1USDであるため、合計で3,200USD必要となる。従って、NLEのコストは約7,800USDであると言える。

5.2 スループットの評価

スループットは、各バッファの読込・書込処理に伴うCPU処理とバッファ・メモリアクセスに必要な時間に基づいて計算する。各バッファが逐次的処理に従った場合の、各メモリ構造に対する読込・書込処理のアクセス回数を表2に示す。ただし、PNBはPartial Name Buffer、PPBはPartial Prefix Bufferの略であり、 α はBFの誤り率、 \bar{m} は1つのnameがpartial nameに分割されたときの平均数、 n_i を i 番目のpartial nameから生成されるpartial prefixの個数と定義する。また、 $S(m)$ と $T(m)$ を $S(m) = \sum_{i=1}^m n_i$ 、 $T(m) = \left\lceil \frac{S(m)}{N} \right\rceil$ と定義する。

表 2 名前検索処理における各メモリの読込・書込アクセス平均回数

	PNB		PPB		DLB-BF		CAM	
	R	W	R	W	R	W	R	W
検索	\bar{m}	\bar{m}	$T(\bar{m})$	$S(\bar{m})$	$T(\bar{m})$	0	$\bar{m} + \alpha S(\bar{m})$	0
追加	\bar{m}	\bar{m}	0	0	0	1	\bar{m}	1

表 2 に基づいて、平均検索時間 T_L と平均追加時間 T_A を計算する。TCAM のアクセス時間を 4ns, SRAM のアクセス時間を 0.45ns, メモリアクセスに伴う処理時間を 1.0ns と仮定すると、 $T_L = 12.25[\text{ns}]$, $T_A = 14.35[\text{ns}]$ と計算できる。逆数を取り、パケットサイズの平均が 256Byte であるという仮定を用いてスループットを計算すると、逐次処理における平均検索速度は 163Gbps, 追加速度は 139Gbps となる。パイプライン処理によって高速化を実現した MATA-NW [10] の処理遅延が $100\mu\text{s}$, スループットが 127Gbps であるのに対して、提案手法は高速な処理が実現可能と分析できる。更に、複数の name を同時並列的に処理する手法により、このスループットは更に改善しうる。

5.3 実現可能性の考察

本アーキテクチャの評価結果から、特に CAM の容量の実現が困難であることが分かる。100Mbit を超える容量の CAM は現存せず、その容量の CAM を複数使用としても、現在の TCAM の消費電力が 1kW/Mbit であることを考慮すると、3.2GB の CAM に必要な電力は約 3 kW となり、これは既存の実装における消費電力とは桁違いに大きく、現実性に欠ける。

また、10M 個のエントリの仮定は将来的に更に大きくなりうるという問題もある。ウェブサイト数は [16] によると 10 億に達そうとしており、FIB に必要なエントリ数もそれに従いうる。PIT に関しても、RTT = 100ms の環境下で 40Gbps のトラフィックを収容するためには、2M 個のエントリが必要であり、5 ポート分の容量しか収容できない。CS で 10M のエントリ分のキャッシュが可能とした場合、ICE の効果が最大限に発揮されたとしても、収容可能なファイル量は 1 日間の都市レベルのトラフィックと同等である [14]。

したがって、CCN ルータのメモリ容量の拡張性を確保することが課題となる。本研究で提案するルータは低速だが拡張性の高いハッシュテーブルを利用することも容易であるが、検索時間を犠牲にしない方法として TCAM ではなく BCAM を用いる方法も考えられる。TCAM の容量は 16T/cell であるのに対して BCAM は 10T/cell であり、より大きなメモリ容量を実現しうる。また、CAM 技術自体も成長を続けており、それによって拡張性の課題が緩和されることが期待できる。

6. 結論と今後の課題

本稿では CCN の実現可能性を示すために、具体的な CCN ルータのハードウェアアーキテクチャに焦点を当てて設計と評価を行った。その中で、CAM を用いることで高速かつ偽陽性の無い NLE を実現し、また、ルータの限られた資源を有効活用するためにキャッシュをより効率化する ICE を提案した。複数の小容量 CAM と DLB-BF を併用することで CAM のメモリ容量の拡張性に対処しつつ、低遅延かつ高速なパケット処理が可能なルータのハードウェア設計を示した。

今回の評価で CAM の拡張性の検討が不十分であることが明

らかとなったため、今後の研究ではハッシュテーブルの併用などを考慮してその問題の解決に取り組む。更に、ハードウェア上で実際にルータの実装を行うことで、実用的なスループットやコストを示す予定である。その上で、パケットやバッファ制御における並列化処理の実装も考察していく必要がある。最終的には、それらの評価結果に基づいたネットワークレベルでの現実的な評価および分析を行う予定である。

謝辞

本研究成果の一部は総務省・戦略的情報通信研究開発推進制度 (SCOPE) の支援による。

文 献

- [1] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard, "Networking named content," Proceedings of the ACM CoNEXT 2009, pp.1–12, Dec. 2009.
- [2] B. Ahlgren, C. Dannowitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-centric networking," IEEE Communications Magazine, vol.50, no.7, pp.26–36, July 2012.
- [3] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J.D. Thornton, D.K. Smetters, B. Zhang, G. Tsudik, K. Claffy, D. Krioukov, D. Massey, C. Papadopoulos, T. Abdelzaher, L. Wang, P. Crowley, and E. Yeh, "Named data networking (NDN) project," Oct. 2010. <http://named-data.net/techreport/TR001ndn-proj.pdf>
- [4] D. Perino and M. Varvello, "A reality check for Content Centric Networking," Proceedings of the ACM SIGCOMM workshop on Information-centric networking, pp.44–49, Aug. 2011.
- [5] M. Varvello, D. Perino, and J. Esteban, "Caesar: a content router for high speed forwarding," Proceedings of the 2nd edition of the ICN workshop on Information-centric networking, pp.73–78, Aug. 2012.
- [6] W. You, B. Mathieu, P. Truong, J. Peltier, and G. Simon, "DiPIT: A distributed bloom-filter based PIT table for CCN nodes," Proceedings of the 21st ICCCN 2012, pp.1–7, July 2012.
- [7] Y. Wang, T. Pan, Z. Mi, H. Dai, X. Guo, T. Zhang, B. Liu, and Q. Dong, "NameFilter: Achieving fast name lookup with low memory cost via applying two-stage bloom filters," Proceedings of the IEEE INFOCOM 2013, pp.95–99, April 2013.
- [8] Y. Wang, K. He, H. Dai, W. Meng, J. Jiang, B. Liu, and Y. Chen, "Scalable name lookup in NDN using effective name component encoding," Proceedings of the IEEE 32nd International Conference on Distributed Computing Systems 2012, pp.688–697, June 2012.
- [9] H. Dai, B. Liu, Y. Chen, and Y. Wang, "On pending interest table in Named Data Networking," Proceedings of the ACM/IEEE 8th Symposium on Architectures for Networking and Communications Systems 2012, pp.211–222, Oct. 2012.
- [10] Y. Wang, Y. Zu, T. Zhang, K. Peng, Q. Dong, B. Liu, W. Meng, H. Dai, X. Tian, Z. Xu, H. Wu, and D. Yang, "Wire speed name lookup: a GPU-based approach," Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation, pp.199–212, April 2013.
- [11] S. Arianfar, P. Nikander, and J. Ott, "On content-centric router design and implications," Proceedings of the ACM Re-Architecting the Internet Workshop, pp.1–6, Nov. 2010.
- [12] , "CCNx," 2013. <http://www.ccnx.org/>
- [13] H. Song, F. Hao, M. Kodialam, and T.V. Lakshman, "IPv6 lookups using distributed and load balanced bloom filters for 100Gbps core router line cards," Proceedings of the IEEE INFOCOM 2009, pp.2518–2526, April 2009.
- [14] F. Guillemin, B. Kauffmann, S. Moteau, and A. Simonian, "Experimental analysis of caching efficiency for YouTube traffic in an ISP network," Proceedings of the 25th International Teletraffic Congress, pp.1–9, Sept. 2013.
- [15] S. Iyer, R.R. Kompella, and N. McKeown, "Designing packet buffers for router linecards," IEEE/ACM Transactions on Networking, vol.16, no.3, pp.705–717, June 2008.
- [16] "netcraft," Dec. 2013. <http://www.netcraft.com/>