

# CC/PP によるコンテンツネゴシエーションの効果と性能

安田 絹子<sup>†1</sup> 浅田 卓哉<sup>†2</sup> 萩野 達也<sup>†3,†4</sup>

本論文では、プロトタイプ実装とその測定結果に基づいて、CC/PPによるコンテンツネゴシエーションの効果と影響について述べる。Webクライアントの端末能力やユーザ嗜好が多様化するにつれ、Webコンテンツを個々のクライアントの属性に合った最適な形で提供するためのコンテンツネゴシエーション機構の必要性が高まっている。CC/PPはこのような機構を提供するための新しい仕様であり、XMLやHTTP Extensionなどの汎用技術を基盤としたオープンな設計を大きな特徴としているが、まだ仕様策定の段階であるためその効果は明らかになっていない。本研究では、CC/PPに対応したWebブラウザとCC/PPに基づいてコンテンツ変換を行うプロキシを開発し、CC/PPの性能と有効性の評価を行った。クライアント属性の2種類の転送方法を比較した結果、プロキシからクライアントまでのネットワークが現在の携帯電話程度の帯域および遅延を持つ場合は間接参照の方が性能が良いが、それよりも高速の場合は直接参照の方が性能が良くなる可能性があることが分かった。さらに、平均的な実験条件においてコンテンツ変換によるトラフィック削減効果がCC/PPのオーバヘッドを上回ることを示し、CC/PPのコンテンツ変換への適用が性能的に有効であることを実証した。

## Effects and Performance of Content Negotiation Based on CC/PP

KINUKO YASUDA,<sup>†1</sup> TAKUYA ASADA<sup>†2</sup> and TATSUYA HAGINO<sup>†3,†4</sup>

We describe our investigation of the effectiveness of Web content negotiation on our CC/PP implementation. CC/PP is a proposed specification of a user-side content negotiation framework to comply with varied clients; with respect to platform capabilities and user preferences. CC/PP is based on common technologies such as XML and HTTP Extension and is expected to provide a generic content negotiation solution, but neither its performance nor effectiveness has ever been shown. We have implemented a CC/PP capable Web browser and a CC/PP proxy, and measured the performance of CC/PP and various different settings. The result shows that CC/PP with *indirect reference* can achieve better performance in terms of retrieval time when the network connection from the client to the proxy is as slow as a today's cellular phone. On the other hand, when the connection is faster than that, the use of *inline encoding* proves better performance in ordinary cases. The observed traffic with content conversion using CC/PP is smaller than the case without CC/PP for all tested environments, and the retrieval time is better or comparable for cellular phone clients. The result confirms that CC/PP is an effective solution for general content negotiation.

### 1. はじめに

Web( World Wide Web )の幅広いユーザ層への普及と、インターネット接続可能な機器の多様化にとも

ない、様々なクライアントに適切な形で Web コンテンツを提供するための汎用的なコンテンツネゴシエーション機構が求められている。特に、限られた表示能力と資源を持つ携帯電話などの端末の増加や、音声などの特殊な入力・出力デバイスしか持たないような端末の登場は、これまでの Web のクライアント像を一変させつつある。

CC/PP ( Composite Capability/Preference Profiles<sup>1),3)~5)</sup>および CC/PP Exchange<sup>2)</sup>は、こういっ

†1 慶應義塾大学政策・メディア研究科  
Graduate School of Media and Governance, Keio University

†2 東京大学情報基盤センター  
Information Technology Center, the University of Tokyo

†3 慶應義塾大学環境情報学部  
Faculty of Environmental Information, Keio University

†4 World Wide Web Consortium ( W3C )

この研究は、情報処理振興事業協会 ( IPA ) が実施している次世代デジタル応用基盤技術開発事業「次世代 WEB 基盤技術プロジェクト」のもとに行われた。

た要求に対応するために W3C および IETF で標準化を進めているコンテンツネゴシエーションのための仕様である。CC/PP ( および CC/PP Exchange ) は、クライアントの属性情報 ( 描画能力や処理可能なフォーマットなど ) を XML ( Extensible Markup Language ) ベースの記述形式である RDF ( Resource Description Framework )<sup>3)</sup> によって記述し、それを交換することで Web サーバやプロキシにクライアントに適した形でコンテンツを提供するための枠組みを提供するというものである。このように、サーバとクライアントの間で合意をとってからコンテンツ提供を行う機構のことをコンテンツネゴシエーションと呼ぶ。

CC/PP は、HTTP extension<sup>6)</sup>への準拠、XML の採用、複数ベンダ・開発者による属性情報提供のサポートなど、そのオープン性と拡張性の高さに大きな特徴を持つ。また、最近では WAP のプロファイル交換の仕様としても採用されている<sup>7)</sup>。しかし、HTTP 上での CC/PP の扱いについてはいまだ標準化の途中にあり、実装やそれに基づく評価は行われていない。また、XML の冗長性によるオーバーヘッドを少なくするために、キャッシュやプロファイル情報のバイナリ圧縮<sup>9)</sup>など様々な提案がされているが、その必要性や効果は不明なままである。我々は、CC/PP の実用性とその効果、影響を検証するために、CC/PP の仕様に基づく Web ブラウザおよびプロキシを開発した。

### 1.1 CC/PP によるコンテンツ変換

我々の開発したシステムは、多様なクライアントに対して Web コンテンツを適応させるために、中間プロキシにおいてコンテンツ変換を行うという構成をとる。このようなコンテンツ変換プロキシについては、CC/PP の出現以前から多くの試みがなされてきた。これらのうち多くは、コンテンツ変換のために特定のクライアント端末やネットワークを仮定しており、幅広いクライアントへの対応や相互運用性という点において拡張性に欠けている<sup>13)~15)</sup>。一方、特定の端末やネットワークに依存しないコンテンツ変換プロキシは、任意のクライアント環境において使用するためにユーザがあらかじめ細かい属性の設定を行う必要があったり<sup>11)</sup>、あるいは画像の圧縮など決まったタイプの変換しか得られなかったものが主であった<sup>12)</sup>。

このような既存のプロキシに対して、我々の CC/PP ベースの変換プロキシは以下のような利点を持つ。

- オープンな仕様や標準技術に基づくことで、特定の端末やネットワークに限定されず、現在のインターネットに新しいコンテンツ変換プロキシやクライアントを採り込んでいくことができる。
- ベンダなどの提供する複数のプロファイルを参照できるという CC/PP の特徴から、ユーザが細かい設定を行わなくても任意のクライアントの環境に応じてコンテンツ変換を提供できる。

### 1.2 本論文の構成

以降の章では、2 章でまず CC/PP の仕様の概要について説明し、その後それに基づいて開発した我々のブラウザとプロキシの実装について述べる。3 章では CC/PP の性能測定を行うための実験環境について説明し、4 章でその上で行った CC/PP の基本性能の測定結果と評価を述べる。最後に、5 章で本論文の結論を述べる。

## 2. プロトタイプ的设计と実装

### 2.1 CC/PP の概要

我々のプロトタイプ実装について述べる前に、CC/PP の仕様を概説する。なお、より詳しい仕様については、CC/PP および CC/PP Exchange の仕様<sup>1)~5)</sup>を参照されたい。

CC/PP の提唱するフレームワークは、CC/PP クライアント、CC/PP レポジトリ、CC/PP プロキシ、CC/PP サーバの 4 種類のノードから構成される。これらのノードは、コンテンツネゴシエーションのために CC/PP プロファイルと呼ばれる XML で記述された属性情報 ( ハードウェア・ソフトウェア属性およびユーザ嗜好 ) を交換する。

以下に各ノードの主な役割を説明する。

CC/PP クライアント： 端末の属性情報を CC/PP プロファイルとして持つ Web クライアント。サーバあるいはプロキシに HTTP リクエストを送信する際に、CC/PP プロファイルを付加して送る。また、途中で属性情報の変更があった場合 ( スピーカが On から Off になるなど )、*profile-diff* と呼ばれる元のプロファイルに対する差分データをリクエストに付加して送る。

CC/PP レポジトリ： 各ベンダの Web サイトなど、ハードウェアやソフトウェアの CC/PP プロファイルを特定の URI と関連づけて公開しているノード。クライアントが接続されているネットワークの帯域が狭い場合、クライアントは CC/PP プロファイルを直接送る代わりにレポジトリに対する URI だけを送ることができる。このようなプロ

ファイルの指定方法を間接参照と呼ぶ。また、プロファイル本体を直接リクエストに付加して送る方法を直接参照と呼ぶ。

**CC/PP プロキシ：** クライアントとサーバの間にある中間ノード。コンテンツを変換・加工する能力を持つ場合、どのような振舞いをしうかについての記述 (*proxy-behaviour* と呼ばれる) を CC/PP プロファイルに追加してサーバに転送する。また、もし宛先サーバが CC/PP に対応していない場合、CC/PP プロファイルを理解・解釈して HTTP/1.1<sup>10)</sup> の Accept ヘッダに変換するなど、既存のサーバが理解できるような形にプロファイルを変換する方法が提案されている。

**CC/PP サーバ：** CC/PP プロファイルを理解・解釈し、リクエストに対するコンテンツが複数バージョン存在する場合はその中から要求されたプロファイルに最も適合するコンテンツを選択し、あるいは生成、変換して返す Web サーバ。

上記の CC/PP プロキシの項で述べているプロファイルから Accept ヘッダへの変換の例としては、次のような場合が考えられる。たとえば、CC/PP プロファイルからクライアント端末がテキスト文書と GIF 画像しか扱えないことが分かったとすると、中間の CC/PP プロキシは “Accept: text/\*, image/gif” という HTTP ヘッダをリクエストに追加して Web サーバに転送する。Accept ヘッダのついたリクエストを受け取った Web サーバは、HTTP/1.1 の仕様によりそれに従った (Acceptable な) コンテンツを返さなければならない。このようにして、Web サーバが CC/PP に対応していなくても、Accept ヘッダを仲介とすることで簡単なコンテンツネゴシエーションを実現できる。

## 2.2 プロトタイプ的设计

CC/PP の構成例として、我々の実装では、CC/PP 対応のブラウザと、CC/PP プロファイルを解釈してコンテンツを変換するプロキシから成るシステムを構築した。本システムでは、サーバは CC/PP サーバではなく通常の Web サーバであることを仮定している。システムの全体構成を図 1 に示す。CC/PP サーバを用いず、また Accept ヘッダへの変換ではなくコンテンツを変換するプロキシを置くような構成をとったのは、以下のような理由による。

- CC/PP サーバにおけるコンテンツ選択の方法についてはまだ仕様が決まっていないため、CC/PP サーバを用いた実験が困難であった。
- 実験用のサーバを用いた場合、想定範囲内のコンテンツのみを扱った実験しか行えないため、実際

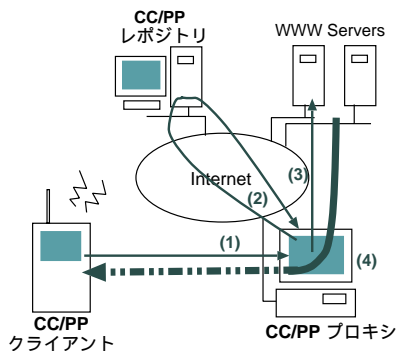


図1 システムの全体構成

Fig.1 Overall structure of the system.

に使われているサーバおよびコンテンツを用いて実験を行えるような環境を構築したかった。

- Accept ヘッダで指定できるクライアント属性は限られており、画面の表示能力やスピーカの有無などは Accept ヘッダに変換できない。このため、CC/PP プロファイルを直接解釈してコンテンツに反映させる構成をとった。

我々のシステムにおける処理の流れを以下に述べる。

- (1) CC/PP クライアントが HTTP リクエストに CC/PP プロファイルを付加して Web サーバに送る (図 1-(1))。
- (2) CC/PP プロキシがクライアントからの HTTP リクエストを横取りし、CC/PP プロファイルを解釈する。外部の CC/PP プロファイルへの間接参照があれば、CC/PP レポジトリからそのプロファイルを取得する (図 1-(2))。
- (3) CC/PP プロキシは宛先の Web サーバへ HTTP リクエストを転送する (図 1-(3))。
- (4) CC/PP プロキシはサーバから返された Web コンテンツを CC/PP プロファイルに基づいてクライアントに適した形に変換し、変換済の Web コンテンツをクライアントへ送り返す (図 1-(4))。

以下、プロキシとブラウザの実装と設計について述べる。

## 2.3 CC/PP プロキシの設計と実装

本プロキシは Perl で書かれており、CC/PP プロキシの基本的な機能とコンテンツ変換機能を実装している。また、実験的に簡易 Web サーバとして動くサーバモードを実装している。サーバモードで起動した場合、本プロキシはコンテンツ変換を行う CC/PP サー

ファイル形式、キャラクタセット、言語、エンコーディング、範囲指定に使える単位の 5 種類

表1 プロキシのコンテンツ変換機能

Table 1 Content conversion functions of our proxy.

変換機能	説明
画像形式変換	GIF, PNG, JPEG 間の相互変換
画像減色変換	表示可能色深度に応じた減色
画像サイズ変換	画面サイズに応じたサイズ変換
HTML 形式変換	HTML バージョン, XHTML 変換
文字コード変換	JIS, SJIS, EUC 間の相互変換
タグ抽出	指定された HTML タグのみを抽出
サマリ抽出	指定された見出しレベル以上を抽出

バとして動作する。コンテンツ変換機能としては、将来的には XSLT による変換など、より汎用的に進んだ変換機構を組み込むことを想定しているが、現在のプロトタイプ仕様では実験的に表 1 に示すような変換を実現している。

なお、我々のシステムでは、既存のサーバを想定して設計したため、プロキシの振舞いを CC/PP サーバに伝えるための記述である *proxy-behaviour* は現在サポートしていない。

#### 2.4 CC/PP ブラウザの設計と実装

CC/PP ブラウザは、標準的な Web ブラウザ機能に加えて、簡易端末エミュレータ機能と CC/PP プロファイル生成・送信機能を持つ。ブラウザはデフォルトプロファイルと呼ばれる端末の基本属性を記述した CC/PP プロファイルを持っている。この基本属性に対して、ユーザは起動オプションや GUI を通じていくつかのパラメータ（画面表示能力、画像品質など）を変更することが可能である。パラメータが変更されると、ブラウザはその変更に応じてデフォルトプロファイルに対する差分データ（*profile-diff*）を生成し、プロファイルの URI とともにリクエストに付加して送る。

### 3. 実験環境

#### 3.1 実験ネットワーク環境

我々は、ネットワーク接続状況の違いによるプロファイル転送のオーバーヘッドを比較するために、図 2 に示すような実験ネットワークを構築した。なお、本論文ではネットワークにおける効果と影響を計測することを主目標とするため、通常の PC（パーソナルコンピュータ）をプラットフォームとして採用した。実験ネットワークでは、CC/PP レポジトリがインターネット上の任意のホストに置かれているような状況を想定して、数百台のマシンとトラフィックを共有するキャンパスネットワーク上にレポジトリを設置した。レポジトリとプロキシの間には 4 ホップの隔りがある。一方、プロキシ-Web サーバ間のネットワークは、プ

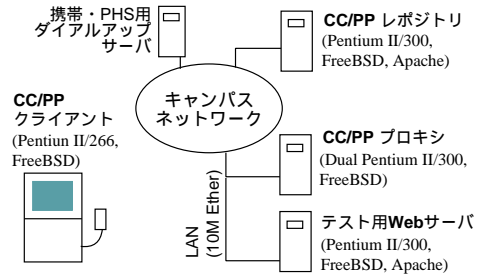


図2 実験ネットワーク環境

Fig. 2 Test network environment.

表2 クライアント-プロキシ間の接続

Table 2 Connection between client and proxy.

コネクション	帯域	RTT
無線 LAN (IEEE802.11b,Std)	2 Mbps	4~10 ms
無線 LAN (IEEE802.11b,Low)	1 Mbps	4~10 ms
PHS/PIAFS	64 kbps	220~260 ms
携帯電話/PDC	9600 bps	500~600 ms

表3 実験ドキュメントの概要

Table 3 Summary of sample pages.

画像数	2~62
取得に必要なリクエスト数	3~65
画像合計サイズ (kBytes)	1.54~134.4
合計サイズ (kBytes)	2.18~231.9
サーバまでのホップ数	10~30
サーバまでの RTT (msec)	6~210
各ドキュメント中のデータの割合	
HTML テキスト (text/html)	14.4~80.5%
GIF 画像 (image/gif)	19.5~85.6%
JPEG 画像 (image/jpeg)	0.0~50.6%
その他	0.0~7.2%

ロファイル転送の計測においてはあまり重要でないため、同一ネットワーク内に設置した。このようなネットワークにおいて、クライアントの接続環境の変化と CC/PP の性能に与える影響の関係を調べるため、クライアント-プロキシ間の接続として表 2 に示すような 4 種類の接続環境を構築して測定を行った。

#### 3.2 実験用 HTML ドキュメント

CC/PP によるコンテンツ変換実験用のサンプルの HTML ドキュメントとして、国内のプロキシで公開されている Web ページの利用統計情報をもとに、アクセス頻度の高いものから個人ページなどを除いた 50 サイトの Web サーバのトップページを使用した。使用したページの概要を表 3 に示す。

<http://www.saitama-u.ac.jp/squid-stats/> (埼玉大学のプロキシ統計データ) の 1999 年 1 月~12 月の情報を参考にサーバを選別した。

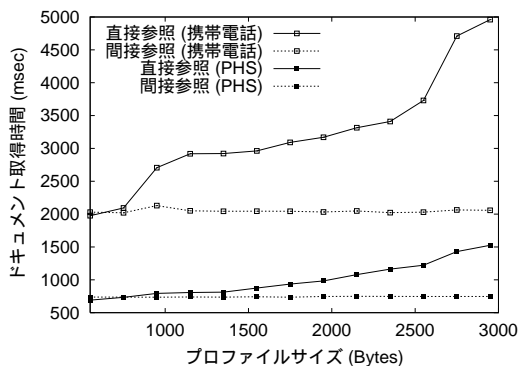


図3 間接参照と直接参照の比較(1)

Fig. 3 Comparison between indirect and direct CC/PP (1).

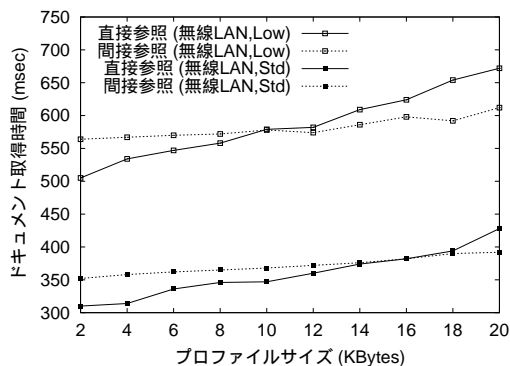


図4 間接参照と直接参照の比較(2)

Fig. 4 Comparison between indirect and direct CC/PP (2).

## 4. 性能測定と評価

### 4.1 CC/PPの基本性能

まず、CC/PPの基本性能を調べ、CC/PPの最適な使用方法を探るために、表2に示した環境においてプロファイル転送の計測および評価を行った。

#### 4.1.1 直接参照と間接参照

2.1節で述べたように、サーバあるいはプロキシにプロファイルを送付する方法として、リクエストにプロファイル本体を付加して送る直接参照と、URIだけを付加して送る間接参照の2種類が存在する。

直接参照を用いた場合、プロキシにおける処理時間は減るが、クライアント-プロキシ間の接続環境が悪い場合にはドキュメント取得時間に大きなオーバーヘッドをもたらす。一方、間接参照を用いた場合、ネットワーク全体のトラフィック総量は直接参照の場合よりも増加するが、クライアント-プロキシ間のトラフィックは大幅に削減される。このようなトレードオフのために、直接参照と間接参照のどちらを使用するかは、使用するプロファイルのサイズやネットワーク環境に応じて決定されるべきである。この境界を調べるために、我々はCC/PPを使用した状態でサイズ0のドキュメントを取得するのにかかった時間を直接参照および間接参照のそれぞれについて計測した。その結果を図3および図4に示す。なお、本計測にはプロキシのプロファイルキャッシュ機能は使用していない。

計測結果から、クライアント-プロキシ間の接続が携帯電話やPHSのような狭帯域のものである場合、プロファイルサイズが非常に小さい(1キロバイト以下)ケースを除いて間接参照の方が効率が良いことが分かった。一方、クライアント-プロキシ間の接続が1~2Mbps程度の広帯域である場合、プロファイルサ

イズが10キロバイト以上というように非常に大きくなり、直接参照の方が効率が良い。平均のプロファイルサイズが数キロバイト程度になると予想すると、クライアントの接続が携帯電話やPHSなど狭帯域である場合は間接参照を用いる方が望ましく、広帯域である場合には直接参照の方が望ましいといえる。

ただし、プロキシがプロファイルをキャッシュしていた場合は間接参照のメリットが増すため、この限りではない。キャッシュを使用した場合の計測結果については次項で示す。

#### 4.1.2 プロファイルキャッシュの効果

CC/PPプロファイルは、CC/PPプロキシやCC/PPサーバの各ノードにおいてキャッシュされる。間接参照とともにキャッシュを使うことにより、プロキシにおけるプロファイル取得時間のオーバーヘッドを削減することができる。一方、直接参照を用いた場合、HTTPにはセッションの概念がないため、HTTPプロトコルを拡張しない限りキャッシュをうまく活用する方法がない。このため、プロファイルキャッシュがもたらす効果は間接参照の場合に限られる。

我々は、間接参照におけるプロファイルキャッシュの効果調べるために、キャッシュを用いた場合とそうでない場合のページ取得時間を計測した。計測結果を図5に示す。図5はクライアントの接続が帯域2Mbpsの無線LANの場合の測定結果を示している。

図5から、キャッシュを使用した場合、クライアントの接続が広帯域でも、プロファイルサイズがある程度(実験では5キロバイトより)大きければ、間接参照を用いた方が直接参照よりも取得時間が短くなるこ

CC/PPの仕様に例示されているプロファイルが500バイト~2キロバイト程度、WAPの仕様に例示されているプロファイルの合計サイズが8キロバイト程度。

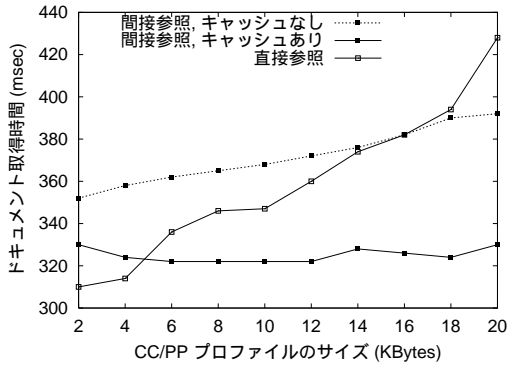


図5 キャッシュの効果  
Fig. 5 Effects of profile cache.

とが分かる．一方で，5キロバイトよりもプロファイルが小さい場合，やはり直接参照の方が効率が良い．このことから，キャッシュが使用可能な場合でも，クライアント-プロキシ間の接続が十分広帯域であり，プロファイルサイズが小さければ，直接参照を用いた方が効率が高くなる可能性があることが分かる．なお，2 Mbps よりも狭帯域なネットワークについては，ほぼすべてのケースにおいて間接参照(キャッシュあり)の場合の方が直接参照の場合よりも取得時間が短かった．

#### 4.1.3 差分転送のオーバーヘッド

間接参照の場合，属性を示すプロファイルを URI で指定するため，属性の一部に変更があると，URI で示されるプロファイルと実際の端末属性が一致なくなってしまう．このために，CC/PP ではプロファイルに対する変更分を *profile-diff* と呼ばれる差分データの形で URI に付加して送ることができる．差分データを用いることで，拡張機器を加えたり設定を変更したりした場合でもプロファイルを新規に作る必要がなく，間接参照を効率良く用いることができる．

しかし，差分データはプロファイルと同様に XML によって記述され，1つの差分データは 250~450 バイト程度と比較的大きい．したがって，プロファイルサイズが小さい場合，URI と差分データの合計サイズが変更後のプロファイル全体のサイズを超えてしまうことがあり，直接参照で変更後のプロファイルを送った方が効率が高くなる可能性がある．

図 6 に，間接参照において URI に差分データを付加して送った場合のドキュメント取得時間の変化を示す．なお，実験には 400~430 バイトの差分データを用いた．

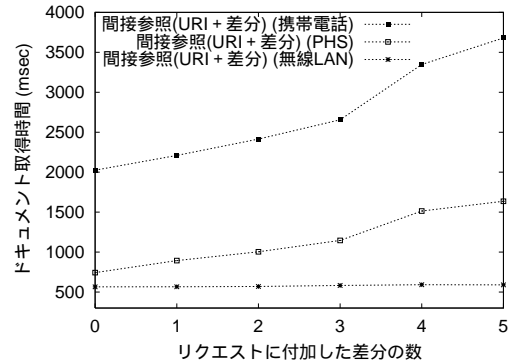


図6 差分転送のオーバーヘッド  
Fig. 6 Overhead caused by *Profile-Diff*.

図 6 と図 3 を比較すると，クライアントの接続が狭帯域の場合は差分データの数が 1, 2 個であればおおむね間接参照の方が効率が良い．しかし，それ以上差分データがある場合，変更後のプロファイルサイズが小さければ(たとえばクライアントの接続が携帯電話のときは変更後のプロファイルが 1 キロバイト以下の場合，また PHS のときは変更後のプロファイルが 2 キロバイト以下の場合)直接参照の方が効率が高くなる事が分かる．

#### 4.2 CC/PP を用いたコンテンツ変換

我々は，CC/PP によって生じるオーバーヘッドとコンテンツ変換の効果を比較するために，表 3 に示した 50 のサンプルページに対して CC/PP によるコンテンツ変換実験を行った．

実験には，約 220 バイトのソフトウェア属性プロファイルと約 330 バイトのハードウェア属性プロファイルから成る，合計約 550~570 バイトの CC/PP プロファイルを使用した．このプロファイルを用いて，それぞれのページに対して以下の 2 つの変換を組み合わせ合わせた合計 15 通りのページ取得実験を行った．なお，変換後の画像フォーマットは PNG とした．

- 画像減色変換：

color(無変換), gray256(8 bit グレイスケールに減色), gray16(4 bit グレイスケールに減色), gray4(2 bit グレイスケールに減色), mono(モノクロに減色)の 5 通り

- 画像サイズ変換：

full(無変換), 240x320(240x320 に縮小), 110x110(110x110 に縮小)の 3 通り

##### 4.2.1 CC/PP のオーバーヘッドと変換効果の比較

それぞれの変換の組合せを用いたときのクライアントにおけるトラフィックを平均し，総トラフィック量が多い組合せから順に並べ換えたものを図 7 に示

直接参照の場合，もともとリクエストごとにプロファイル全体を送るため，差分転送を用いるメリットはない．

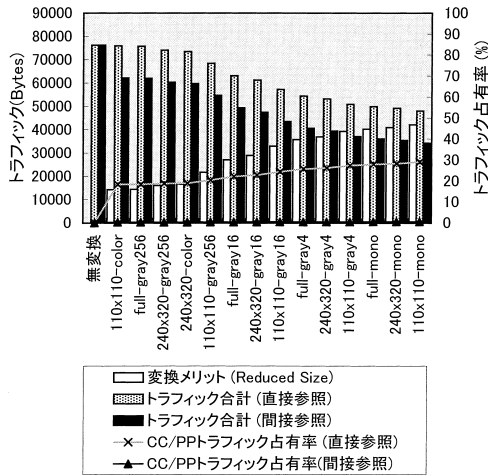


図7 CC/PP オーバヘッドと変換効果の比較  
Fig.7 CC/PP overhead and conversion effects.

す。図中の「変換メリット」とは、元のドキュメントサイズから変換後のドキュメントサイズを引いた値を示す。この値が大きいほど、変換によって削減されるトラフィック量が大きいことを意味する。また、横軸は適用した変換の組合せを示している。たとえば、gray256-110x110は、画像を8bitグレイスケールに減色し、画像サイズを110x110ドットに縮小した場合の測定結果を示す。

図7から、CC/PP オーバヘッドのトラフィック占有率は直接参照時で25~30%、間接参照時は0.2~0.7%程度であり、特に間接参照時のオーバヘッドは総トラフィック量から見ると非常に少ない。また、どちらの参照方法でも、すべての組合せの変換においてCC/PPを使用しない場合よりも合計トラフィックは少なかった。つまり、平均的なケースにおいて、CC/PPによるオーバヘッドよりもコンテンツ変換によって得られる効果の方が大きいといえる。

4.2.2 トラフィックにおける影響

コンテンツ変換の効果がCC/PPによるトラフィック増加を上回る条件を探るために、ページに含まれる画像サイズの変化によるコンテンツ変換効果の変化をグラフにしたものを図8に示す。図8の横軸はページに含まれる画像の合計サイズであり、縦軸はそのページに対して得られた変換効果(CC/PPを使用しなかった場合の総トラフィック量から使用した場合の総トラフィック量を引いたもの)である。この値が正の値であるとき、CC/PPを用いた方がトラフィック量が少なかったことを示す。なお、ここでは、変換効果の差が明確な減色変換についてのみ比較を行った。また、間接参照の場合はトラフィック上のCC/PPオーバヘッ

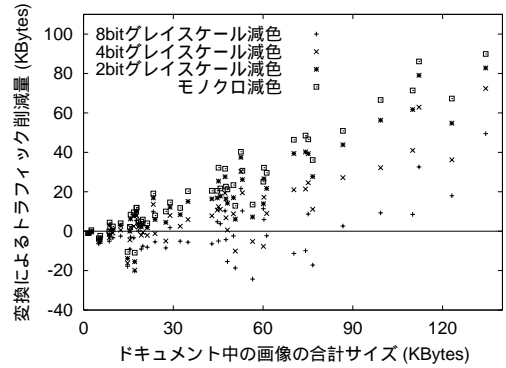


図8 減色変換によるトラフィック削減効果  
Fig.8 Conversion effects on network traffic.

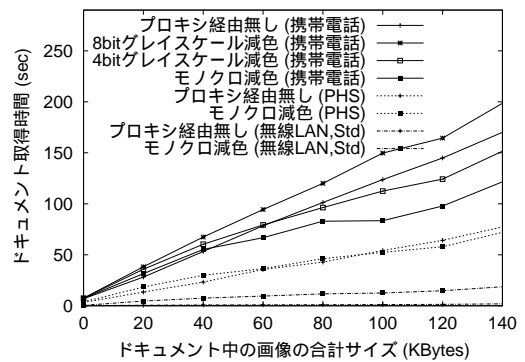


図9 減色変換によるドキュメント取得時間の変化  
Fig.9 Conversion effects on retrieval time.

ドがほとんどないため、直接参照を用いた場合の結果を示している。

図8から、ドキュメント中に含まれる画像の合計サイズが小さい場合はCC/PPを使用した方がトラフィックが増えるが、画像が多い場合は(実験では90キロバイト以上)変換効果の方が大きくなることが分かる。また、4bitグレイスケールよりも低品質への画像変換を行った場合、大半のドキュメントにおいて(4bitグレイスケールの場合66%,2bitグレイスケールの場合86%以上のドキュメント)CC/PPを用いた方がトラフィックが削減された。

以上から、4bitグレイスケールよりも低品質の画像への変換が必要な場合、CC/PPによるコンテンツ変換の効果は一般的にそのオーバヘッドを上回ることが分かる。また、画像が多いページを主に取得する場合、端末の表示能力にかかわらずCC/PPによるコンテンツ変換は有効であるといえる。

4.2.3 ドキュメント取得時間における影響

CC/PPのオーバヘッドと変換効果がドキュメント取得時間に与える影響を調べるために、様々なクライ

アント接続環境においてドキュメント中の画像サイズが変化した場合のドキュメント取得時間を測定した。測定結果を図9に示す。なお、本測定には図2に示したネットワーク環境を使用した。また、プロファイルの転送にはキャッシュなしの間接参照を用いた。

図9から、クライアントの接続環境が携帯電話程度であり、かつ4bitグレイスケールよりも低い画質への画像変換が行われた場合、CC/PPを用いた方がドキュメント取得時間を削減できる可能性が高いことが分かる。一方、PHS程度の接続環境である場合、モノクロ減色への変換を行った場合において、CC/PPを用いたケースと用いないケースのドキュメント取得時間がほぼ同程度となる。無線LANのような広帯域のネットワークでは、すべての場合でCC/PPを用いなかった方がドキュメント取得時間が短かった。

以上から、CC/PPのドキュメント取得時間におけるオーバーヘッドは、クライアントの接続環境が携帯電話程度の狭帯域のものである場合を除いて、比較的大きいことが分かる。しかし、本プロキシがまだプロトタイプ段階で性能調整を行っていないこと、また測定にキャッシュを使用していないことなどから、実際の運用におけるCC/PPのオーバーヘッドは本計測結果よりも大きく削減できる可能性が高い。また、クライアントの接続環境が携帯電話のように狭帯域のものである場合は、プロキシの性能にかかわらずCC/PPによるコンテンツ変換が有効な選択肢となりうる事が分かる。

## 5. 結 論

本論文では、CC/PPの仕様に基づいたブラウザとプロキシのプロトタイプを実装し、その性能測定を行うことで、CC/PPの性能と有効性を検証した。

CC/PPの基本性能の測定結果から、クライアントが携帯電話やPHS程度の狭帯域のネットワークに接続されている場合は、ドキュメント取得時間の削減において間接参照の使用が効果的であることが分かった。一方で、クライアントの接続が1~2Mbps程度の広帯域である場合、プロキシにおいてプロファイルがキャッシュされたとしても、使用するプロファイルのサイズが小さければ(2Mbpsで5キロバイト以下)直接参照の方が性能が良くなる可能性があることが分かった。また、元のプロファイルに対する差分データの数が一定以上増えた場合、間接参照に対して差分データを用いた場合に生じるオーバーヘッドが直接参照を用いた場合のオーバーヘッドよりも大きくなる可能性があることを示した。

さらに、CC/PPを用いたコンテンツ変換の有効性を調べたところ、クライアントが4bitグレイスケールよりも低品質な画像への変換を要する場合、コンテンツ変換によってCC/PPのオーバーヘッドを上回るトラフィック削減効果が見込めることを示した。また、クライアントの接続が携帯電話程度であれば、CC/PPを用いた方がドキュメント取得時間を削減できる可能性が高いことを示した。これらの実験結果から、CC/PPのコンテンツ変換への適用が有効であることを示した。

## 参 考 文 献

- 1) Reynolds, F., Hjelm, J., Dawkins, S. and Singhal, S.: Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation, <http://www.w3.org/TR/NOTE-CCPP>, W3C Note (1999).
- 2) Ohto, H. and Hjelm, J.: CC/PP exchange protocol based on HTTP Extension Framework, <http://www.w3.org/TR/NOTE-CCPPexchange>, W3C Note (1999).
- 3) Nilsson, M., Hjelm, J. and Ohto, H.: Composite Capabilities/Preference Profiles: Requirements and Architecture, <http://www.w3.org/TR/CCPP-ra/>, W3C Working Draft (2000).
- 4) Klein, G.: CC/PP Attribute Vocabularies, <http://www.w3.org/TR/CCPP-vocab/>, W3C Working Draft (2000).
- 5) Reynolds, F., Woodrow, C. and Ohto, H.: Composite Capabilities/Preference Profiles: Structure, <http://www.w3.org/TR/CCPP-struct/>, W3C Working Draft (2000).
- 6) Frystyk, H., Leach, J. and Lawrence, S.: An HTTP Extension Framework, RFC 2774 (2000).
- 7) Wireless Application Protocol Forum: *WAGUAPROF Version10—Wireless Application Group User Agent Profile Specification*, Wireless Application Protocol Forum (1999).
- 8) Brickley, D. and Guha, R.V.: Resource Description Framework (RDF) Schema Specification 1.0, <http://www.w3.org/TR/rdf-schema>, W3C Candidate Recommendation (2000).
- 9) Martin, B. and Jano, B.: WAP Binary XML Content Format, W3C Note (1999).
- 10) Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, J. and Berners-Lee, T.: Hypertext Transfer Protocol—HTTP/1.1, RFC 2616 (1999).
- 11) Spyglass, *Spyglass Prism 3.0*, Spyglass.,



<http://www.spyglass.com/>.

- 12) Intel, QuickWeb, <http://www.intel.com/>.
- 13) Fox, A., Goldberg, I., Gribble, S.D., Lee, D.C., Polito, A. and Brewer, E.A.: Experience With Top Gun Wingman: A Proxy-Based Graphical Web Browser for the 3Com PalmPilot, *Proc. Middleware 98*, The Lake District, Springer (1998).
- 14) Baret, R. and Maglio, P.: Intermediaries: New Places for Producing and Manipulating Web Content, *Proc. 7th International WWW Conference*, Brisbane, pp.509-518, Elsevier Science B.V. (1998).
- 15) PUMATECH, Browse-It Server, <http://www.proxinet.com/>.

(平成 12 年 7 月 27 日受付)

(平成 13 年 3 月 9 日採録)



安田 絹子 (学生会員)

1996 年慶應義塾大学環境情報学部卒業。1998 年同大学院政策・メディア研究科博士前期課程修了。現在同大学院政策・メディア研究科博士後期課程。モバイルコンピューティング, システムソフトウェア, World Wide Web 等に興味を持つ。電子情報通信学会学生会員。



浅田 卓哉

1993 年千葉大学工学部情報工学科卒業。1995 年同大学院工学研究科情報工学専攻修士課程修了。同年株式会社スリー・イー・システムズに入社。1998 年から 2000 年にかけて慶應義塾大学 SFC 研究所 W3C 研究員として World Wide Web 関連の企画策定・ソフトウェア開発などを行う。現在東京大学情報基盤センター助手。World Wide Web サーバの管理・運用および研究を行っている。



萩野 達也 (正会員)

1981 年京都大学理学部卒業。1983 年同大学院理学研究科数理解析専攻博士前期課程修了。1986 年英国エジンバラ大学理学部計算機科学科理論 Ph.D. コース修了。Ph.D. in Computer Science. 1987 年京都大学大型計算機センター助手。1990 年慶應義塾大学環境情報学部専任講師。1992 年同助教授。2001 年同教授, 現在に至る。プログラミング言語, 情報処理教育, システムソフトウェア, World Wide Web 等に興味を持つ。W3C Duputy Director for Asia を兼任。電子情報通信学会, ソフトウェア科学会, ACM, IEEE, USENIX 各会員。