

第3世代移動通信網を介した高速インターネットアクセスのためのステルス型ゲートウェイの提案と実装評価

高橋 修^{†1} 関口 克己^{†2} 鶴巻 宏治^{†3}
 稲村 浩^{†1} 渥美 幸雄^{†1} 水野 忠則^{†4}

本論文では、第3世代移動通信網である W-CDMA パケット網を経由してインターネットアクセスを高速に行うためのゲートウェイの実現方式の提案と、その実装による評価結果について述べる。最初に、高速化のためのプロトコル手法の概要と GW への要求条件について述べる。また、W-CDMA パケット網向けにチューンアップされた TCP である W-TCP と既存 TCP の相互変換、各種アドレス情報の End-End レベルでの透過性、および任意の AP プロトコルの中継を可能とするステルス型 GW を提案する。さらに、提案したステルス型 GW 方式の試作ソフトウェアによる実証評価結果について述べる。これらの結果、ステルス型 GW 方式により End-End のアドレス情報 (IP アドレス、TCP ポート番号) の透過性を維持したまま、スルーットおよび応答時間をそれぞれ 2~3 倍程度向上すること、および GW における中継処理時間が無視できるほど小さいことを明らかにし、提案方式が実用的なものであることを示す。

Implementation and Evaluation of Stealth Gateway System for 3G Mobile Network

OSAMU TAKAHASHI,^{†1} KATSUMI SEKIGUCHI,^{†2} KOJI TSURUMAKI,^{†3}
 HIROSHI INAMURA,^{†1} YUKIO ATSUMI^{†1} and TADANORI MIZUNO^{†4}

The third generation wireless communication system known as IMT-2000 with wide bandwidth of a maximum of 384 kbps, in turn, provides high-speed data transmission and access to the Internet. As a channel utilization characteristic of IMT-2000, propagation delay for a message and the corresponding reply (RTT: Round Trip Time) is considered to be much larger compared to the bandwidth. The higher the delay, the greater the stress placed on the transport protocol (TCP) to operate efficiently. There is a huge impact on TCP performance seen for conventional TCP with greater amounts of data held "in transit" in the network, hence, inefficiency. TCP profile of W-CDMA has been proposed to enable such TCP to more effectively utilize the available capacity of the network path. In this paper, we propose a stealth gateway method to convert TCP profile of W-CDMA to and from legacy TCP, and implement this method to gateway system. We performed simulation under the third generation mobile network and evaluated its efficiency.

1. はじめに

第3世代移動通信網 (IMT-2000) の商用サービスとして W-CDMA をベースとした FOMA 等の商用サービスがすでに始まっている。FOMA では、高速

パケット交換サービス (上り最大 64 kbps, 下り最大 384 kbps) が提供されており、従来に比較してより高速なインターネットアクセスが可能となっている。この高速移動パケット通信網の特性として、無線区間を含む網内伝送遅延時間が、伝送帯域幅に比較して大きい (すなわち帯域幅遅延積が大きい) ことが分かっている。このような特性を持つ移動通信網において、帯域幅遅延積が比較的小さい有線網を前提に設計された TCP を利用したアプリケーションが高速移動通信網の特徴を生かしきれず、十分な性能 (スルーット、応答時間) を得られないことが明らかになっている¹⁾。プロトコル最適化による性能改善の試みとして、

†1 株式会社 NTT ドコモマルチメディア研究所
 Multimedia Laboratories, NTT DoCoMo Inc.

†2 株式会社 NTT ドコモマルチメディア開発部
 Multimedia Development Division, NTT DoCoMo Inc.

†3 株式会社 NTT ドコモ東海
 NTT DoCoMo TOKAI Inc.

†4 静岡大学情報学部
 Faculty of Information, Shizuoka University

無線区間と有線区間の TCP コネクションを分離する Split TCP^{2),3)}や, PEP(Performance-Enhancing Proxy)^{4),5)}がある。しかし, これらは TCP コネクションをプロキシで終端させることが前提となっており, TCP コネクションは, クライアント-プロキシ間とプロキシ-サーバ間でそれぞれ独立に設定・管理され, 一方の End のアドレス情報(たとえば, IP アドレス, ポート番号, 等)が他方の End で認識できなくなる。すなわち, End-End でアドレス情報の透過性はなくなる。このような状況においては, インターネットで提供される各種サービスの中で利用できないものが発生する可能性がある。また, アプリケーションプロトコル(以下 AP プロトコルと略す)には種々のものがあり, 今後も新サービスの登場にともない, 新しいプロトコルが開発されることになる。このため, プロキシでは任意の AP プロトコルを使える仕組みを用意する必要があるが, 現状のプロキシは AP プロトコルを限定して実装しており, 汎用性がない。

一方, End-End に TCP を適用することを前提とした TCP のプロファイルとして W-TCP(TCP over 2.5G and 3G Wireless Networks)が提案されている^{1),10)}。この場合, たとえば W-CDMA パケット網に接続するためのパソコン等のドライバソフトウェアの配布とあわせて TCP プロファイル設定ソフトウェアを配布すること等によって, クライアントを比較的容易に W-TCP に変更可能である。しかし, インターネット上に無数に存在するホストのすべてにこのプロファイルセットを適用することは非現実的である。

上記 2 つの問題を解決するためには, クライアントとインターネット上のサーバ間で W-TCP と TCP を変換し中継するゲートウェイ(以下, GW と略す)が必要となる。GW では, W-TCP と TCP の変換を行いつつ, End-End レベルでの IP パケットの透過性と任意の AP プロトコルを使える仕組みが必要であるが, これらを実現したものはまだない。

本論文では, 最初に W-CDMA パケット網等の高速移動通信網を経由してインターネットアクセスを高速に行うためのプロトコル手法の概要について述べる。次に, 上述した課題を解決するため, IP アドレス等の透過性を保ちつつ, W-TCP と TCP の相互変換と HTTP(Hypertext Transfer Protocol) リクエストパイプライン⁶⁾を含む任意の AP プロトコルの中継を可能とするステルス型 GW を提案する。さらに, 提案したステルス型 GW 方式の試作による実証評価結果について述べる。これらの結果, ステルス型 GW 方式により End-End のアドレス情報の透過性を維持し

たまま, スループットおよび応答時間をそれぞれ 2~3 倍程度向上すること, および GW における中継処理時間が無視できるほど小さいことを明らかにし, 提案方式が実用的なものであることを示す。

2. インターネットアクセスの高速化のためのプロトコル手法の概要

FOMA パケット交換サービスでは, 高速パケット交換サービス(上り最大 64 kbps, 下り最大 384 kbps)が提供されており, またそのパケット遅延時間(RTT: Round Trip Time)は実測で平均 600 msec 程度であり, 電波の状態によって大きく変動する。このような環境で Web アクセスを含む各種インターネットアプリケーションの高速化を行うために必要なプロトコル(IETF 等で標準化されているものを前提としている)のうち, 基本的なプロトコルである TCP と HTTP の検討状況について概説する。

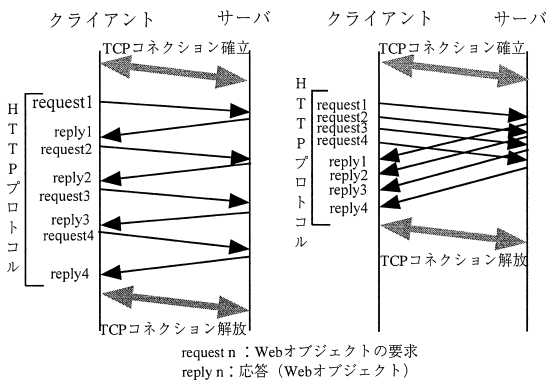
(1) W-TCP

W-CDMA パケット網の特徴は高速・高遅延であり, これを有線ネットワークを前提に開発された TCP にそのまま適用すると回線の使用率(スループット)が低下することが課題となる。このような特性を有する無線ネットワークにおける TCP の最適化のため様々な解決方法が提案されている^{8),9)}。これらの手法は, スロースタート期間の短縮のために輻輳ウィンドウの制御方法の変更を行う等, いずれも現在流通している TCP の実装の大幅な変更が必要となったり, 既存 TCP との相互接続性が損なわれたりする可能性がある。一方, 現状の各種 TCP の実装に適用可能であり, かつ, 既存の TCP との相互接続性を考慮した最適化手法をとりまとめたのが W-TCP であり, 現在 IETF で標準化が進められている¹⁰⁾。

W-CDMA パケット網では, 無線区間で発生するパケットロスに対処するためデータリンクレベルで再送制御が行われ, TCP から見ると, あたかも RTT が増大したかのように見える。W-TCP で規定されている各種手法の中から, このような特性を持つ W-CDMA パケット網に TCP を適用するときに性能向上に特に有効と想定されるものを選択して利用することが提案されている¹¹⁾。具体的には, 以下の TCP 関連パラメータを W-CDMA パケット網の特性に合わせてチューニングする。

1) TCP バッファサイズの拡大

帯域幅遅延積に対して十分な広告受信ウィンドウサイズを確保するため TCP 受信ウィンドウサイズを 64KB まで拡大する¹²⁾。



(a) 逐次的に処理 (従来方式) (b) バイプライン処理

図1 HTTP/1.1パイプライン処理の概要

Fig.1 Outline of HTTP/1.1 pipelining.

2) 初期ウィンドウサイズの拡大

TCP コネクション確立後の転送初期のスループット向上のため送出量の拡大を図るために初期ウィンドウサイズを3セグメントまで拡大する¹³⁾。

3) 最大転送パケット長の拡大

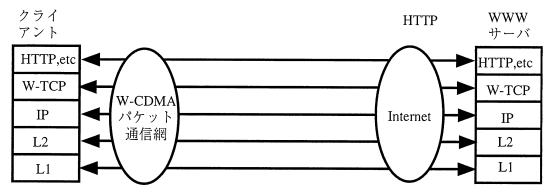
輻輳ウィンドウは最大 TCP セグメント長 (MSS: Maximum Segment Size) を単位に管理されているが、スロースタート期間の短縮のためには MSS を長くすることが有効である。MSS はリンクごとの最大転送パケット長 (MTU: Maximum Transfer Unit) を基に決まるため、MTU は運用実績のある値で最大の 1.5 KByte とする。

4) 選択的確認応答

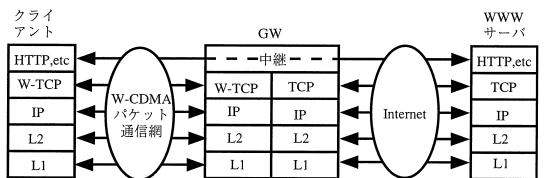
ネットワークの輻輳等により、パケットロスが発生したときの再送を効率良く行うために、選択的再送機構 (SACK: Selective ACKnowledgement)¹⁴⁾を使用する。

(2) HTTP バイプラインング

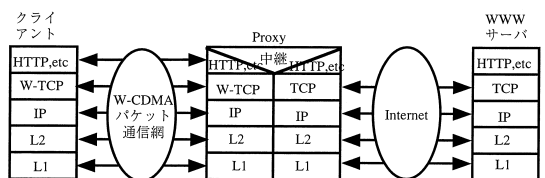
HTTP/1.1 では、パーシステントコネクションの利用が可能となっている。パーシステントコネクションでは、1つの TCP コネクションの中で、Web オブジェクトの取得 (HTTP の Request-Response) を順次連続して行うことが可能となる。リクエストパイプラインングはパーシステントコネクションの利用を前提とし、1つの TCP コネクションの中で、複数の Request を Response の受信を待たずに一括して連続的に送信することができる (図1)。リクエストパイプラインングは、特に RTT の大きいネットワークで用いた場合にその効果が大きいと実測評価されている⁷⁾。



a) End-Endモデル



b) GWモデル (提案方式)



c) (参考) Proxy方式モデル

HTTP: Hyper Text Transfer Protocol
W-TCP: TCP over 2.5G and 3G Wireless Networks

図2 基本アーキテクチャモデル

Fig.2 Architecture of communication model.

表1 基本方式の特徴

Table 1 Features of base models.

	APプロトコルの透過性	アドレス情報の透過性	W-TCPの適用
End-Endモデル	○	○	× (すべてのサーバへの適用は困難)
GWモデル	○	○	○
(参考) Proxyモデル	× (APプロトコルがスプリットされる)	× (TCPがスプリットされる)	○

3. ステルス型 GW 実装方式の提案

3.1 基本アーキテクチャと GW への要求条件

(1) 基本アーキテクチャ

2章で述べたプロトコルを実装する基本アーキテクチャとして、End-End でこれらを実装する方式と両 End の間に GW を設置する方式がある (図2, 表1)。W-TCP は End-End の通信モデルを前提とし、IETF で標準化された拡張技術のみを利用しているが、現在市場に出荷されている各種 OS の TCP の実装は初期値で W-TCP のプロファイル (2章で示した TCP パラメータ) が有効でない OS も多数存在することが分かっており、OS の TCP パラメータの設定変更が必要になる。クライアント側は、W-CDMA パケット

網へ接続するために必要となるドライバソフトウェアのインストールに合わせて、W-TCP への設定変更が比較的容易に実現可能である。しかし、インターネットに無数に存在するサーバに対して W-TCP を適用するには、OS のパラメータ変更や最新版への改版を待たねばならず、現実的にはすべてのサーバ側 OS の設定変更は不可能である。以上の理由により、本論文では GW 方式を採用することとした。

(2) GW への要求条件

GW で実現すべき機能の要求条件を整理する。

1) W-TCP と TCP の変換

クライアント側は W-TCP に設定されているという前提で、GW で W-TCP と TCP を変換する。

2) 各種アドレスの透過変換処理

GW でプロトコル変換処理を実施すると、IP アドレスや TCP ポート番号等のアドレス情報も同時に変換される場合があり、たとえば、サーバからは、クライアントの IP アドレスが分からないことになる。このことが、サービスによっては大きな制約になる可能性がある。このため、クライアント側およびサーバ側から見て、互いに直接通信しているように見なせることが重要であり、GW では、アドレス情報の透過性を保証する必要がある。

3) AP プロトコルの透過性

HTTP パイプラインングをはじめとして、任意の AP プロトコルの中継を可能とするためには、TCP で運ばれる情報をそのまま中継できる必要がある。

3.2 ステルス型 GW 実装方式

移動通信網や衛星回線等の RTT の大きなネットワークを経由したインターネットアクセスを効率良く行うための GW 実装の研究例^{15)~17)}はいくつかあるが、前節で述べた要求条件を満たす GW の実装例はまだない。このような要求条件を実現する GW は、IP アドレス等のアドレスの透過性を実現し、両 End のホストからは認識する必要がないため、本論文では「ステルス型 GW」と呼ぶ。ステルス型 GW の特徴は、クライアントおよびサーバからはあたかも End-End で通信しているように見なせ、かつスループットおよび応答時間の向上が図れることである。このため、サーバおよびクライアント側では、GW に関する情報をいっさい事前に設定する必要がない利点がある。

ステルス型 GW では、W-TCP と TCP との変換処理、および AP プロトコルのトランスペアレントな中継を実現する。さらに、IP アドレス/TCP ポート番号変換処理によりアドレス情報の透過性を実現する。これらの変換中継機能を GW で実装する方式として

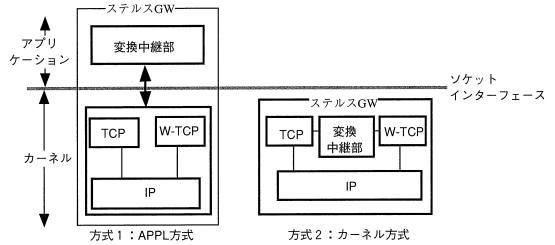


図3 GW 変換中断部の実装方式
Fig. 3 Implementation method of GW.

表2 GW の変換中断部の実装方式の比較

Table 2 Features of GW implementation method.

	性能	保守性・運用性	移植性(流通性)	開発規模・期間
方式1	× (オーバーヘッド大)	○	○	○
方式2	○	× (カーネルの知識が必要)	× (カーネル種別毎に必要)	△ (カーネル構造に依存)

以下の方式がある(図3, 表2)。

方式1: GW でアプリケーション (APPL) として実現する

方式2: OS (カーネル) で実現する

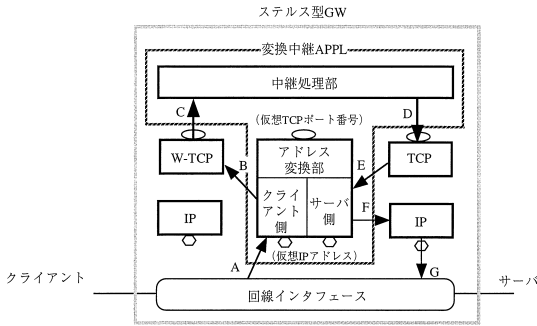
方式1は、方式2と比較して、アプリケーションであるためソケットインタフェースを経由した制御および情報の受け渡しがあるためオーバーヘッドが大きい。保守・運用性、移植性に優れ実用的であるため、これらを重視し方式1を採用することとした。

以下では、方式1を前提にステルス型GWの特徴的な機能であるアドレス透過変換方式と、TCPコネクション確立方式について述べる。

3.3 アドレス透過変換方式

提案するステルス型GWではEnd-EndレベルでのIPアドレスおよびTCPポート番号の透過性を実現するために以下の透過変換処理方式を実現する。透過変換処理はIPパケットの透過処理を行うアドレス変換部、TCPコネクション管理・TCPセグメントの中継処理部(すなわち、W-TCPとTCPの相互変換のための中継処理)、プロトコル処理部(TCP, W-TCP, IP)からなる(図4)。

アドレス変換部は、一方の回線経由で受信したIPパケットのIPアドレス、およびTCPポート番号を記憶するとともに、GW内部での中継処理のため、それぞれのGW内部処理用アドレス情報に変換し、中継処理部に渡す。中継処理部で中継されたIPパケットは、再びアドレス変換部に渡され、記憶されたアドレス情報を基に、GW内部処理用アドレスからオリジナルアドレス情報に再び変換され、他方の回線経由で



備考1: A-HはIPパケットをGWで受信してから透過変換処理を行い送信するまでの処理の流れ
 2: 各機能部が持つアドレス情報
 ○はTCPポート番号, ⊙はIPアドレスを示す
 アドレス変換部は、1つの仮想TCPポート番号と、2つの仮想IPアドレスを持つ

図4 透過型アドレス変換方式の構成

Fig. 4 Addresses transparent conversion method.

表3 アドレス変換で使用する各種情報

Table 3 Summary of addresses information using conversion.

TCPコネクション 対識別子	IPアドレス	TCPポート番号
アドレス変換部 仮想TCPポート番号 (任意)	クライアント (任意)	クライアント (任意)
	サーバ (任意)	サーバ (AP固定)
	GWクライアント側 (固定)	GWクライアント側 (固定)
	GWサーバ側 (固定)	GWサーバ側 (任意)
	アドレス変換部クライアント側 (固定) アドレス変換部サーバ側 (固定)	アドレス変換部 仮想TCPポート番号 (任意)

備考: 網掛部がステルス型GWの中継処理部、アドレス変換部で新たに定義したアドレス情報

送出される。

これらのアドレス変換処理は、TCP コネクション対ごとに行う必要がある。このため、TCP コネクション対を識別するためにアドレス変換部に付与する仮想TCPポート番号を使用する。また、受信したIPパケットがクライアント側あるいはサーバ側なのかを識別するためにアドレス変換部にクライアント側/サーバ側に対応する2つの仮想的なIPアドレスを付与する。アドレス変換に使用される各種アドレス情報をまとめて表3に示す。

IPパケットの透過変換処理の具体的な手順をクライアント側ホストからIPパケットを受信し、サーバ側ホストに送り出すまでの例を以下に示す(図4参照)。

手順1: アドレス変換部(クライアント側)でIPパケット受信(図4のA)

IPアドレス変換: 送信元IPアドレス→仮想IPアドレス(クライアント側)

宛先IPアドレス→GWのIPアドレス(クライアント側)

TCPポート番号変換: 送信元TCPポート番号→仮想TCPポート番号(TCPコネクション確立時は新規に取得する。以降、これをキーに各種アドレス情報を参照する)

宛先TCPポート番号→中継処理部のTCPポート番号(クライアント側)

手順2: 変換後のIPパケットをTCP/IP処理部へ送信(図4のB)

手順3: 中継処理部で中継(図4のC)

宛先IPアドレス: 仮想IPアドレス(サーバ側)

宛先TCPポート番号: 仮想TCPポート番号

手順4: TCP処理部へ送信(図4のD)

手順5: アドレス変換部(サーバ側)でIPパケット受信, 変換テーブルを参照し以下の処理を実行(図4のE)

IPアドレス変換: 宛先IPアドレス→サーバホストIPアドレス

送信元IPアドレス→クライアントホストIPアドレス

TCPポート変換: 宛先ポート番号→サーバホストポート番号

送信元ポート番号→クライアントホストポート番号

手順6: IP処理部へ送信(図4のF, G)

備考: A→B: A欄をB欄で置き換えることを意味する

また、アドレス変換部では、アドレス変換した後に必要に応じてIPヘッダとTCPヘッダのチェックサムの再計算を行う。

3.4 TCPコネクション確立方式

TCPコネクションは、クライアント側とサーバ側で同期して確立する必要があるが、クライアントとGW間のTCPコネクションが先に確立される従来方式では、サーバの障害等により、サーバ側TCPコネクション確立が失敗した場合においても、クライアントからHTTPリクエストが送出され、無効なパケット送出が発生する可能性がある(図5-a)。この現象を回避するため、クライアントからのTCPコネクション確立要求(SYN)パケット受信後、クライアントへ応答(SYN+ACK)パケットの送信を保留し、サーバ側のTCPコネクションの確立を先に行う方式を提案する。

提案方式では、クライアントへSYN+ACKの送信を保留している間、クライアント側のTCPコネクション確立処理は待機状態とし、サーバ側のTCPコネクション確立処理が完了した時点で、クライアント側のTCPコネクション確立処理を再開する(図5-b)。な

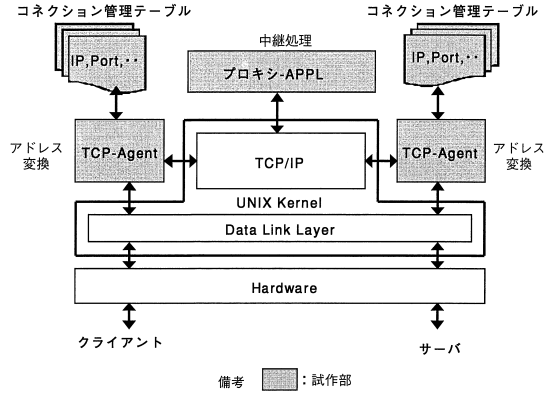
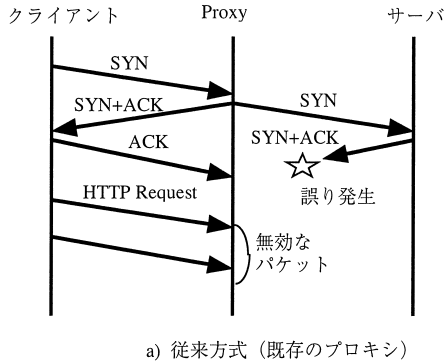


図 6 ステルス型 GW 試作ソフトウェア構成
Fig.6 Prototype software stealth GW.

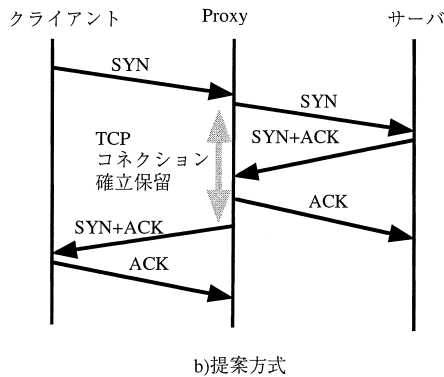


図 5 TCP コネクション確立方式
Fig.5 TCP connection establishment method.

お、サーバ側の TCP コネクション確立失敗時は、クライアントへ応答を破棄することによって TCP コネクションの確立を行わない。この場合、クライアントから無用な SYN 再送の発生可能性があるが、ステルス型 GW では、End-End の一方で発生している事象を可能な限り忠実に他方に伝えることを基本としているため、受信した SYN はすべてサーバ側に送信する。また、サーバ側の TCP コネクション確立中に、クライアントの TCP コネクション監視タイムがタイムアウトし、クライアント側から SYN を重複受信する可能性がある。この場合、重複受信した SYN は廃棄し、最初の SYN 受信時に行ったサーバ側 TCP コネクション要求に対する応答を待つ。

具体的には以下の手順で TCP コネクションの確立を行う。

- a) クライアントからの SYN 受信後、タイム(以降、Connect タイムと呼ぶ)をスタートし、サーバ側の TCP コネクション確立を行う。
- b) クライアントから再度 SYN を受信した場合、受信した SYN を廃棄する。

- c) サーバ側 TCP コネクション確立が失敗した場合、Connect タイムをストップし、処理を終了し、関連する情報をリセットする。
- d) サーバ側 TCP コネクション確立が成功した場合、Connect タイムをストップし、クライアント側の SYN 受信以降の TCP コネクション確立処理を行う。
- e) Connect タイムがタイムアウトした場合は、処理を終了し、関連するテーブル情報をリセットする。この手順により、サーバ側の TCP コネクションは、クライアント側の TCP コネクションよりも先に確立されることになり、宛先不明のサーバへの TCP パケットがクライアントから送られることを避けることができる。

なお、Connect タイムは、サーバ側の TCP コネクション確立要求がリトライアウトするまでの最大時間を考慮し、十分大きな値とする。

4. 実装評価

4.1 ステルス型 GW の試作

試作したソフトウェアの構成を図 6 に示す。試作したソフトウェアは、アドレス変換処理を行う TCP-Agent と、TCP セグメントの中継の中継処理を行うプロキシ-APPL (Application) の 2 つのプロセスと、IP アドレス変換前後の IP アドレス等と TCP コネクションの状態を保持するコネクション管理テーブルからなる。

(1) TCP-Agent

TCP-Agent は、GW 装置のネットワークインタフェースに入ってきたすべての IP パケットのキャプチャリングを行い、取り込み条件に一致した IP パケットのアドレス変換処理を行い、ソケットへ送信する。

取り込み条件に一致しなかった IP パケットは破棄する．具体的な取り込み条件は，

- a) プロトコル番号が 6 (TCP)
- b) 宛先 IP アドレスが GW 装置のネットワークインタフェースの IP アドレス以外
- c) SYN であるかまたは，SYN 以外であるが，送信元および宛先 IP アドレス対がすでにコネクション管理テーブルに登録されている場合

である．

SYN 受信時には，IP ヘッダの送信元 IP アドレス，宛先 IP アドレス，TCP ヘッダの送信元 TCP ポート番号，宛先 TCP ポート番号の変換処理を行い，これらの情報をコネクション管理テーブルに登録する．

仮想 TCP ポート番号は，TCP コネクション設定要求ごとに 1 つ取得され，受信 IP パケットとコネクション管理テーブルを関係づけるキーとして使用するためユニークとなるよう管理する．すなわち，割り振られた仮想 TCP ポート番号は，その TCP コネクションが解放するまで再利用されないようにする．

(2) プロキシ-APPL

プロキシ-APPL は，3.4 節で記述した TCP コネクションの確立処理を行うとともに，透過 TCP-Agent から送信されてくるパケットを受信し，TCP で運ばれる情報のトランスペアレントな中継を行う．このことにより，通常の HTTP の処理に加え，HTTP リクエストパイプラインの中継をはじめとして，任意の AP プロトコルの中継を可能としている．

4.2 評価環境

評価環境を図 7 に示す．W-CDMA パケット網には，W-CDMA パケット網を擬似するシミュレータを使用し，クライアント，ステルス型 GW，サーバを接続した．

クライアントには，Windows2000 に W-TCP のプロファイルを適用し，WWW ブラウザとして Internet Explorer 5 にリクエストパイプラインのプラグインを実装したものをを用いた．WWW サーバには FreeBSD 4.4 (通常の TCP) 上で Apache (WWW サーバソフトウェア) を動作させたものをを用いた．ま

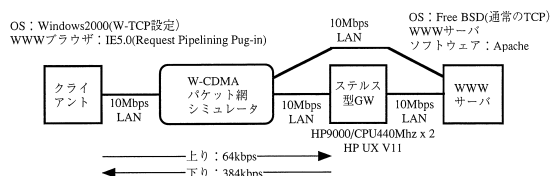


図 7 実証評価システム構成
Fig. 7 Configuration of evaluation system.

た，ステルス型 GW は，商用 UNIX (ハードウェア : HP9000/CPU440 Mhz × 2, OS : HP-UX 11.1) 上に構築した．W-CDMA パケット網シミュレータは上り 64 kbps, 下り 384 kbps とし，RTT に関しては，200 msec から 1,000 msec まで変化させ，ステルス型 GW の効果を確認した．なお，本評価環境では，ネットワークの輻輳は考慮していないためパケットロスは発生しない．

実験方法は，Internet Explorer より 2 Mbyte のバイナリファイルのダウンロードと，8 kbyte の画像ファイルを 100 個貼り付けた HTML ファイルの取得の 2 通りを，それぞれステルス型 GW を経由した場合と，しない場合の 2 通りのルートで測定を行った．

4.3 評価結果と考察

(1) 性能評価結果とその考察

2 Mbyte のバイナリファイルのダウンロードを行った場合の RTT とスループットの関係を図 8 に示す．GW を経由しない場合 (W-TCP を適用しない場合)，RTT が小さい場合 (200 msec 程度) では，ほぼ回線の持つ帯域幅の理論限界値に近い値が得られているが，RTT が増えるに従い TCP の受信バッファの枯渇が発生し，急激に低下する．たとえば，FOMA パケットサービスの RTT の実測値は約 600 msec 程度であり，W-TCP を適用しなければ FOMA の帯域幅を効率的に利用できていないことが分かる．一方，GW を経由し W-TCP と TCP の相互変換をした場合，RTT を 1,000 msec まで増やした場合でも 300 kbps を超える値が得られており，GW を経由しない場合に比べ，約 2 ~ 3 倍の性能改善が可能なが分かる．なお，

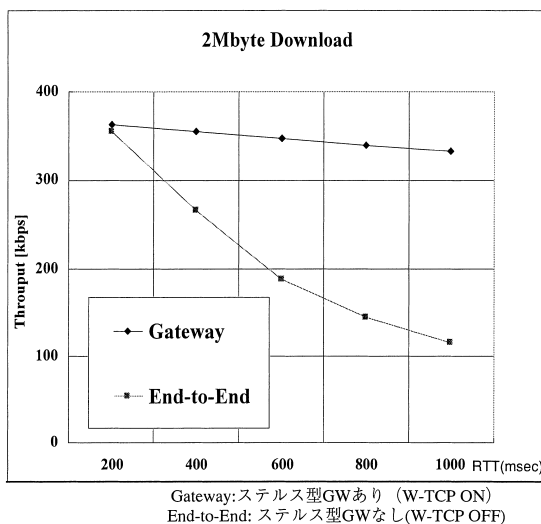


図 8 HTTP によるファイルダウンロードのスループット
Fig. 8 File download throughput by HTTP.

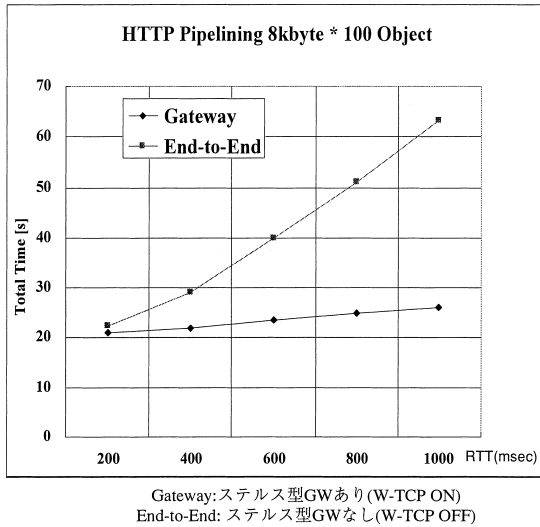


図9 HTTPによるホームページアクセスの応答時間
Fig.9 Response time of WWW access.

End-EndにW-TCPを適用した場合も図8のGW方式と同等の評価結果が得られており、GWのオーバーヘッドによるスループット低下はほとんど認められなかった。

また、実際のW-CDMAパケット網では、ネットワーク輻輳等によるパケットロスが発生することが想定されるが、W-TCPでは、選択的再送機構を使用しており、より効果的に誤り回復が行われ、性能の改善率がより顕著になることが期待できる。

HTTPリクエストパイプラインングでHTMLファイルを取得した場合の遅延時間と応答時間(すべてのオブジェクトをダウンロードし表示し終わるまでの時間)を図9に示す。RTTが小さい場合(200msec程度)では、GWの効果はないが、RTTが大きくなるに従い、2~3倍の性能改善が可能となることが分かる。リクエストパイプラインングを適用するとバルクデータのダウンロードとほぼ同じ状況になり(本実験の場合は800kbyteのファイルダウンロード相当)、W-TCPの受信バッファ拡大の効果が顕著に表れている。一般に、リクエストパイプラインングは、RTTが大きく、Webのページを構成するオブジェクトのサイズが小さく、かつオブジェクト数が多いときに有効である。これに対して、W-TCPを適用するときには、RTTが大きく、転送データ量(オブジェクトサイズ×オブジェクト数)が大きいかほど効果があり、リクエストパイプラインングがより有効に機能することが予想できる。

さらに、GWを設置することのオーバーヘッドを測

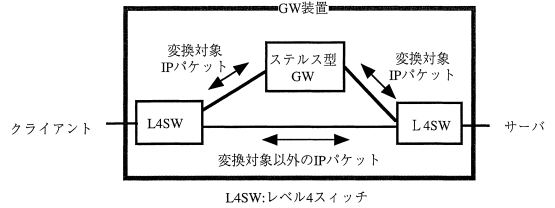


図10 GW装置の構成
Fig.10 Configuration of GW system.

定するため、GWのCPUに負荷をかけた場合の、処理遅延時間(IPパケットを受信してから、回線に送り出すまでの時間)を測定した。処理遅延時間は、数msec(CPU使用率:0%)~数十msec(CPU使用率:70%)程度であった。このオーバーヘッドは、GWでの変換中継処理部をカーネル内に実装する方式(3.2節の方式2)の方がより小さくなることが期待できるが、スループットの評価結果からAPPLで実現する提案方式(3.2節の方式1)でも、ほぼ無視できるといえる。すなわち、提案方式は、欠点であるオーバーヘッドは無視できるくらい小さく、かつその特徴(保守性・運用性等に優れる)を十分生かすことが可能であることを意味している。

これらの評価結果により、提案したステルス型GW方式は、End-EndレベルのIPパケットの透過性を実現しつつ、W-TCPにより、大幅な性能(スループット、応答時間)の改善を可能とする実用的な方式であることが分かった。

(2)GWシステム構成法に関する考察

GW装置として実際のシステムを構築するためには、TCPだけではなく広くインターネットで使用されているUDP¹⁸⁾やIPsec¹⁹⁾等の各種プロトコルに対応する必要がある。そのためには、GW装置の前段にレイヤ4スイッチ(L4SW)を設置し、変換対象とするIPパケットのみをステルス型GWに方路制御し、他のIPパケットは、2台のL4SWを直接つなぐ回線を設定し、そこに方路制御することで変換対象としないことが実用的な構成と考える(図10)。

また、W-CDMA網は大きく無線網とコアネットワークからなり、GW装置の設置場所として、無線網の終端、コアネットワークの終端、等が考えられる。実用化にあたっては、GW開発の簡素化・効率化を図ること、ネットワークの輻輳や故障が発生したときに容易に回避が可能なこと等、を考慮する必要がある。そのためには、コアネットワークから独立させ、コアネットワーク終端(インターネットとの接続ポイント)に設置するのが最も適切と考えられる(図11)。この



図 11 W-CDMA パケット網への導入位置

Fig. 11 Implementation of GW system in W-CDMA network.

場合，GW 装置にトラフィックが集中することになるため，負荷分散や信頼性が課題となり，実用化の際のシステム設計時に配慮する必要がある。

5. 終わりに

本論文では，W-CDMA パケット網を介して，インターネットアクセスを高速に行う場合の GW 方式について述べた．具体的には，End-End でアドレス情報が透過となる W-TCP と TCP の相互変換を行うステルス型 GW 方式を提案するとともに，実装評価を行った．その結果，インターネット上に多数存在する W-TCP が適用されていないサーバに対するアクセスにおけるスループットと応答時間を，提案方式で 2~3 倍程度に改善できることを明らかにした．さらに，GW での処理遅延時間は，数 msec (CPU 使用率：0%)~数十 msec (CPU 使用率：70%) 程度であり，提案方式は，欠点であるオーバヘッドは無視できるくらい小さく，かつその特徴 (保守性・運用性等に優れる) を十分生かすことが可能である実用的な方式であることを明らかにした．

実際の GW の運用に際しては，1 つの GW でいくつまでの TCP セッションの同時運用が可能であるか等のスケーラビリティが重要である．このため，GW 装置のハードウェア/OS を含めたプラットフォームの性能と，同時にサポートできる TCP コネクション数の関係については，今後の課題としたい．同様に，実際の W-CDMA パケット網では，コアネットワークの輻輳等によるパケットロスが発生することが予想されるため，FOMA 等の実環境を利用した評価を今後実施することとしたい．

謝辞 日頃，ご指導いただく (株) NTT ドコモマルチメディア研究所中野博隆所長に感謝する．

参考文献

- 1) 石川太郎，稲村 浩，高橋 修：W-CDMA 向け TCP プロファイル，情報処理学会 MBL/ITS 合同研究会 (Nov. 2000).
- 2) Bakre, A. and Badrinath, B.R.: Handoff and Systemsec Support for Indirect TCP/IP,

- Proc. 2nd USENIX Symposium on Mobile and Location-Independent Computing*, Ann Arbor, Michigan, April 10-11 (1995).
- 3) Brown, K. and Singh, S.: M-TCP: TCP for Mobile Cellular Networks, *ACM Computer Communication Review*, Vol.27, No.5, pp.19-43 (1997).
- 4) Kojo, M., Raatikainen, K., Liljeberg, M., Kiiskinen, J. and Alanko, T.: An Efficient Transport Service for Slow Wireless Telephone Links, *IEEE Journal on Selected Areas of Communication*, Vol.15, No.7 (1997).
- 5) Yavatkar, R. and Bhagawat, N.: Improving End-to-End Performance of TCP over Mobile Internetworks, *Proc. Workshop on Mobile Computing Systemsec and Applications* (Dec.1994).
- 6) Fielding, R., Gettys, J., et al.: Hypertext Transfer Protocol — HTTP/1.1, RFC2616 (June 1999).
- 7) 関口克己，北口雅哉，鶴巻宏治，高橋 修：モバイルインターネットアクセスにおけるリクエストパイプラインの有効性，情報処理学会 MBL/ITS 合同研究会 (Nov. 2000).
- 8) Allman, M.: On the New Generation and Use of TCP Acknowledgements, *Computer Communication Review*, Vol.28, No.5 (1998).
- 9) Balakrishnan, H., Padmanabhan, V., et al.: The effects of asymmetry on TCP performance, *ACM Mobile Networks and Applications (MONET)*, Vol.4, No.3, pp.219-241 (1999).
- 10) Inamura, H., Montenegro, G., et al.: TCP over 2.5 G and 3G Wireless Networks, Internet draft, draft-ietf-pilc-2.5g3g-07-txt (2002).
- 11) 稲村 浩，石川太郎，高橋 修，渥美幸雄：W-CDMA 網でのリンク層 ARQ と TCP の特性評価，情報処理学会論文誌，Vol.43, No.12 (2002).
- 12) Jacobson, V., Braden, R. and Borman, D.: TCP Extensions for High Performance, RFC1323 (May 1992).
- 13) Allman, M. and Floyd, S.: Increasing TCP's Initial Window, RFC2414, (Sep. 1998).
- 14) Mathis, M., Mahdavi, J., et al.: TCP Selective Acknowledgment Options, RFC2018 (Oct. 1996).
- 15) 原 政博，中川 格，藤野信次：携帯電話網での Wireless Enhanced TCP の実装，情報処理学会 DICO2001 (2001).
- 16) 長谷川輝之，長谷川亨，加藤聰彦，鈴木健二：広域 ATM 網を介した LAN 間接続のための TCP ゲートウェイの実装と性能評価，信学論，Vol.J79-B-I, No.5, pp.262-270 (1996).
- 17) 三宅 優，長谷川輝之，長谷川亨，加藤聰彦：TCP によるデータ転送速度を高速化する衛星イ

ンターネット用 TCP ゲートウェイの提案, 信学論, Vol.J84-B, No.12, pp.2330-2341 (2001).

- 18) Postel, J.: User Datagram Protocol, RFC768 (Aug. 1980).
 19) Kent, S. and Atkinson, R.: Security Architecture for the Internet Protocol, RFC2401 (Nov. 1998).

(平成 14 年 7 月 11 日受付)

(平成 14 年 12 月 3 日採録)



高橋 修 (正会員)

1975 年 3 月北海道大学大学院工学研究科修士課程修了。同年 4 月電電公社 (現 NTT) 入社。情報通信研究所でコンピュータネットワークアーキテクチャの研究開発, および

OSI の標準化に従事。1999 年 NTT ドコモに異動。マルチメディア研究所にてモバイルインターネットの研究開発に従事。電子情報通信学会会員。



関口 克己

1997 年群馬大学大学院工学研究科博士前期課程電気電子工学専攻修了。同年 NTT ドコモ入社。以来, 移動パケット通信システム, モバイル VPN プロトコル, ゲートウェイシ

ステムの研究開発に従事。現在, 同社マルチメディア開発部主査。電子情報通信学会会員。



鶴巻 宏治 (正会員)

1984 年名古屋大学工学部卒業。同年日本電信電話公社入社。1999 年 NTT 移動通信株式会社入社。現在, (株)NTT ドコモ東海に在籍。日本ソフトウェア科学会会員。



稲村 浩 (正会員)

1990 年慶應義塾大学大学院理工学研究科計測工学専攻修士課程修了。同年日本電信電話(株)入社。分散トランザクションシステム, 分散ファイルシステムの研究開発に従事。1998

年より NTT ドコモ。モバイル環境におけるトランスポートプロトコルに関する研究開発に従事。ACM 会員。



渥美 幸雄 (正会員)

1975 年慶應義塾大学工学部電気工学科卒業。1977 年同大学大学院修士課程修了。同年電電公社 (現 NTT) 横須賀電気通信研究所入社。主に通信プロトコル, 通信制御ソフトウェアの研究開発に従事。1994 年 (株) 超高速ネットワ

ーク・コンピュータ技術研究所。プロトコルアーキテクチャの研究に従事。1999 年より (株)NTT ドコモ・マルチメディア研究所に勤務。次世代のモバイルインターネット方式の研究開発に従事。博士 (情報工学)。電子情報通信学会会員。



水野 忠則 (正会員)

1968 年名古屋工業大学経営工学科卒業。同年三菱電機(株)入社。1993 年静岡大学工学部情報知識工学科教授。現在, 同大学情報学部情報科学科教授。工学博士。情報ネットワ

ーク, モバイルコンピューティング, 放送コンピューティングに関する研究に従事。著書としては「プロトコル言語」(カットシステム)、「コンピュータネットワーク概論」(ピアソン・エデュケーション)等がある。電子情報通信学会, IEEE, ACM 各会員。当会フェロー。