

マルチホップ衛星通信システムにおける HTTP の性能評価

小畑 博靖[†] 竹内 達^{††} 舟阪 淳一[†] 石田 賢治[†]
廣瀬 功一^{†††} 山崎 克之^{†††}[†] 広島市立大学 情報科学部 情報工学科, ^{††} 広島市立大学大学院 情報科学研究科 情報工学専攻, ^{†††} KDDI 研究所
[†], ^{††} 〒 731-3194 広島市安佐南区大塚東 3-4-1, ^{†††} 〒 356-8502 埼玉県上福岡市大原 2-1-15
E-mail: [†] {obata, funa, ishida}@ce.hiroshima-cu.ac.jp, ^{††} takeuchi@nets.ce.hiroshima-cu.ac.jp,
^{†††} {hirose, yamazaki}@kddilabs.jp

あらまし

地球を半周するような長距離通信を行う手段として、複数の衛星回線を利用するマルチホップ衛星通信システムが注目されている。また、マルチホップ衛星通信システムは、災害時等における海底光ファイバの代替路として有用である。しかしながら、マルチホップ衛星通信システムにおける、HTTP (HyperText Transfer Protocol) 性能の理論的および実験的評価は行われていない。

そこで本稿では、マルチホップ衛星通信システムにおける HTTP 性能の理論的、および、実験的評価を行った。まず、マルチホップ衛星通信システムにおける、HTTP1.1 による Web データ送信時間を求める新たな評価式を導出した。また、パイプライン要求機能に対応した評価式の導出も行った。次に、得られた評価式の値と、マルチホップ衛星通信システムの1つである SKYCAST システムを運用して得られた実測値とを比較評価した。その結果、評価式から得られた値は実測値に良く近似していることが分かった。

キーワード

マルチホップ衛星通信システム, Web データ送信, HTTP1.1, パイプライン要求

On HTTP Performance over Multi-Hops Satellite Communication System

Hiroyasu Obata[†] Satoru Takeuchi^{††} Junichi Funasaka[†] Kenji Ishida[†]
Kouichi Hirose^{†††} Katsuyuki Yamazaki^{†††}[†] Department of Computer Engineering, Faculty of Information Sciences, Hiroshima City University
^{††} Graduate School of Information Sciences, Hiroshima City University ^{†††} KDDI R&D Laboratories Inc.
[†], ^{††} 3-4-1 Ozuka-higashi, Asa-minami-ku, Hiroshima 731-3194, Japan
^{†††} 2-1-5 Ohara, Kami-fukuoka-shi, Saitama 356-8502, Japan
E-mail: [†] {obata, funa, ishida}@ce.hiroshima-cu.ac.jp, ^{††} takeuchi@nets.ce.hiroshima-cu.ac.jp,
^{†††} {hirose, yamazaki}@kddilabs.jp

Abstract

In order to provide a long distance communication such as the one between two nodes separated by about halfway around the earth, a multi-hops satellite communication system which goes through multiple satellites is attracted. Furthermore, to cope with disasters, the system is useful as the alternative links of submarine optical fiber links. However, the existing researches have not evaluated the HTTP (HyperText Transfer Protocol) performance on such multi-hops system both theoretically and experimentally.

Therefore, we evaluate web data transfer time based on HTTP over the multi-hops satellite communication system. Firstly, we propose a new formula for the HTTP1.1 performance over the system. Secondly, we also propose a formula taking pipeline requesting into account. Finally, the calculation results are compared with the experimental ones based on the SKYCAST system, which is one of the multi-hops satellite communication systems. As a result, it is shown that the values based on the evaluation formulas are close to those from the experimental results.

key words

Multi-Hops Satellite Communication System, Web Data Transfer, HTTP1.1, Pipeline Requesting

1. はじめに

1つの通信衛星経由のみでは地球を半周するような離れた2地点間を結ぶ通信サービスの提供は困難である。このような長距離通信を可能とする技術としてマルチホップ衛星通信システムが注目されている。また、衛星通信システムが持つ耐災害性により、マルチホップ衛星通信システムは海底光ファイバの代替路として重要である。また、Webシステムの普及により、アプリケーションプロトコルとしてHTTPの利用が多くなっている。しかしながら、このようなマルチホップ衛星通信システムにおけるHTTPを用いたWebデータ送信時間の評価を理論的および実験的に行っている研究はほとんどない。

そこで本稿では、マルチホップ衛星通信システムにおけるHTTPの性能評価を理論的および実験的に行った。まず、マルチホップ衛星通信システムにおけるHTTPによるWebデータ送信時間を求める新たな評価式をパイプライン要求機能の有無に分けて導出した。ここで、導出した評価式は文献[1]の拡張となっている。次に、提案した評価式より得られた結果とマルチホップ衛星通信システムの1つであるSKYCASTシステム[2]による実測値を比較評価した。その結果、提案したHTTP評価式より得られた値は実測値によく近似していることが分かった。しかしながら、最初の実験では実測値と評価式の値は完全には一致していなかったため、その差の原因について考察した。その結果、差はSKYCASTシステム中のSKYCASTハブ局内の処理時間によって生じていることが分かった。このSKYCASTハブ局内の処理時間を考慮したところ、評価式の精度は極めて良いものになった。

以下、2章では本研究で対象としたネットワークモデル、および関連研究を示す。3章では導出したHTTP評価式について説明する。4章では導出したHTTP評価式に基づく値と実測値との比較を行い、比較結果について考察する。最後に、5章でまとめを行う。

2. 準備

ここでは、ネットワークモデルと関連研究について示す。

2.1 ネットワークモデル

図1に本稿で対象とするマルチホップネットワークを示す。サーバからのデータはサーバとクライアント間の衛星回線(下り)を $N(\geq 2)$ 回経由してクライアントへ送られる。また、同様にクライアントからの確認応答も衛星回線(上り)を N 回経由してサーバへ届けられる。図1において、中継ノードはマルチホップネットワークにおける通信衛星または中継地球(ハブ)局を表す。また、クライアントからサーバへの上り回線の帯域を $B_u(m)$ ($m = 1, 2, \dots, N$)、サーバからクライアントへの下り回線の帯域を $B_d(m)$ ($m = 1, 2, \dots, N$)とする。上りの伝搬遅延を $\tau_u(m)$ ($m = 1, 2, \dots, N$)、下りの伝搬遅延を

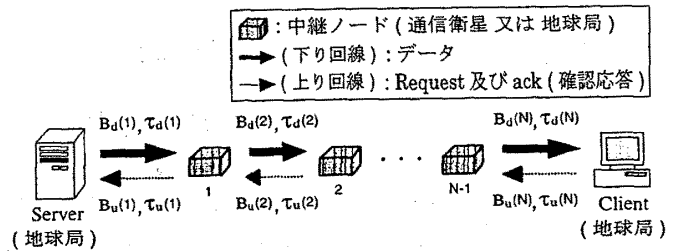


図1: マルチホップネットワーク

$\tau_d(m)$ ($m = 1, 2, \dots, N$)とする。さらに、データセグメントサイズを P_{seg} 、ackセグメントサイズを P_{ack} 、および、TCPとIPの合計ヘッダサイズを P_{header} とする。

本稿で対象とするマルチホップネットワークは、閉じたネットワークであり、帯域が大きく変動せず確保されるという仮定を置く。さらに、本稿ではパケットロスを考慮しない。これらの仮定は通信衛星を用いた実験結果[1]に基づいており、現実的な仮定である。

2.2 関連研究

衛星通信システム上におけるHTTP1.1の性能評価について既にいくつかの報告がある。まず、文献[1]はHTTP1.1[3]に基づくWebデータ送信時間を求める評価式を導出し、実測値との比較評価を行っている。さらに、HTTP1.1のパイプライン要求機能を用いた測定、および、評価式の導出も行っている。次に、文献[4]ではHTTP1.1のパイプライン要求機能を用いた時のWebデータ送信時間を理論的、および、実験的に評価している。さらに、輻輳ウィンドウの初期値を変化させた時のHTTP1.1の評価も行っている。しかしながら、これらの文献は、いずれも1ホップ衛星通信システムを利用した場合のHTTPの性能評価にとどまっている。

マルチホップ衛星通信ネットワークに関する研究としては文献[5]がある。文献[5]は、マルチホップ衛星通信ネットワークにおけるTCPの性能をシミュレーションにより評価している。この評価は、通信衛星の切り替えによるハンドオーバー発生時における複数コネクションに対する帯域割り当ての公平性に着目して行われている。しかしながら、マルチホップ衛星通信システムにおけるHTTPの性能評価は行われていない。また、文献[5]は中継ノードとして通信衛星のみを考慮している。

以上のように、中継ノードとして通信衛星と中継地球(ハブ)局を合わせて考慮したマルチホップ衛星通信システムにおけるHTTP性能評価は未だ行われていない。

3. マルチホップ衛星通信システムに対応した新たな評価式

マルチホップ衛星通信システムに対応した新たなHTTP1.1の評価式を導出する。また、パイプライン要求機能を考慮した場合の評価式も導出する。

3.1 HTTP1.1 動作概要

まず、HTTP1.1の動作概要をパイプライン要求機能の有無に分けて述べる。

図2はHTTP1.1でパイプライン要求がない場合とある場合のWebデータ送信の様子を示す。まず、クライアントにおいてWebデータ要求が発生すると、接続の確立が行われる。次に、クライアントはサーバに向けてHTTPリクエストを行い、その後、サーバはTCPによるデータ送信を行う。ここで、(a)パイプライン要求がない場合、(b)ある場合、の一連のデータ送信の違いを以下に簡単にまとめる。

- (a) まず、HTMLファイル送信の際に接続設定を行う。その後、クライアントは要求したWebページに含まれる複数データの内1つのデータに対する要求(HTTP get)を行い、サーバからの応答を待ってから新しい要求を行う動作を繰り返す[3]。
- (b) 接続の設定後、要求したWebページに含まれる複数のデータに対する送信要求をサーバからの応答を待たずに連続して行う。データ要求を受け取ったサーバは、要求を受け取った順番でクライアントにデータを送信する[3]。

以上より、HTTP評価式はTCPによるデータ送信時間に接続設定時間を加えることで導出可能である。なお、HTTP評価式は、クライアントがHTMLで書かれたWebページ(以降、HTMLファイルと呼ぶ)にアクセスし、そのページに含まれる複数の画像等のデータをすべて受信するまでの時間を計算する式である。

3.2 評価式の導出

ここでは、評価式の導出を3つの部分に分けて説明する。まず、(I)でHTTPによる接続設定、Webデータ要求に要する時間を求める式を示す。次に、(II)でTCPによるWebデータ送信時間を導出する式を示す。最後に、(III)で(I)(II)の式を用いてHTTP1.1によるWebデータ送信時間を求める式を示す。

なお、本評価式の特徴は、(イ)Webデータのような小さなデータを複数送信する場合に対応させるため、ウィンドウサイズの時間変化を細かく考慮してTCPによるデータ送信時間を導出している点、(ロ)各リンクの伝搬遅延、および、帯域を考慮している点、にある。

(I) 接続設定、データ要求に要する時間

$P_{connect}$ を接続設定時、または、Webデータ要求時にやり取りされるパケットのサイズとする。図2より、接続設定時間は、パケットの送出時間と各リンクの伝搬遅延時間の総和となり、(1)式となる。ただし、上り・下り回線それぞれのボトルネックリンクの帯域を B_u 、 B_d とする。従って、 B_u 、 B_d はそれぞれ、

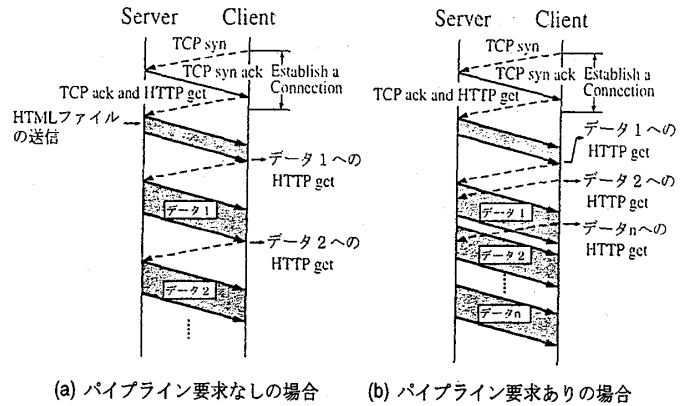


図2: HTTP1.1でパイプライン要求の有無によるWebデータ送信の様子

$B_u = \min(B_u(m))$ 、 $B_d = \min(B_d(m))$ となる。ただし、 $\tau_u = \sum_{m=1}^N \tau_u(m)$ 、 $\tau_d = \sum_{m=1}^N \tau_d(m)$ とする。

$$T_{connect} = \frac{P_{connect}}{B_d} + \frac{2P_{connect}}{B_u} + \tau_d + 2\tau_u \quad (1)$$

また、HTTP1.1では2度目以降からのWebデータ要求では接続設定を行わないため、この時のデータ要求に要する時間 $T'_{connect}$ は(2)式ようになる。

$$T'_{connect} = \frac{P_{connect}}{B_u} + \tau_u \quad (2)$$

(II) TCPによるデータ送信時間

次に、TCPによるデータ送信時間を求める式を示す。

TCPによるデータ送信時間は、ある時点でのウィンドウサイズによるデータ送信と次のウィンドウサイズによるデータ送信までの間隔(送信間隔)をデータ送信終了まで足し合わせることで導出できる。この送信間隔 $t(k)$ は、ラウンドトリップ時間 r_{tt} 、 k 個目のデータ送信時におけるウィンドウサイズ $w(k)$ 分のデータを送信する時間 $a(k)$ 、ウィンドウサイズ $w(k)$ 分のackセグメントを送信する時間 $q(k)$ の3つのパラメータの内最大値を取ることで求まる[1]。ここで、 k を1つのWebデータの送信に必要なTCPによるデータ送信回数とする。

まず、 r_{tt} は(3)式により得られる。ただし、ウィンドウサイズが1(初期値)の時は(4)式となる。

$$r_{tt} = \frac{2(P_{seg} + P_{header})}{B_d} + \frac{P_{ack}}{B_u} + \tau_d + \tau_u \quad (3)$$

$$r_{tt} = \frac{P_{seg} + P_{header}}{B_d} + \frac{P_{ack}}{B_u} + \tau_d + \tau_u + d \quad (4)$$

また、 $a(k)$ 、 $q(k)$ はそれぞれ(5)式、(6)式となる[1]。

$$a(k) = \frac{(P_{seg} + P_{header}) \cdot w(k)}{B_d} \quad (5)$$

$$q(k) = \frac{P_{ack} \cdot \lceil \frac{w(k)}{2} \rceil}{B_u} \quad (6)$$

次に、 j 個目の Web データ $F(j)$ を送信するために要する時間を示す。ただし、送信 Web データ数を n とすると、 $0 < j \leq n$ である。

まず、コネクション設定後の Slow Start アルゴリズムが適用される (初期状態における)、 j 個目の Web データ送信に要する時間 $T_{slow}(j)$ は (7) 式で与えられる。

$$T_{slow}(j) = \sum_{k=1}^{th} t(k) + \tau_d + \frac{(F(j) - \sum_{k=1}^{th} w(k) \cdot P_{seg}) + P_{header}}{B_d} \quad (7)$$

ただし、閾値 th は (8) 式により得られる。

$$\sum_{k=1}^{th} \{w(k) \cdot P_{seg}\} \leq F(j) < \sum_{k=1}^{th+1} \{w(k) \cdot P_{seg}\} \quad (8)$$

次に、ウィンドウサイズ $w(k)$ が最大ウィンドウサイズとなった時の j 個目の Web データ送信時間 $T_{mws1}(j)$ は、(9) 式により得られる。ただし、 t_{max} は最大ウィンドウサイズによりデータ送信される時の送出間隔を示す [1]。

$$T_{mws1}(j) = \lfloor \frac{F(j)}{MWS} \rfloor \cdot t_{max} + \tau_d + \frac{(F(j) \bmod(MWS)) + P_{header}}{B_d} \quad (9)$$

さらに、パイプライン要求がある場合、HTML ファイルに含まれる複数のデータに対して連続してデータ要求を行う。そのため、サーバから送出される複数のデータを送信するために要する時間は単一のデータ $F_{pipe}(= \sum_{j=2}^n F(j))$ を送信する時間とみなすことで導出できる [1]。従って、パイプライン要求がある場合の Web データ送信時間 T_{mws2} は (10) 式となる。

$$T_{mws2} = \lfloor \frac{F_{pipe}}{MWS} \rfloor \cdot t_{max} + \tau_d + \frac{(F_{pipe} \bmod(MWS)) + P_{header}}{B_d} \quad (10)$$

(III) HTTP1.1 によるデータ送信時間

HTTP1.1 でパイプライン要求がない場合、 n 個の Web データを送信するのに要する時間 $T_{nopipeline}$ は、コネクション設定時間 $T_{connect}$ 、2 個目以降のデータ要求に要する時間 $(n-1)T'_{connect}$ 、HTML ファイル送信に要する時間 $T_{slow}(1)$ 、および、HTML ファイルに含まれる 2 個目以降のデータ送信に要する時間の和で求められ、(11) 式となる。

$$T_{nopipeline} = T_{connect} + (n-1)T'_{connect} + T_{slow}(1) + \sum_{j=2}^n T_{mws1}(j) \quad (11)$$

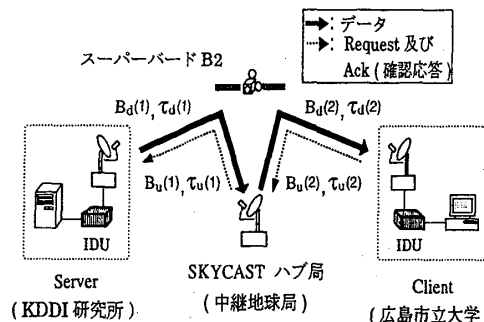


図 3: SKYCAST システムを用いた 2 ホップ衛星通信システム

表 1: 2 ホップ衛星通信システムのパラメータ

データサイズ (KByte)	5,10,20
データセグメントサイズ P_{seg} (Byte)	988
ACK セグメントサイズ P_{ack} (Byte)	40
$B_u(2), B_d(1)$ (Kbps)	150
$B_u(1), B_d(2)$ (Kbps)	768
$\tau_u(k), \tau_d(k)$ ($k=1, 2$) (msec)	248

次に、パイプライン要求がある場合、 n 個の Web データを送信する時間 $T_{pipeline}$ は、コネクション設定時間 $T_{connect}$ 、2 個目以降のデータ F_{pipe} の要求に要する時間 $T'_{connect}$ 、HTML ファイルを送信する時間 $T_{slow}(1)$ 、および、HTML ファイルに含まれる 2 個目以降のデータ送信に要する時間の和で求められる [1]。従って、 $T_{pipeline}$ は (12) 式となる。

$$T_{pipeline} = T_{connect} + T'_{connect} + T_{slow}(1) + T_{mws2} \quad (12)$$

4. 実験的評価

ここでは、マルチホップ衛星通信システムにおける HTTP1.1 の性能評価を、3 章で求めた評価式と 2 ホップ衛星通信システムである SKYCAST システムを用いて行う。

4.1 ネットワークパラメータ

図 3 に評価実験に用いた 2 ホップ衛星通信システムの概要を示す。実験環境は、VSAT (Very Small Aperture Terminal) 衛星通信システムを用いた SKYCAST システムである。サーバからのデータは中継地球局である SKYCAST ハブ局を経由しクライアントへ送られる。クライアントからの確認応答も SKYCAST ハブ局を経由しサーバへ届けられる。

なお、SKYCAST システムでは $B_u(2)$ 、 $B_d(1)$ がボトルネックリンクの帯域となる。

表 1 に 2 ホップ衛星通信システムのパラメータを示す。なお、Web サーバには Apache 2.0.40 [6] を使用し、クライアントプログラムとして W3C Libwww 5.4.0 内のロボット (*webbot*) [7] を使用した。

4.2 実験結果

図4は1つのWebページに含まれるデータのサイズを5KByteに固定し、そのページに含まれるデータ数を10個、20個、30個と変化させた時のデータ送信時間を示す。図5は、1つのWebページに含まれるデータ数を10個に固定し、各データサイズを5KByte、10KByte、20KByteと変化させた時のデータ送信時間を示す。

まず、2ホップ衛星通信システムにおけるHTTP1.1の性能についてWebデータ送信時間に着目し、パイプライン要求がない場合とある場合とを比較する。図4の実測値より、パイプライン要求がない場合、送信するデータ数が増加するにつれ送信時間が急激に増加している。一方で、パイプライン要求がある場合では、データ数が増加しても送信時間の増加の割合は、パイプライン要求がない場合の約5分の1に抑えられている。これは、以下のような理由に起因すると考えられる。図2から分かるように、パイプライン要求がない場合、クライアントによるデータ要求とサーバからの応答が繰り返されるため、1セッション毎にクライアント側ではサーバからの応答待ち時間が、サーバ側ではクライアントからの要求待ち時間が生じてしまう。それに対して、パイプライン要求がある場合は、クライアントはサーバからの応答を待たずに複数のデータに対する送信要求を行うことができる。従って、データ数に比例して時間が大幅に削減される。

次に、図5の実測値より、送信するデータのサイズを増加させた場合、データ数を変化させた時ほどではないが、パイプライン要求がある方が平均で約15秒送信時間が短い。

以上より、評価式の値だけではなく実測値からも、2ホップ衛星通信システムのようなマルチホップ衛星通信システムにおいてHTTP1.1の仕様にあるパイプライン要求機能を利用する効果は高いことが確認できる。また、図4、図5より、評価式による値と実測値がよく近似していることが分かる。

4.3 考察

図4、図5より、特にパイプライン要求がない場合、評価式の値と実測値の間に差があることが分かる。そこで、ここではSKYCASTシステムを用いたHTTPの評価実験で生じた現象から考えられる評価式の値と実測値の差の原因について考察する。

まず、本実験に用いたSKYCASTシステムにおけるデータ送受信の様子を説明する。SKYCASTシステムでは、サーバがクライアントに向けてデータを送出した際に、サーバ側に設置してあるIDU (In Door Unit : 図3参照) からサーバに対して擬似的に確認応答 (以降、擬似ackと呼ぶ) を返送するよう設計されている。これは、衛星通信システムを用いた通信において、伝搬遅延により

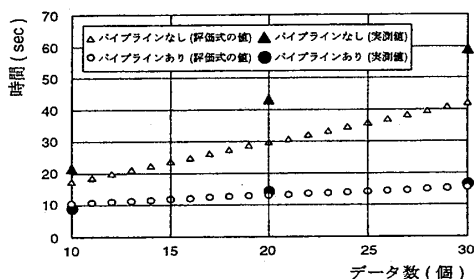


図4: データの数を变化させた時のWebデータ送信時間

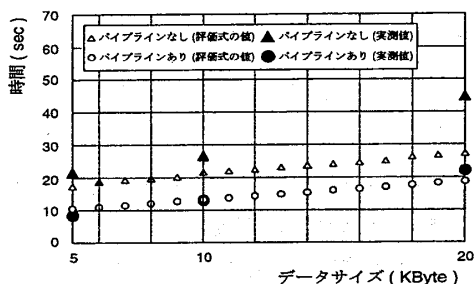


図5: データサイズを変化させた時のWebデータ送信時間

スループットが大きく低下するのを防ぐために用いられ、高速なデータ送信を可能にしている。

次に、パイプライン要求がない場合、 n 個のデータの内の j 個目のデータにおけるパケット送受信の様子を示す(図6参照)。まず、 j 個目のデータに対する送信要求がクライアントにより行われる(図6の[a])。サーバによる j 個目のデータ送信は、サーバ側のIDUが返送する擬似ackにより高速に行われる(図6の[b])。しかし、サーバはクライアントがデータの受信を完了し、自身が次($j+1$ 個目)のデータに対するクライアントからのHTTP getを受信するまで待たなければならない(図6の[c])。以上の動作概要を考慮しつつ、tcpdumpコマンドを用いて実際のデータ送受信の様子を解析した。

その結果、HTTP getが送出される毎にSKYCASTハブ局では各パケットに対し、マルチホップに伴う処理時間と推定される遅延が加わっていることが分かった(図6の Δt_1 , Δt_2)。具体的には、データサイズが5KByteの場合、クライアントが j 個目のデータに対するHTTP getを送出し、サーバによる j 個目のデータを受信し始めるまで、SKYCASTハブ局では約0.56秒の時間を要する処理が行われていることが分かった。従って、データ1個を要求する毎にSKYCASTハブ局において処理時間が余計に加わっていたことが明らかとなった。

一方、パイプライン要求がある場合は、データ数に関わらずデータサイズが5KByteの時、データ送信が終了するまでのSKYCASTハブ局における処理時間は約0.56秒となっている事が分かった。これは、パイプライン要求がある場合、クライアントはデータが届くのを待たずに

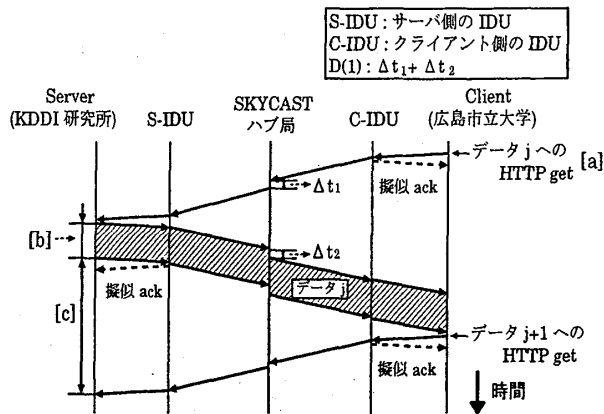


図 6: 擬似 ack によるデータ送信の様子 (パイプライン要求がない場合)

HTTP get を連続して送出するためであると考えられる。つまり、2 個目以降の HTTP get に対する SKYCAST ハブ局における処理時間は Web データ送信時間に吸収されているため、上記の SKYCAST ハブ局における処理時間の要因となる HTTP get は 1 回と考えることができる。

上記の結果を基に、中継ノードにおける処理時間 (図 6 の $\Delta t_1 + \Delta t_2$) を考慮できるように評価式の変更を行った。

各中継ノードにおける処理時間 (図 6 の $\Delta t_1 + \Delta t_2$ にあたる値) を $D(m)$ とすると、補正した評価式は、(11) 式、(12) 式に $D(m)$ を考慮した (13) 式、(14) 式となる。ただし、処理時間を有さない中継ノードについては $D(m) = 0$ とする。また、 $1 \leq m < N$ である。なお、SKYCAST システムでは実測値より、データサイズが 5KByte の時、 $D(1) = 0.56$ 秒である。

$$T_{nopipeline_d} = T_{connect} + (n-1)T'_{connect} + T_{slow} (1)$$

$$+ \sum_{j=2}^n T_{mws1}(j) + n \sum_{m=1}^{N-1} D(m) \quad (13)$$

$$T_{pipeline_d} = T_{connect} + T'_{connect} + T_{slow} (1)$$

$$+ T_{mws2} + \sum_{m=1}^{N-1} D(m) \quad (14)$$

図 7, 図 8 は、処理時間を考慮した (13) 式、(14) 式による値と実測のデータ送信時間を示す。図 7, 図 8 より、(13) 式、(14) 式を適用することで実測値にさらに近い値を導出可能であることが分かる。

5. おわりに

本稿では、マルチホップ衛星通信システムにおいて HTTP1.1 を利用した Web データ送信時間を与える評価式を提案し、その評価を行った。まず、Web データ送信時間を求める新たな評価式の導出においては、パイプライン要求機能を考慮しない場合と考慮する場合の評価式を新たに導出した。本評価式の特徴は、マルチホップネットワークに対応するため、各リンクの帯域、伝搬遅延を

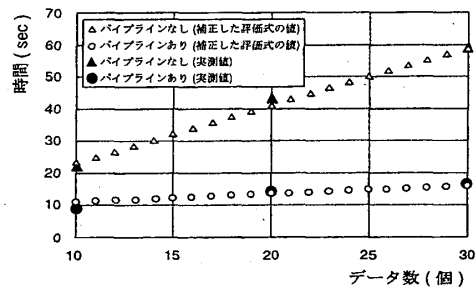


図 7: データの数を変化させた時の Web データ送信時間

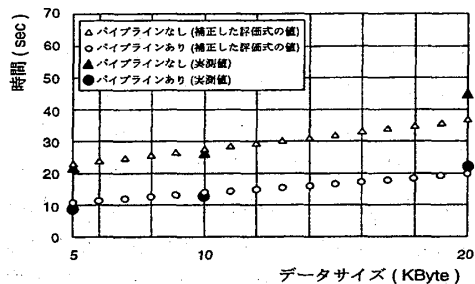


図 8: データサイズを変化させた時の Web データ送信時間

考慮している点、および、各中継ノードにおける処理時間を考慮している点である。次に、提案した評価式の精度を調べるため、マルチホップ衛星通信システムの 1 つである SKYCAST システム [2] を用いた実験的評価を行った。その結果、提案した評価式による値は実測値に良く近似していることが分かった。

謝辞

本研究を進めるにあたり通信実験にご協力頂いた KDDI 研究所 浅見 徹 所長、前島 治氏に深謝致します。

参考文献

- [1] H. Obata, K. Ishida, J. Funasaka, K. Amano: "Data Transfer Time by HTTP 1.0/1.1 on Asymmetric Networks Composed of Satellite and Terrestrial Links," IEICE Trans. Comm., E85-B, No.12, pp.2895-2903, Dec. 2002.
- [2] 河合 直行, 平井 立彦: 「双方向 VSAT/ソフトウェアの開発」, <http://www2.crl.go.jp/mt/b150/SJR/17/SpecialReportJ.pdf>, Oct. 2001.
- [3] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Le: "Hypertext Transfer Protocol HTTP/1.1," IETF RFC2616, June 1999.
- [4] H. Kruse, M. Allman, J. Griner, D. Tran: "Experimentation and Modeling of HTTP Over Satellite Channels," International Journal of Satellite Comm., 19(1), pp.51-68, Feb. 2001.
- [5] T. Tarik, N. Katou, Y. Nemoto: "On TCP Performance over Multi-Hops Satellite Constellations," 信学技報 SAT2002-75, pp.27-32, Oct. 2002.
- [6] The Apache Software Foundation: "Apache," <http://httpd.apache.org/>, Oct. 2002.
- [7] H. Nielsen: "Libwww robot," <http://www.w3.org/>, Oct. 2002.