

## TCP 輻輳制御緩和技術の開発と検証

柳生理子\* 田中 功一\* 高橋 俊悟\*

\*三菱電機(株)情報技術総合研究所

衛星回線などを介したTCP通信における伝送遅延による速度性能劣化の改善を目的とし、高遅延回線前のゲートウェイによる送達確認応答(ACK)の代理応答を用いたアルゴリズムを開発した。このアルゴリズムは速度性能の劣化に対し、非常に有効である一方で、衛星区間における輻輳制御を緩和しているため、衛星区間における輻輳に対する課題があった。本稿においてはこの課題に対し、TCPのSACKによる通信量の削減効果およびオプション性に着目し、SACK適用範囲の拡大による改良案を提案する。

### Development and Verification for Mitigations of Congestion Control Algorithm Over TCP

Riko Yagiu \* Kouichi Tanaka\* Toshinori Takahashi\*

\*Mitsubishi Electric Corp. Information Technology R & D Center

We have developed an algorithm, which addresses the problem of high latency links resulting in poor TCP performance due to the delay of ACKs. We simulated this environment and showed good performance, when using the algorithm. The algorithm, however, uses mitigations of congestion control algorithm over TCP, so there is the other problem with congestion control on satellite links. We add one more improvement using SACK and this strategy can be used to improve the problem.

#### 1 はじめに

近年、衛星回線上でTCPを用いたデータ配信における速度性能の劣化改善に対する要望が高まってきた。

衛星回線は高速かつ大容量の通信網である長所をもつ一方で、物理的伝送距離の長さゆえの伝送遅延がある。この遅延は、長いRTT(Round Trip Time)を必要とする。一方、TCPの輻輳制御機構は、この長いRTTを「回線の輻輳の信号」と解釈し、輻輳を緩和させるために送信量を抑制する。この結果、ネットワークが輻輳していないにも関わらず、TCPは不必要にその転送速度を抑制し、致命的な影響を及ぼす。

これらの問題に対し、様々な技術の開発や提案が行われてきている。例えば、TCPを用いずUDP上に構築された遅延の影響を受けない高信頼のアプリケーションを用いるもの[1][2]、衛星通信向

けの輻輳制御を緩和したTCPなどが挙げられる。しかし、これらの方法は、PC(パーソナルコンピュータ)およびインターネットの普及により、要望の大きなPC上の一般的なアプリケーションを用いた通信の速度性能向上に対しては有効ではない[3][4]。

上述の問題に対し、我々は、TCPの輻輳制御機構の緩和を促すアルゴリズム(TCP Spoofingアルゴリズム)の研究開発を行ってきた。このアルゴリズムは通信系路上に設置するゲートウェイ(GW)に適用することを前提とするものである。このため、ユーザ側のアプリケーションを変更することなく速度性能の劣化を改善を可能とする。

## 2 TCP Spoofing アルゴリズムの概要

### 2.1 TCP Spoofing の動作

図1は、TCP Spoofing アルゴリズムを適応した通信環境例、および動作を示したものである。

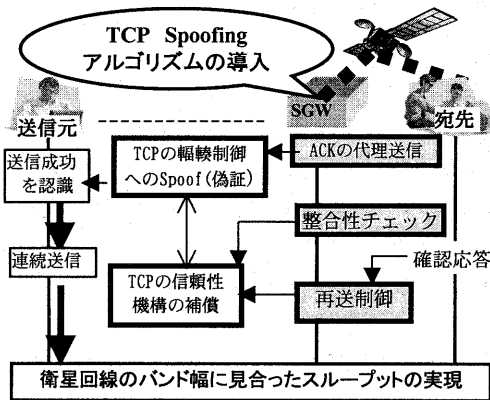


図1 TCP Spoofing アルゴリズム適用環境

我々が開発したアルゴリズムは2セグメント・スプリッティングと呼ばれる方式に分類される形態であり、通信経路を図に示すように衛星区間の手前に設置されたこのアルゴリズムを導入したゲートウェイ (SGW) によって2つに分割する方式である。このGWにおいて、パケットを転送すると共に、受信側 TCP において転送パケットを受信した場合に作成する送達確認応答 (ACK) と同等のもの (擬似 ACK) を作成し、送信側 TCP に対し、返送 (ACK の代理送信) する機能[5][6]である。この ACK の代理送信機能により、送信元の TCP は宛先にまでセグメントが到達したと Spoof (偽証) され、送信 Window をスライドし連続的なデータ送信が可能となる。

#### 2.1.1 通信開始時の動作

TCP は 3-way-hand shake という互いの状況を確認しあうコネクションの確立を行う。これは、通信開始時には、ネットワークおよびは通信先の状況は不明であり、不用意に大量のデータを送信し、ネットワークの輻輳を招く事を避けるための方式である。従って、Spoofing アルゴリズムにおいても、この方針を踏襲し、図2に示すように、ACK

の代理送信は行わず、ホストが作成したパケット

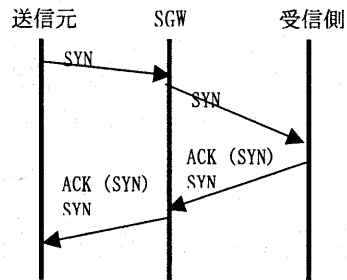


図2 通信開始時動作

の中継する際に、双方の通信管理情報の取得のみを行う。

#### 2.1.2 データ通信時 (接続確立後) の動作

コネクション確立後は、図3に示すように Spoofing 適用対象の通信に所属するデータパケットを受信した場合、宛先への転送と共に、送信元-宛先間における配信結果の整合性が保たれている範囲でのパケットあれば、擬似 ACK の作成・送信を行う。

この ACK の受信により送信元においては、輻輳

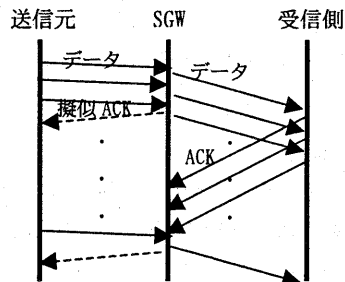


図3 データ通信時動作

制御アルゴリズム (スロースタートと輻輳回避) を用い送信量を増やしつつ、連続的にデータ送信を行う。通常、TCP においては、ネットワークの輻輳状況を確認しながらスループットを上げることを目的とし、RTT (Round Trip Time) や再送タイムアウト発生などを用い、ACK の受信状況に応じたスループットの制御を行っている。具体的には、輻輳制御 Window (cwnd) を用いて、与えられた

Window サイズ (sssthresh) に達するまでの間、スロースタートでは、1ACK 受信毎にセグメントサイズ (segsize) 分、輻輳回避では 1ACK 受信毎に、 $segsize * segsize / cwnd$  分、cwnd を増加させる。従って、スルーブットは、ACK 受信毎の cwnd の増加量、送信 Window 内における送信済みデータ量および確認済みデータ量に依存する。擬似 ACK はこの ACK 受信情報を Spoof することにより、スルーブットの増加を促す。

### 2.1.3 エラー発生時の動作

TCP Spoofing アルゴリズムにおいては、2 つのエラー発生状況に対応する。

第一の状況は、送信元と Spoofing 搭載 GW との間（衛星区間の手前）でエラーが発生した場合である。

この時、TCP Spoofing アルゴリズムは、受信側 TCP が作成するものと等しい ACK を作成する。即ち、通信ホスト双方が SACK オプションをサポートしている場合は、SACK を用いた確認応答を、それ以外の場合は Duplicate ACK を作成、送出する事となる。図 4 は、送信元と GW の間においてエラー

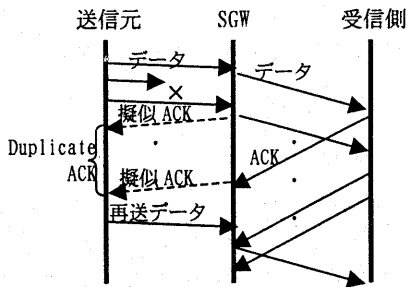


図 4 衛星区間手前でのエラー時動作

が発生し、擬似 ACK が Duplicate ACK として送信元において検知された場合のシーケンスである。

送信元 TCP においては、図 4 に示すような Duplicate ACK または再送タイムアウト (RTO) のいずれかを検知した場合、再送を行う。この時

ず、現在の Window サイズ<sup>1</sup>の 1/2 のサイズに sssthresh を設定する。タイムアウトによる輻輳の検知の場合には、さらに cwnd を 1 セグメントに設定する。即ち、SACK (Selective Acknowledgement : RFC2018) の受信によるエラー検知においては、スルーブットは減少せず<sup>2</sup>、duplicate ACK の受信をした場合はタイムアウトの発生イベントを受けた場合と比べ、スルーブットの減少の割合は通常小さい。従って、遅延をさけ Duplicate ACK の受信が可能となるため、衛星区間の手前において発生したエラーに対して、Spoofing は有効に働くと言える。

第二のエラーの状況として、Spoofing 搭載 GW と受信側ホストとの間（衛星区間以降）においてエラーが発生した場合である。図 5 は、このような

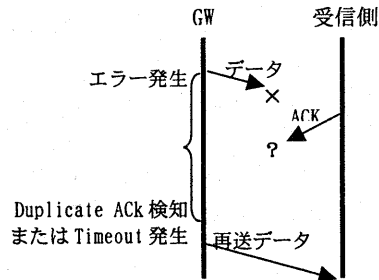


図 5 衛星区間エラー時動作

場合のシーケンス例である。この時、Spoofing アルゴリズムは、送信側 TCP と同様に再送タイムアウトあるいは Duplicate ACK の受信に基づき、エラー状況の判定を行う。一方、再送時のスルーブットは、エラー発生原因によらず、回線速度に応じた速度を用いる。この再送機構により、Spoofing による TCP の信頼性の補完を行う。

<sup>1</sup>  $\text{Max}(\text{Min}(cwnd, \text{受信側から告知されたウィンドウ}), 2 \text{セグメント})$

<sup>2</sup> 重複しない SACK



図に示すように受信側において、データ4を受信した際に、「次に受信すべきデータは5である、ただし、データ2および3の受信には失敗した」ことを示す SACK5 [2, 3 失敗] という確認応答を送る。

また図7は、受信側のみ SACK サポートしている環境において、送信元とSGW間あるいはS-GWにてエラーが発生し、データ2およびデータ3の受信に失敗した場合の例である。

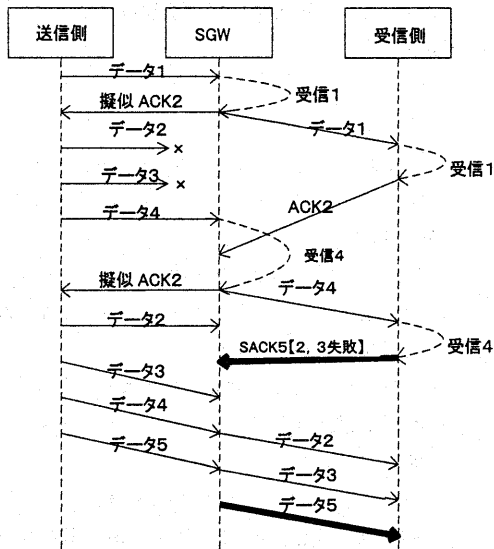


図 7

送信側は SACK をサポートしていないため、SGW は送信側に対しては SACK を用いず、データ4受信し受信側に転送すると共に、擬似 ACK 2 を返し、送信側では、擬似 ACK 2 の Duplicate ACK の受信あるいは再送タイムアウト発生を気にデータ2の再送を行う。

一方、受信側では SACK をサポートしているため、データ1受信後、ACK 2 を送り、その後、データ2および3の受信なくデータ4を受信し、データ2および3の受信失敗の SACK オプション情報付き ACK (SACK 5 [2, 3 失敗] にて表す) を返送する。この SACK 5 [2, 3 失敗] を受けた SGW は、送信

側からのデータ2, 3 および5の受信の後、SACKに基づき、受信側に対し、これらのみ(即ちデータ4の二重転送はしない)を転送する。

以上のことから、受信側のみ SACK サポートしている場合において、この SACK オプションへの Spoof を行う事により、受信側に対し SGW が SACK を用いた再送を行えるため、受信側-SGW 間の輻輳を緩和する事が可能となる。さらにまた、受信側の受信能力やバッファ容量などが小さい場合は、再送による受信負荷をの軽減や、バッファオーバーフローの回避などの効果も期待できる。

### 3.3.2 送信側のみ SACK 環境における効果

次に、送信側においてのみ SACK をサポートしている環境における効果について示す。図8は、SGWにおいて、データ1受信後、データ2およびデータ3の受信に失敗した場合の例である。図に示す

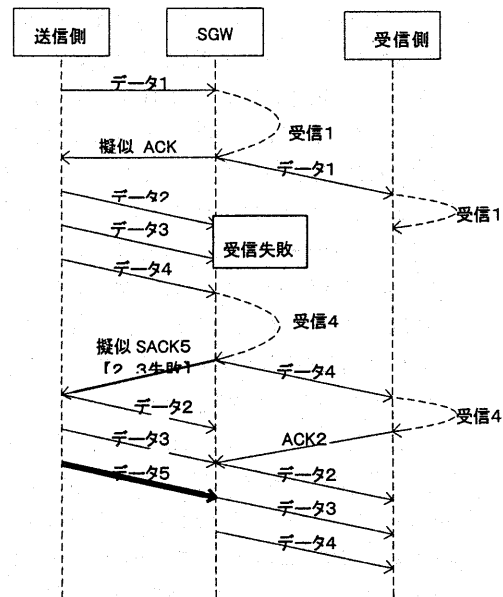


図 8

ように SGW において、データ4を受信後、「次に受信すべきデータは5である、ただし、データ2および3の受信には失敗した」ことを示す擬似 SACK5 [2, 3 失敗] という確認応答を送る。これを受けた、SACK オプションへの Spoof をサポートして

いる送信元は、データ2とデータ3の再送およびデータ5の送信を行う。

以上のことから、送信側のみ SACK サポートしている場合において、SACK オプションへの Spoof を導入し、送信側に対し SGW が SACK を用いた擬似 ACK を返送することにより、送信側が SACK を用いた再送を行えるため、送信側のスループット向上および送信側-SGW 間の輻輳を緩和する事が可能となる。さらにまた、SGW の再送による受信負荷をの軽減や、バッファオーバーフローの回避などの効果も期待できる。

#### 4 まとめと今後の課題

我々は、伝送遅延の大きな回線における TCP/IP 通信の速度性能劣化の改善を目的とした TCP Spoofing アルゴリズムの研究開発を行っている。TCP Spoofing アルゴリズムは、通信系路上、遅延回線の手前に置かれたゲートウェイ装置に適用するためのもので、以下の機能を有する。

- ・ 受信先で作成する ACK と同等の「擬似 ACK」をデータの送信元へ、受信先の ACK より先送りする事 TCP 上のアプリケーションに対するスループット低下の改善を促す
- ・ 受信先へのデータを転送機能、転送データの一時保存機能、受信先からの ACK による受信確認機能およびこれらを用いた再送機能を有し、送達確認応答の代理送信機能により損なわれた信頼性の保障を行う

TCP Spoofing アルゴリズムは速度性能の劣化に対し、非常に有効である一方で、衛星区間における輻輳制御を緩和しているため、衛星区間における輻輳に対する課題があった。本稿においてはこの課題に対し、輻輳制御の強化を行う「SACK オプションへの Spoof 機能」導入による改良提案を行った。この改良により、受信側のみ SACK サポートしている場合において、受信側に対しゲートウェイが SACK を用いた再送を行えるため、受信側-ゲートウェイ間の輻輳を緩和する事が可能となる。また、送信側のみ SACK サポートしてい

る場合において、SACK オプションへの Spoof を導入し、送信側に対しゲートウェイが SACK を用いた擬似 ACK を返送することにより、送信側が SACK を用いた再送を行えるため、送信側のスループット向上および送信側-ゲートウェイ間の輻輳を緩和する事が可能となる。さらにまた、送信側およびゲートウェイの再送による受信負荷をの軽減や、受信側およびゲートウェイのバッファオーバーフロー回避などの効果も期待できる。今後、S/W シミュレータを用いた性能検証を行ってゆく予定である。

#### Reference

- [1] 秋山他：衛星利用共有型情報配信システムの実装と評価、情処学会マルチメディア通信と分散処理研究会 No. 91, 1999
- [2] 柳生他：衛星利用共有型情報配信システムの設計と実装、情処学会マルチメディア通信と分散処理研究会 No. 96, 2000
- [3] 柳生他：高伝送遅延回線における速度性能劣化の改善、電子情報通信学会 2001年 総合大会、B-8-14
- [4] 柳生他：伝送遅延の大きな回線における TCP を用いた通信に関する一考察、情処学会第61回全国大会 2H-02, 2000
- [5] Y. Zhang. "Satellite Communications in the Global Internet: Issues, Pitfalls and Potential". Proceeding of INET' 97
- [6] T. R. Henderson. "Transport Protocol for Internet-Compatible Satellite Networks". IEEE Journal of Selected Areas of Communication, vol.17, no.2, pp.345-359, February 99.
- [7] M. Mathis 他. "TCP Selective Acknowledgment Options". Request for Comments:2018, October 1996