

明 細 書

発明の名称：通信装置および通信システム

技術分野

[0001] 本発明は、通信装置および通信システムに係り、特に、端末間の通信を中継する通信装置及び通信システムに関する。

背景技術

[0002] グローバル拠点間の通信網として、IP-VPN (Internet Protocol-Virtual Private Network) 技術等を用いたWAN (Wide Area Network) を用いることが、一般的になっている。

ある拠点にある端末が、別の海外の拠点にある端末と通信する場合は、自拠点LAN (Local Area Network) と国内WANを接続する回線と、国内WANと海外WANを接続する回線と、海外WANと別拠点LANを接続する回線と、を経由して通信が行われる。これらの回線は、契約帯域によって、使用可能な帯域幅が制限されている。

[0003] 端末間の通信では、TCP (Transport Communication Protocol) を用いるのが一般的である。TCP通信では、送信端末が送ったデータに対して、受信端末が受信済みデータ量を送信端末にフィードバック通知する。送信端末は、フィードバック通知される受信済みデータ量が増加しなくなると、廃棄検出と判定する。

さらに、送信端末は、ウィンドウサイズ (受信したことを受信端末から通知されなくても送信可能なデータサイズ) と呼ばれるパラメータを管理しており、RTT (Round Trip Time) や廃棄検出の有無に応じて、ウィンドウサイズを変化させる。

[0004] 送信端末は、RTT増加時や廃棄検出時にネットワークが混雑していると判定して、ウィンドウサイズを減少させることで、送信帯域を間接的に減少させて、ネットワークの混雑を回避する。また、RTT減少時や廃棄が無い

時にネットワークが空いていると判定して、ウィンドウサイズを増加させることで、送信帯域を間接的に増加させて、ネットワークの回線帯域を有効利用する。以上のように、TCPを用いた通信では、送信帯域がRTTと廃棄率に大きく左右される。

TCP通信と同様の送信帯域制御技術として、非特許文献1に記載のATM (Asynchronous Transfer Mode) のABR (Available Bit Rate) や、特許文献1に記載の技術を用いた通信がある。本通信では、受信端末から送信端末へ、受信帯域や輻輳有無の情報がフィードバック通知され、送信端末は、フィードバック通知された情報に基づいて、送信帯域を制御する。

[0005] 他に、TCP通信と同様の送信帯域制御技術として、特許文献2に記載の技術がある。本技術を用いた通信では、データ本体の送信帯域、誤り符号データの送信帯域、廃棄したデータの再送帯域の総和が一定値となるように制御を行う。再送帯域が増加したときは、誤り符号データの送信帯域を減少させる。

更に、TCP通信と同様の送信帯域制御技術として、特許文献3に記載の技術がある。本技術では、帯域制限を行う通信装置における、入力前の帯域と、出力後の帯域を比較し、帯域制限によるパケット廃棄に係るパケット廃棄帯域が一定値を超過したときに、帯域制限値を変化させる。

先行技術文献

特許文献

- [0006] 特許文献1：特開2004-080070号公報
特許文献2：特開2005-064648号公報
特許文献3：特開2008-141736号公報

非特許文献

- [0007] 非特許文献1：ATM Forum、Traffic Management Specification Version 4.0、af-tm-0056.00、pp. 7-10 April 1996.

発明の概要

発明が解決しようとする課題

- [0008] TCPを用いた通信では、RTTや廃棄検出の有無に応じて、送信帯域を制御するため、送信帯域がRTTと廃棄率に大きく影響を受ける。このため、WANのようなRTTが大きく、ホップ数が大きく廃棄発生箇所が多い環境では、契約帯域を大幅に下回る送信帯域しか得られない場合があるという課題がある。
- [0009] また、非特許文献1や特許文献1に記載の技術では、受信側からフィードバック通知された受信帯域や輻輳有無の情報を用いて送信帯域を制御する。輻輳有無は通知されるが、輻輳の大きさが分からないため、どの程度、送信帯域を増減させればよいか分からない、という課題がある。
- [0010] また、特許文献2に記載の技術では、データ本体の送信帯域、誤り符号データの送信帯域、廃棄したデータの再送帯域の総和が一定値となるように制御を行う。制御帯域の総和が一定であるため、1つの回線を複数の通信が共有するケースなど、他の通信と競合して使用可能な帯域が減ったときに、再送帯域が制御帯域を超過して、データが全く送信できなくなってしまう、という課題がある。
- [0011] また、特許文献3に記載の技術は、装置の入力帯域と出力帯域の差分であるパケット廃棄帯域が一定値を超過したときに、帯域制限値を変化させることができるが、装置から受信端末までの通信経路上の輻輳状態・廃棄帯域を考慮した帯域制限をすることができない、という課題がある。
- [0012] しかし、通信装置が、TCPプロトコルに基づいて、送信端末から末尾のデータを受信して確認応答のACKパケットを返信し、その直後に故障した場合、送信端末では送信が完了している一方で、受信端末では受信が完了していない、というケースが発生し得る。このケースでは、受信端末側のデータが更新されないまま、送信端末側のアプリケーションが終了してしまい、受信端末に書き込むはずだったデータが書き込まれないまま消えてしまう、という課題が発生する。

[0013] 本発明は、以上の点に鑑み、送信端末では送信が完了している一方で、受信端末では受信が完了していない、というケースの発生を防ぐことを目的とする。また、本発明は、通信装置が、送信端末から末尾のデータを受信し、その直後に故障した場合でも、送信端末との通信の状態と、受信端末との通信の状態を一致するようにすることを目的の一つとする。

課題を解決するための手段

[0014] 上記課題を解決するために、本願発明の一態様では、受信端末側からの第一のデータパケットに対するACKパケット待ち状況に応じて、第一のデータパケットより後に送信端末から送信された第二のデータパケットに対するACKパケットを、送信端末に送信する、送信端末と受信端末とのパケット送受信を仲介する通信装置である。特に、通信装置内にACK待ちの送信中データも整列待ちの受信中データも無い状態でデータパケットを受信した場合はACK返信を行わず、通信装置内にACK待ちの送信中データ、或いは整列待ちの受信中データが有る状態でデータパケットを受信した場合は1つ前に受信したデータパケットに対するACKを返信し、受信端末側からACKを受信してACK待ちの送信中データも整列待ちの受信中データも無い状態になったタイミングで通常のACKを返信するための手段を備える。

本発明の別の態様によると、

2つの端末間に位置し、端末間のTCP通信を中継する第一の通信装置と第二の通信装置を備え、

前記第二の通信装置が、通信中に受信できなかったデータの箇所を検出すると、未受信箇所の識別情報を確認応答パケットに含めて、前記第一の通信装置に対してフィードバック通知し、

前記第一の通信装置は、フィードバック通知された確認応答パケットに含まれる未受信箇所のデータを含む再送パケットを前記第二の通信装置に送信し、

前記第二の通信装置は、前記未受信箇所のデータを含む再送パケットを受

信するまで、前記未受信箇所の識別情報を含む確認応答パケットを、前記第一の通信装置に対して、定期的にフイーロバック通知する通信システムが提供される。

また、本発明の別の態様によると、

第一のデータ通信と第二のデータ通信の2つのデータ通信を中継する通信装置において、
 第二のデータ通信においてデータ付パケットを受信した場合は、少なくともひとつ前に受信したデータ付パケットに対する確認応答、又は、最後尾データより手前のデータに対する確認応答を返し、
 第一のデータ通信において確認応答待ちの送信中データが有り、かつ、第二のデータ通信において整列待ちの受信中データが無い状態で第一のデータ通信において確認応答を受信して確認応答待ちの送信中データが無くなったタイミングで、第二のデータ通信において受信した最後尾データに対する確認応答を返信する通信装置が提供される。

[0015] 本発明の第1の解決手段によると、

第一のTCP通信と第二のTCP通信の2つのTCP通信を中継する通信装置において、
 第二のTCP通信においてデータ付パケットを受信した場合は、少なくともひとつ前に受信したデータ付パケットに対する確認応答、又は、最後尾データより手前のデータに対する確認応答を返し、
 第一のTCP通信において確認応答待ちの送信中データが有り、かつ、第二のTCP通信において整列待ちの受信中データが無い状態で第一のTCP通信において確認応答を受信して確認応答待ちの送信中データが無くなったタイミングで、第二のTCP通信において受信した最後尾データに対する確認応答を返信する通信装置が提供される。

この通信装置によると、通信装置が故障しても、送信端末では送信が完了

している一方で、受信端末では受信が完了していない、というケースの発生を防ぐ効果がある。

[0016] 本発明の第2の解決手段によると、

2つの端末間に位置し、端末間のTCP通信を3つのTCP通信に分割して中継する第一の通信装置と第二の通信装置を備え、

前記第一の通信装置がデータ送信端末との第一のTCP通信を実施し、前記第二の通信装置がデータ受信端末との第二のTCP通信を実施し、前記第一の通信装置と前記第二の通信装置の間で第三のTCP通信を実施し、各TCP通信が送信データの再送と受信データの整列を独立して実施する通信システムであって、

前記第二の通信装置が、通信中に受信できなかったデータの箇所を検出すると、未受信箇所の識別情報を確認応答パケットに含めて、前記第一の通信装置に対してフィードバック通知し、

前記第一の通信装置は、フィードバック通知された確認応答パケットに含まれる未受信箇所のデータを含む再送パケットを前記第二の通信装置に送信し、

前記第二の通信装置は、前記未受信箇所のデータを含む再送パケットを受信するまで、前記未受信箇所の識別情報を含む確認応答パケットを、前記第一の通信装置に対して、定期的にフィードバック通知する前記通信システムが提供される。

この通信システムによると、RTT毎の送信量を定めたウィンドウサイズ2605を制御することにより、帯域制御を行うTCP通信しかできない端末間でも、RTTや廃棄率に依存しない通信を実現することができる。

発明の効果

[0017] 本発明によると、通信装置が、送信端末から末尾のデータを受信し、その直後に故障した場合でも、送信端末との通信の状態と、受信端末との通信の状態が一致するため、送信端末では送信が完了している一方で、受信端末では受信が完了していない、というケースが発生しなくなる。

図面の簡単な説明

[0018] [図1]通信装置 910 / 920 を W A N と L A N の境界に設置したシステム図。

[図2]パケットのフォーマット図。

[図3]通信装置のブロック図。

[図4]通信装置内部の標準 T C P 部 1007 のブロック図。

[図5]送信バッファ、受信バッファの管理用ポインタの説明図。

[図6]送受信共用バッファの管理用ポインタの説明図。

[図7]送信バッファ、受信バッファに追加した管理用ポインタの説明図。

[図8]通信装置 100、101 が 1 パケット分ずらして A C K 返信するシーケンス図。

[図9]送信端末 103 と通信装置 100 の間で再送が発生した場合の問題点を説明するためのシーケンス図。

[図10]送信端末 103 と通信装置 100 の間で再送が発生した場合の A C K 返信方法を説明するためのシーケンス図。

[図11]送信端末 103 と通信装置 100 の間で再送が発生した場合の A C K 返信方法を説明するためのシーケンス図。

[図12]通信装置 600 が次に来るコマンドを予測して予めコマンドをサーバ側へ送信する場合の A C K 返信方法を説明するためのシーケンス図。

[図13]通信装置 600 が次に来るコマンドを予測して予めコマンドをサーバ側へ送信する場合の A C K 返信方法を説明するためのシーケンス図。

[図14]バッファ管理ポインタ更新と通信装置の送受信パケットのシーケンス図。

[図15]バッファ管理ポインタ更新と通信装置の送受信パケットのシーケンス図。

[図16]バッファ管理ポインタ更新と通信装置の送受信パケットのシーケンス図。

[図17]通信装置の送受信パケットとバッファのデータ有無のシーケンス図。

[図18] N I F O T C P モジュール 1 0 0 7 の受信履歴更新部 3 1 0 6 の処理フローチャート。

[図19] N I F O T C P モジュール 1 0 0 7 の T X パケット再送部 3 1 0 4 の処理フローチャート。

[図20] バッファ管理ポイント更新のフローチャート図。

[図21] A C K 返信のフローチャート図。

[図22] ロストセグメント更新のフローチャート図。

[図23] 受信履歴更新部 3 1 0 6 がペイロード付きデータを受信した時の処理手順を示したフローチャート図。

[図24] T X パケット再送部 3 1 0 4 が A C K パケットを受信した時の処理手順を示したフローチャート図。

[図25] 通信装置 8 0 0 が P S H フラグ付パケットを受信した場合に A C K 返信しないことを説明するためのシーケンス図。

[図26] 通信装置 9 1 0 / 9 2 0 を W A N と L A N の境界に設置して、通信装置間で独自 T C P を用いて通信するシステム図。

[図27] 帯域制御方式の説明図。

[図28] 再送制御方式の説明図。

[図29] 部分的未確認応答 N A C K パケットの再送方法の説明図。

[図30] 輻輳制御方式の説明図。

[図31] 通信装置内部の独自 T C P 部 1 0 0 8 のブロック図。

[図32] T X 経路・シェーパテーブルのフォーマット図。

[図33] 送信帯域制御部 3 2 0 6 が制御帯域を更新する際の概念的なフローチャート。

[図34] シェーパ毎の送信・再送帯域テーブルのフォーマット図。

[図35] シェーパ毎の送信・再送帯域テーブル 3 2 0 5 が保持する値の意味を説明するための概念図。

[図36] 制御帯域の更新のフローチャート図。

[図37] 実施例 4 における N I F O T C P モジュール 1 0 0 7 の受信履歴更

新部 3 1 0 6 の処理フローチャート。

[図38]実施例 4 における N I F O T C P モジュール 1 0 0 7 の T X パケット再送部 3 1 0 4 の処理フローチャート。

[図39]実施例 4 を用いた場合に、通信装置間の独自 T C P を用いた通信において、ロスセグメントが発生したときに、どのようにポインタが移動し、どのような A C K パケットが返信されるかの一例を示したシーケンス図。

[図40]実施例 4 を用いた場合に、通信装置 # 1 (3 6 0 0) が通信装置 # 2 (3 6 0 2) から N A C K を受信したときに、どのようにポインタが移動するかを示したシーケンス図。

[図41]従来のプロキシ装置 2 0 0、2 0 1 の問題点を説明するためのシーケンス図。

[図42]シェーパ毎の送信・再送帯域テーブル 3 2 0 5 が保持する値の意味を説明するための概念図である。

[図43]送信帯域制御部 3 2 0 6 が再送帯域の情報を用いて制御帯域を更新するフローチャート。

[図44]図 4 3 の具体的な制御帯域の更新のフローチャート図。

[図45]送信帯域制御部 3 2 0 6 が再送率の変化率を用いて制御帯域を更新するフローチャート。

[図46]送信帯域制御部 3 2 0 6 が再送率の変化率を用いて、制御帯域を増加あるいは減少をさせ、制御帯域を更新するフローチャート。

[図47]図 4 6 の具体的な制御帯域の更新のフローチャート図。

[図48]図 9 に示すデータ通信を行う前に、T C P コネクションを確立するためのシーケンス図。

[図49]図 9 に示すデータ通信を行う前に、T C P コネクションを確立するためのシーケンス図。

[図50]図 9 に示すデータ通信を行う前に、T C P コネクションを確立するためのシーケンス図。

[図51]図 9 に示すデータ通信を行った後に、T C P コネクションを切断する

ためのシーケンス図。

[図52]図9に示すデータ通信を行った後に、TCPコネクションを切断するためのシーケンス図。

[図53]図9に示すデータ通信を行った後に、TCPコネクションを切断するためのシーケンス図。

発明を実施するための形態

[0019] 本発明を実施するための代表的な形態は、以下のとおりである。

まず、第一のTCP通信と第二のTCP通信の2つのTCP通信を中継する通信装置が、各TCP通信に対して送信バッファと受信バッファを持ち、

(1) 第一のTCP通信の送信バッファ内にACK待ちの送信中データが無く、尚且つ、第二のTCP通信の受信バッファ内に整列待ちの受信データも無い状態で、第二のTCP通信においてデータパケットを受信した場合はACK返信を行わず、(2) 第一のTCP通信の送信バッファ内にACK待ちの送信中データが有り、或いは、第二のTCP通信の受信バッファ内に整列待ちの受信データが有る状態で、第二のTCP通信においてデータパケットを受信した場合は少なくとも1つ前に受信したデータパケットに対するACKを返信し、(3) 第一のTCP通信の送信バッファ内にACK待ちの送信中データが有り、尚且つ、第二のTCP通信の受信バッファ内に整列待ちの受信データが無い状態で、第一のTCP通信においてACKを受信してACK待ちの送信中データが無くなった状態になったタイミングで、第二のTCP通信において受信したデータパケットのうち、最後尾に位置するデータパケットに対するACKを返信するための手段を備える態様とした。この態様により、受信端末が、送信端末から送信した末尾データを受信するまで、送信端末は末尾データに対するACKを受け取ることができないため、通信装置が故障しても、送信端末では送信が完了している一方で、受信端末では受信が完了していない、というケースの発生を防ぐ効果がある。本態様の詳細は、実施例1を中心に後述する。

[0020] 別の様態として、TCPフラグとしてPSHフラグを持つデータパケット

を受信したときのみ、実施例 1 に記載の A C K 返信方法を実施する態様とした。この態様により、送信端末が末尾データ送信時に P S H フラグを付加するケースでは、実施例 1 と同等の効果が得られる。本態様の詳細は、実施例 2 を中心に後述する。

[0021] 更に、別の態様として、通信装置を端末間に 2 台設置し、端末間の T C P 通信を 3 つの T C P 通信に分割して、通信装置が中継する 2 つの T C P 通信のうち、端末と通信装置間の T C P 通信では、標準 T C P を、通信装置間の T C P 通信では特許出願 1 に記載の技術を用いる態様とした。この態様により、標準的な T C P 通信しかできない端末間でも、R T T や廃棄率に依存しない通信を実現することができる。本態様の詳細は、実施例 3 を中心に後述する。

[0022] 更に、別の態様として、通信装置を端末間に 2 台設置し、端末間の T C P 通信を 3 つの T C P 通信に分割して、通信装置が中継する 2 つの T C P 通信のうち、端末と通信装置間の T C P 通信では、R T T 毎の送信量を制御することによる標準的な T C P (図 2 7 参照) を、通信装置間の T C P 通信では、特許出願 1 (特願 2 0 0 9 - 2 1 4 0 1 5 号、P C T / J P 2 0 1 0 / 0 6 3 9 7 3) に記載の技術を用いつつ、さらに、実施例 1 に記載の A C K 返信方法を用いる態様とした。この態様により、T C P 通信しかできない端末間でも、R T T や廃棄率に依存しない通信を実現することが出来、尚且つ、通信装置のいずれか一方が故障しても、送信端末では送信が完了している一方で、受信端末では受信が完了していない、というケースの発生を防ぐ効果がある。本態様の詳細は、実施例 4 を中心に後述する。

[0023] 他の本発明の態様は、以下に述べる種々の実施例で説明される。

以下、本発明の態様を詳細に説明するべく、通信の中継を行う通信装置の構成や処理シーケンスや、当該装置がネットワークを介して接続されることにより構成されるシステム等の詳細について実施例を用いて述べる。

実施例 1

[0024] 図 1 ~ 図 2 2 を用いて、2 つの T C P 通信を中継中に故障した場合でも、

送信端末では送信が完了している一方で、受信端末では受信が完了していない、というケースを発生させない通信装置の実施例を示す。

図1には、通信装置#1(910)と通信装置#2(920)をネットワーク0(900)とネットワーク1、2(901、902)の境界に設置して、ネットワーク1(901)に接続している端末(903・904)と、ネットワーク2(902)に接続している端末(905・906)の間で、3つのTCP通信を経由して通信が行われる様子を示す。通信装置#1(910)は、ネットワークインターフェースNIF0/1(915/916)と、演算部911を備える。演算部911は、TCP通信を行うTCPモジュール913/914と、データの乗せ換えを行うProxyモジュール912を実行する。なお、本実施例では、通信装置(910、920)はひとつでもよい。また、通信装置(910、920)は、例えば、プロキシ装置でもよい。

[0025] 図2には、本実施例でやりとりされるパケットのフォーマット図を表す。パケットは、MACヘッダ2900、IPヘッダ2904、TCPヘッダ2909、TCPオプションヘッダ2916、ペイロード2927を含む。MACヘッダ2900は、宛先MACアドレスを表すDMAC2901と、送信元MACアドレスを表すSMAC2902と、MACフレームタイプを表すType2903を含む。IPヘッダ2904は、MACヘッダを除くパケット長を表すIP length2905と、プロトコル番号を表すprotocol2906と、送信元IPアドレスを表すSIP2907と、宛先IPアドレスを表すDIP2908とを含む。TCPヘッダ2909は、送信元ポート番号を表すsrc. port2910と、宛先ポート番号を表すdst. port2911と、送信シーケンス番号を表すSEQ2912と、受信シーケンス番号を表すACK2913と、TCPフラグ番号を表すflag2914と、TCPのヘッダ長を表すtcp hlen2915とを含む。TCPオプションヘッダ2916は、オプション種別を表すoption kind2917と、オプション長を表すoption leng

t h 2 9 1 8 と、どこからどこまで部分的に受信できたかを記載する l e f t _ _ e d g e _ _ 1 ~ 4 (2 9 1 9 、 2 9 2 1 、 2 9 2 3 、 2 9 2 5) 、 r i g h t _ _ e d g e _ _ 1 ~ 4 (2 9 2 0 、 2 9 2 2 、 2 9 2 4 、 2 9 2 6) を含む。

[0026] 図 4 1 には、従来のプロキシ装置の課題を説明するためのシーケンス図を表す。送信端末 1 0 3 から受信端末 1 0 4 に向けて、従来のプロキシ # 1 (2 0 0) と従来のプロキシ # 2 (2 0 1) とを経由して、4 3 8 0 バイトから構成されるファイルデータを 1 4 6 0 バイトのデータを含むパケット 3 個 (1 1 0 、 1 1 3 、 1 1 6) に分割して送信する場合 (1 0 5) にやりとりされるパケットを示している。

[0027] 図 4 1 の送信端末 1 0 3 のパケット 3 個 (1 1 0 、 1 1 3 、 1 1 6) には、送信シーケンス番号として 0 、 1 4 6 0 、 2 9 2 0 の値が記載される。プロキシ # 1 (2 0 0) は、パケット (1 1 0 、 1 1 3 、 1 1 6) を受け取ると、受信パケット記載の送信シーケンス番号にペイロード長 1 4 6 0 を加算して、その値を受信シーケンス番号 A C K にした確認応答用の A C K パケット (1 2 1 、 1 2 2 、 1 2 7) を送信端末 1 0 3 に返信する。送信端末 1 0 3 は、末尾データに対する A C K パケット (1 2 7) を受信すると、送信が完了したと判断する (2 3 0) 。その後で、プロキシ # 1 (2 0 0) に障害が発生したとする (2 3 2) 。この場合、パケット (1 1 0 、 1 1 3) に記載されていたデータについては、プロキシ # 2 に向けてパケット (1 1 1 、 1 1 4) として送信され、プロキシ # 2 経由でパケット (1 1 2 、 1 1 5) として受信端末 1 0 4 に送り届けられる。それぞれのパケットに対する A C K パケット (1 2 0 、 1 2 4 、 1 1 9 、 1 2 3) も返信される。一方、パケット (1 1 6) に記載されていたデータは、障害発生 (2 3 2) と共に消滅してしまい、プロキシ # 1 (2 0 0) からは送信されなくなる。このため、受信端末 1 0 4 は、受信が未完了のままである (2 3 1) 。すると、送信端末 1 0 3 では、送信が完了したと判断しているにもかかわらず (2 3 0) 、受信端末 1 0 4 では、受信が完了していないと判断している状態の不一致

が発生することとなる。これは、プロキシ#2(201)の障害発生でも起こりうる。

[0028] 上記の状態の不一致が発生した状態で、例えば、送信端末103でアプリケーションを使って編集したデータを受信端末104に送信完了してから、アプリケーションを終了するケースでは、受信端末104が編集データを受け取れないまま、送信端末のアプリケーションが終了してしまい、編集データが消滅してしまう、といった事態が起こる。

以下、上述の状態の不一致を防ぐための構成・処理を詳細に説明する。

[0029] (装置構成)

図3には、ハードウェアで実装した、本実施例の通信装置910のプロック図を表す。通信装置920も同様の構成である。通信装置910は、外部ネットワークとのパケットの送受信を行うネットワークインターフェースNIF0/1(1011/1012)と、TCP以外のUDPパケット等を素通しさせるためのフイルタ(1009/1010)と、TCP通信のための制御を行うNIF0/1向けTCPモジュール1007/1008と、NIF0向けTCPモジュール1007の管理するN個(Nは1以上の整数)の送信バッファ1013およびN個の受信バッファ1015と、NIF1向けTCPモジュール1008の管理するN個の送信バッファ1016およびN個の受信バッファ1014と、送受信バッファ間でデータの乗せ換えを行うプロキシモジュール1000と、N個のエントリを備えた状態テーブル1001を有する。ソフトウェアで実装する際は、状態テーブル1001と送受信バッファ(1013~1016)を記憶部に持ち、フイルタ(1009/1010)とTCPモジュール1007/1008とプロキシモジュール1000の部分をモジュールとして演算部912に対応)に持たせる。

[0030] 状態テーブル1001は、N個のエントリを持ち、各エントリは、IPアドレス、TCPのポート番号等を含むコネクション特定のための情報1002と、NIF0側のTCPのOPEN/CLOSE等の状態情報1003と

、N I F 1 側のT C PのO P E N / C L O S E等の状態情報1 0 0 4と、N I F 0 側の送受信バッファ管理用のポインタ情報1 0 0 5と、N I F 1 側の送受信バッファ管理用のポインタ情報1 0 0 6とを登録する。

[0031] 図4には、T C Pモジュール1 0 0 7 / 1 0 0 8のブロック図を表す。

標準T C Pを実現するT C Pブロック1 0 0 7は、受信処理を行うR X部（受信処理部）3 1 0 2と、送信処理を行うT X部（送信処理部）3 1 0 1を有する。R X部3 1 0 2は、受信パケットをT C P制御パケットとデータ付パケットとA C K / 部分的確認応答用S A C Kパケットに分離するパケット解析部3 1 0 8と、受信したT C P制御パケットに基づいて状態テーブル1 0 0 1内のT C P状態1 0 0 3を変更するT C P制御部3 1 0 7と、受信したデータパケットの送信シーケンス番号S E Q 2 9 1 2と受信シーケンス番号A C K 2 9 1 3とに基づいて、状態テーブル1 0 0 1内のバッファ管理ポインタ1 0 0 5を変更してA C Kパケットや部分的確認応答S A C K付A C Kパケットを返信する受信履歴更新部3 1 0 6、とを有する。受信履歴更新部3 1 0 6のバッファ管理ポインタ1 0 0 5の変更については、後述する。T X部3 1 0 1は、状態テーブル1 0 0 1内のT C P状態1 0 0 4を用いてT C P制御パケットを送信するT C P制御部3 1 0 3と、受信したA C Kパケットに基づいて状態テーブル1 0 0 1内のバッファ管理ポインタ1 0 0 5を変更し、受信した部分的確認応答S A C K付A C Kパケットを用いて送信バッファ1 0 1 5からデータを読み出してパケットを再送するT Xパケット再送部3 1 0 4と、送信バッファ1 0 1 5からデータを読み出してパケットを送信して、状態テーブル1 0 0 1内のバッファ管理ポインタ1 0 0 5を変更する送信履歴更新部3 1 0 5と、A C K / S A C Kパケット、T C P制御パケット、再送パケット、データパケットをF I F Oで集約して出力するマルチプレクサ（集約部）3 1 0 9およびバッファ3 1 1 0 ~ 3 1 1 3とを有する。なお、実施例1、2において、T C P 1 0 0 7 / 1 0 0 8は同様の構成としてもよい。

[0032] （ポインタの説明）

図5には、送受信バッファ管理用のポインタの説明図を表す。本図では、データを左から右に書き込んでいき、左から右に読み出していくものとする。

受信バッファ1013/1014は、受信したデータの先頭（受信右端、データを書き込む先頭位置）を示すポインタ `right__recv1103` と、整列済みデータと未整列データの境界を示すポインタ `left__recv1102` と、プロキシモジュール1000による読出し済データと未読出しデータの境界を示す `left__rbuf1101` と、受信できずに歯抜けになっているデータ箇所（ロスセグメント）の左端と右端を表すポインタ `left/right__loss1107-1~SEG__SIZE`（最大 `SEG__SIZE` 個）を管理する。

[0033] 受信したデータの先頭を示すポインタ `right__recv1103` は、ロスの発生が無く、前から順番にデータを受信すると、受信したデータサイズ分増えて、右へ移動する。受信できずに歯抜けになっているデータ箇所（ロスセグメント）が存在する状態で、再送パケットを受信して、ロスセグメントが消滅すると、整列済みデータと未整列データの境界を示すポインタ `left__recv1102` は、最も小さなロスセグメントの左端まで移動する。プロキシモジュールは、読出し済データと未読出しデータの境界を示す `left__rbuf1101` から順番にデータを読み出し、読み出したデータサイズ分、`left__rbuf1101` を右へ移動させる。読み出せるデータサイズの最大値は、`left__recv1102` と `left__rbuf1101` の差である。

[0034] 送信バッファ1015/1016は、プロキシモジュール1000が書き込み済みで送信可能状態となっているデータの先頭（送信右端）を表すポインタ `right__sbuf1106` と、送信済みデータの先頭を示すポインタ `right__send1105` と、受信側から確認応答を受信済みのデータの先頭を示すポインタ `left__send1104` と、再送中のデータ箇所の左端と右端を表すポインタ `left/right__rts1108-1`

～SEG__SIZE（最大SEG__SIZE個）を管理する。

[0035] プロキシモジュールが書き込み済みで送信可能状態となっているデータの先頭を表すポインタ `right__sbu f 1 1 0 6` は、プロキシモジュールがデータを書き込む毎に、書き込んだデータサイズ分増加し、右へ移動する。送信したデータの先頭を示すポインタ `right__s e n d 1 1 0 5` を始点として、新たなデータが送信され、`right__s e n d 1 1 0 5` は、送信されたデータサイズ分増加し、右へ移動する。受信側から、`l e f t__s e n d 1 1 0 4` よりも大きな受信シーケンス番号を持つ確認応答パケットを受信すると、`l e f t__s e n d 1 1 0 4` は、確認応答パケットに記載された受信シーケンス番号へと増加し、右へ移動する。同一の受信シーケンス番号を持つ確認応答パケットを重複して受信するなど、再送が発生するケースでは、再送すべき箇所が `l e f t / r i g h t__r t s 1 1 0 8` に記載され、実際に再送されると `l e f t / r i g h t__r t s 1 1 0 8` に 0 が代入され、再送済みとなる。

[0036] 図 6 には、N I F O 向け T C P モジュール 1 0 0 7 の送信バッファ 1 0 1 5 と、N I F 1 向け T C P モジュール 1 0 0 8 の受信バッファ 1 0 1 4 を共有するケース、或いは、N I F 1 向け T C P の送信バッファ 1 0 1 6 と、N I F O 向け T C P の受信バッファ 1 0 1 3 を共有するケースを示す。送信バッファ 1 0 1 6 と受信バッファ 1 0 1 3 間で、データを共有するために、プロキシモジュール 1 0 0 0 は、受信したデータを受信バッファ領域から送信バッファ領域に移動する場合である。送信バッファと受信バッファの 2 つのバッファを、1 つの送受信共用バッファとし、送信バッファ用のポインタ `right__sbu f 1 1 0 6` と、受信バッファ用のポインタ `l e f t__r b u f 1 1 0 1` が同一の値となる。この同一の値により、送信バッファ領域か、受信バッファ領域かが区別される。

[0037] 図 7 には、状態の不一致を防ぐための後述する図 8 に記載の通信装置（1 0 0、1 0 1）の A C K 返信方法を実現するために追加で必要となるバッファ管理用ポインタの説明図を示す。図 5 に示したポインタに加えて、直近の

left__recv1102の値を記録するためのポインタprev__left__recv1900をさらに有する。なお、図6のようにバッファを構成した場合も同様である。

[0038] (基本シーケンス)

図8には、本実施例における通信装置(100、101)のACK返信方法を記載したシーケンス図を表す。

図41と同様に、送信端末103から受信端末104に向けて4380バイトから構成されるファイルデータを1460バイトのデータを含むパケット3個(110、113、116)に分割して送信する例を示している。送信端末103からの1つ目のパケット(110-112)には、送信シーケンス番号として0、パケット長として1460が記載される(0番目のバイトから1459番目のバイトが送信される)。2つ目のパケット(113-115)には、送信シーケンス番号として先頭データを表す1460、パケット長として1460が記載される(1460番目のバイトから2919番目のバイトが送信される)。同様に、3つ目のパケット(116-118)には、送信シーケンス番号として先頭データを表す2920、パケット長として1460が記載される(2920番目のバイトから4379番目のバイトが送信される)。通信装置#1(100)・通信装置#2(101)(図1の通信装置910、920に相当)は、先頭パケット(110・111)を受け取ってもACKを返信しない。通信装置#1(100)・通信装置#2(101)は、2つ目のパケット(113・114)を受け取ってから、1つ目のパケットに対するACKパケット(121、120)を返信する。更に、3つ目のパケット(116・117)を受け取ると、2つ目のパケットに対するACKパケット(122、124)を返信する。送信端末103は、3つ目のパケット(116)を送信しても、2つ目のパケットに対するACKパケット(122)しか受信しないので、送信が完了していないと判断したままである(128)。通信装置#2(101)から3つ目のパケット(118)が受信端末104に到達して初めて、3つ目のパケットに対

する末尾のACKパケット（125）が受信端末104から返信され、受信端末130は受信が完了したと判断する（130）。通信装置#2（101）・通信装置#1（100）は、末尾のACKパケット（125・126）を受け取ると、3つ目のパケットに対する末尾のACKパケット（126・127）を送信側へ送信する。送信端末103は、末尾のACKパケット127を受け取り、送信が完了したと判断する（129）。受信端末130で受信が完了したと判断されない限り（130）、末尾のACKパケット（125、126、127）が送信端末103に返信されず、送信端末129が送信完了と判断しないため、受信端末104が編集データを受け取れないまま、送信端末のアプリケーションが終了してしまい、編集データが消滅してしまう、といった事態を防ぐことができる。なお、上述のように、ひとつ前に受信したデータパケットに対するACKを送信する以外にも、2つ前、3つ前など少なくともひとつ前のデータパケットに対するACKを送信してもよい。また、先頭パケット110に対しては、ACKを送信しない以外にも、先頭パケットの最後尾データに対応する受信シーケンス番号を含まないような（例えば、0～1459）ACKを返送し、先頭パケットの最後尾データに対応する受信シーケンス番号を含むACKは返送しないようにしてもよい。

[0039]（再送への対処）

図9には、通信装置間の回線帯域が狭いなど、受信端末104からの末尾のACKパケット（125）の返信が遅れた場合のシーケンス図を示す。この場合、送信端末103は、3つ目のパケット（116）が途中で廃棄したと判断して、パケットの再送を行う（500）。通信装置#1（100）は、再送パケット（501）を受信すると、2つ目のパケット（113）に対するACKパケット（502）を重複して返信する。すると、送信端末103は、再びパケット廃棄が生じたと判断して、再度送信を行う（503）。通信装置#1（100）は、再送パケット（504）を受信すると、2つ目のパケットに対するACKパケット（505）を重複して返信する。すると

、送信端末１０３は、再三パケット廃棄が生じたと判断して、再度送信を行う（５０６）。通信装置＃１（１００）は、再送パケット（５０７）を受信すると、２つ目のパケットに対するＡＣＫパケット（５０８）を重複して返信する。

このように、何度も再送をした後に、３つ目のパケット（１１６）に対するＡＣＫパケット（１２７）を受け取ると、送信端末（１０３）がＴＣＰフラグ番号をＲＳＴ（reset）にセットしたパケット（５０９）を送信して、受信側では、強制的にＴＣＰコネクションを切断しなければならなくなる。

[0040] 図１０には、上記のＲＳＴパケット送信を防ぐための通信装置（１００、１０１）のＡＣＫ返信方法を記載したシーケンス図を表す。受信端末１０４からの末尾のＡＣＫパケット（１２５）の返信が遅れた場合、送信端末１０３は、３つ目のパケット（１１６）が途中で廃棄したと判断して、上述のようにパケットの再送を行う（３００）。通信装置＃１（１００）は、再送パケット（３０１）を受信すると、受信シーケンス番号を、２つ目のパケットに対するＡＣＫパケット（１２２）よりも大きく、３つ目のパケットに対するＡＣＫパケット（１２７）よりも小さく設定した、ＡＣＫパケット（３０２）を返信する。

[0041] 送信端末は、戻ってくるＡＣＫパケットの受信シーケンス番号がインクリメントしている間は、送信したパケット３０１の一部のデータが受信端末に届いており、送信端末１０３は通信回線がまだ繋がっていると判断するため、ＲＳＴを送信しなくなる。通信装置は、強制的にＴＣＰコネクション切断する制御をする必要がなくなる。

[0042] 図１１のように、送信端末１０３が、３つ目のパケット（１１６）が途中で廃棄したと判断してパケットを再送するケース（４００、４０３、４０６）が多い場合、受信シーケンス番号を１ずつ増加させてＡＣＫパケットを返信してもよい（４０２、４０５、４０８）。この場合、再送パケット（４０１、４０２、４０３）の送信シーケンス番号は１ずつ増加し、ペイロード長は１ずつ減少する。

[0043] (予測コマンドへの対処)

図12には、通信装置#1(100)と同一のACK返信を行う通信装置#1(600)が、アプリケーション内部でウィンドウサイズを持っているCIFS(Common Internet File System)等のアプリケーションを高速化させるために、次に来るコマンドを予測して送信する機能を持っている場合に、本実施例の処理により不具合が発生することを説明するためのシーケンス図を示す。クライアント(クライアント装置)603が、サーバ604に向けて、ファイルの0バイト目から0xf000(61440)バイト読み出すreadコマンドを記載したパケット610を送信したとする。通信装置#1(600)は、パケット610を中継してパケット611としてサーバ604に向けて送信すると共に、次にやって来るパケットには、ファイルの0xf000バイト目から0xf000(61440)バイト読み出すreadコマンドが記載されていると予測して、予め予測したreadコマンドを記載したパケット613をサーバに向けて送信する。クライアントの発行したコマンドを記載したパケット611と、予測したコマンドを記載したパケット613は、通信装置#2(601)経由で、パケット(612、614)としてサーバ604に届けられる。サーバ604は、コマンド記載パケットに対するACKパケット615を返信すると共に、コマンド記載の、ファイルの0バイト目から0xf000(61440)バイトのデータ(618)と、ファイルの0xf000バイト目から0xf000(61440)バイトのデータ(621)をクライアント(603)に向けて送信する。これらのデータ(618、621)は、通信装置#2(701)経由で通信装置#1(700)へデータ(619、622)として送られる。通信装置#1(700)は、クライアントの要求してきたデータ(619)を受信端末603へ送信する(620)一方で、後続データ(622)については、受信端末603への送信と末尾データに対するACKの受信が完了していないため、受け取ったデータサイズ0x1e000よりも少ないサイズを受信シーケンス番号に記載したACKパケット62

4をサーバ側に返信する。クライアント603は、コマンドで要求したデータ(623)を受け取ると、受信が完了したと判断して(629)、再びコマンドを発行できるようになる。ここで、クライアント603が、通信装置#1(600)が予測したコマンド613とは異なるコマンド(例えば、`trans`、`create`)を記載したパケット626を送信したとする。この場合、サーバ604は、送信したデータサイズよりも小さな受信シーケンス番号を持つACKパケット(625)が来て、送信が未完了だと判断している状態で(630)、次のコマンドを記載したパケット628を受信することになり、前のコマンドに対する処理が終わっていないため、次のコマンドに対する処理が行われず、アプリケーションが停止してしまう。

[0044] 図13には、前記のアプリケーションの停止を防ぐためのシーケンス図を示す。通信装置#1(700)が、予測コマンドを発行している状態では、受け取ったデータサイズ0x1e000と同一のサイズを受信シーケンス番号に記載したACKパケット724をサーバ側に返信する。自身が予測コマンドを発行している状態で、通常通りのACK返信を行うことで、サーバ604は、送信したデータサイズと同一の受信シーケンス番号を持つACKパケット(725)を受け取ることによって、送信が完了したと判断することができ(730)、次のコマンド(628)がやって来たときに、次のコマンドに対する処理を続けて実行することが可能となる。アプリケーションが、次のコマンドに対する処理を行わずに、停止することが無くなる。予測コマンドを発行している状態か否かは、装置内に予測コマンドを発行することが予め設定されてもよいし、予測コマンドを発行した際にフラグを立ててそれを参照するなど、適宜の手法で判断できる。

図8、図10、図11、図13に示す通信装置の動作シーケンスは、図1に示すシステム、図3、図4のブロックを持つ装置によって実現される。

[0045] 図14には、状態の不一致を防ぐために1パケット／1バイトずらしてACK返信を行う通信装置2000が、先頭データ2005を受信したときに、どのようにポインタが移動するかを示したシーケンス図を示す。ここで、

通信装置 2000 は、通信装置 910、920 である。通信装置 2000 が送信端末 2001 との間の TCP 通信に用いる受信バッファ 2003 のポインタ移動と、通信装置 2000 が受信端末 2002 との間の TCP 通信に用いる送信バッファ 2004 のポインタ移動を示す。

[0046] 通信装置 2000 が、送信端末 2001 から送信シーケンス番号が right_recv 1103 と等しい先頭データ付パケット 2005 を受信する

と、left_recv 1102 と right_recv 1103 を、受信したデータサイズ分、右側へ移動させる (2006)。prev_left_recv 1900 は不変である。受信シーケンス番号を left_recv

v 1102 にした確認応答用の ACK パケット 1309 を送信端末 1301 へ送る。left_recv 1102 が left_rbuf 1101 よりも

大きくなると、left_rbuf 1101 から left_recv 1102 まで書かれているデータを、right_subuf 1106 を先頭に送信

バッファ 2007 へ移動させて (2013)、left_rbuf 1101 と right_subuf 1106 を、移動させたデータサイズ分、右側へ移

動させる (2008、2009)。right_subuf 1106 が right_send 1105 よりも大きくなると、right_send 1105 を、送信したデータサ

イズ分、右側へ移動させる (2012)。上記のシーケンスにより、通信装置内に ACK 待ちの送信中データも整理

待ちの受信中データも無い状態でデータパケットを受信した場合は ACK 返信を行わない手段が実現される。

[0047] 図 15 には、この状態から、更に、通信装置 2000 が、送信端末 2001 から送信シーケンス番号が right_recv 1103 と等しい先頭データ付パケット 2101 を受信した場合のシーケンス図を示す。

[0048] 通信装置 2000 が、パケット 2101 を受信すると、prev_left_recv 1900 に left_recv 1102 の値を代入してから、

left__recv1102とright__recv1103を、受信したデータサイズ分、右側へ移動させて(2103)、受信シーケンス番号ACK2913をprev__left__recv1900にした確認応答用のACKパケット2104を送信端末2001へ送る。なお、受信したパケットに対してACKを返信する場合は、受信シーケンス番号をleft__recv1102にした確認応答用のACKパケット1318を送信端末1301へ送ることになるが、本実施例では、1つ前のパケットに対するACKが送信される。left__recv1102がleft__rbuf1101よりも大きくなると、left__rbuf1101からleft__recv1102まで書かれているデータを、right__sbuf1106を先頭に送信バッファ2102へ移動させて(2110)、left__rbuf1101とright__sbuf1106を、移動させたデータサイズ分、右側へ移動させる(2106、2105)。right__sbuf1106がright__send1105よりも大きくなると、right__send1105からright__sbuf1106までのデータを受信端末2002へ送信して(2107)、right__send1105を、送信したデータサイズ分、右側へ移動させる(2109)。なお、left__send1104は、確認応答用のACKパケットを受信端末から受信していないので移動しない。

上記のシーケンスにより、通信装置内にACK待ちの送信中データ、或いは整列待ちの受信データが有る状態でデータパケットを受信した場合は1つ前に受信したデータパケットに対するACKを返信する手段が実現される。

[0049] 図16には、状態の不一致を防ぐために1パケット／1バイトずらしでACK返信を行う通信装置2000が、末尾データ2013を受信したときに、どのようにポインタが移動するかを示したシーケンス図を示す。通信装置2000が送信端末2001との間のTCP通信に用いる受信バッファ2010のポインタ移動と、通信装置2000が受信端末2002との間のTCP

P通信に用いる送信バッファ2012のポインタ移動を示す。

[0050] 通信装置2000が末尾データを受信端末2002に送信した後、送信バッファ2012では、`right__send1105`と`right__sbu`
`f1106`が同じ値となり、`right__send1105`は`left__s`
`end1104`よりも末尾データサイズ分大きい値となっている。受信バッ
ファ2010では、`left__recv1102`と`right__recv1`
`103`が同じ値となっており、未整列の受信データが存在せず、`prev`
`__left__recv1900`が`left__recv1102`よりも小さい
値となっている。この状態で、受信シーケンス番号が`right__send`
`1105`と同じ値である末尾データに対するACKパケット2013を受信
すると、`left__send1104`が`right__send1105`と同
じ値となり（2014）、受信バッファ2010に未整列の受信データが
無い状態で、送信バッファ2012に確認応答待ちのデータが無くなる。こ
のタイミングで、通信装置2000は、受信シーケンス番号を`left__r`
`ecv1102`としたACKパケット2016を受信端末2001に対して
返信する。

上記のシーケンスにより、受信バッファに整列待ちの受信データが無い
状態で、受信端末側からACKを受信して、送信バッファに確認応答待ちの
送信中データが無くなったタイミングで通常のACKを返信する手段が実現
される。

[0051] 図17には、図8のシーケンス図に、送信側TCP通信で用いる受信バッ
ファにおける受信データの有無と、受信側TCP通信で用いる送信バッ
ファにおける送信中データの有無の情報を加えたシーケンス図を示す。

[0052] 通信装置100/101は、データ付パケット110/111を、受信
データも送信中データも無い状態1606/1609で受け取るので、先頭
データと判断してACKを返信しない。また、通信装置100/101は、
受信シーケンス番号4380のACKパケット125/126を受け取ると
、受信データも送信中データも無い状態1630/1631に変化するた

め、末尾データに対するACKパケット126／127を返信する。データパケット（113、114、116、117）を受信したときは、受信データか送信中データがある状態なので、一つ前のデータパケットに対するACKパケットを返信する。また、受信データや送信中データがある状態で、ACKパケット（119、120、123、124）を受信すると、受信済みデータに対するACKパケットを送信側へ返信しない。

[0053] 図18には、NIFOTCPモジュール1007の受信履歴更新部3106の処理フローチャートを表す。

処理がスタートすると（ステップ3801）、受信履歴更新部3106は、ペイロード長が0より大きなパケットを受信するまで待機する（ステップ3802）。パケットを受信すると、受信履歴更新部3106は、NIFOTCPの受信バッファの管理ポインタ1005を変更し（ステップ3803）、NIFOTCPの受信バッファ1013へパケットデータを記録する（ステップ3804）。更に、受信履歴更新部3106は、NIFOTCPの送信バッファ1016に送信中データが残っている、またはNIFOTCPの受信バッファに残っている受信データが受信パケットのペイロードサイズよりも大きいかなどを判定する（ステップ3805）。受信履歴更新部3106は、否と判定された場合は、ステップ3802に戻る。受信履歴更新部3106は、真と判定された場合は、受信済みデータの1パケット前、または1バイト前に対するACKを返信する（ステップ3806）。ステップ3805により、図14に示した、最初に到着したデータに対してACKを返信しないシーケンス処理が実現される。また、ステップ3805に続くステップ3806により、図15に示した、受信済みデータの1パケット前、または1バイト前に対するACK返信のシーケンス処理が実現される。

[0054] 図19には、NIFOTCPモジュール1007のTXパケット再送部3104の処理フローチャートを表す。

処理がスタートすると（ステップ3901）、TXパケット再送部3104は、ペイロード長が0のACKパケットを受信するまで待機する（ステッ

プ3902)。ACKパケットを受信すると、TXパケット再送部3104は、NIF0の送信バッファと、NIF1の受信バッファにデータが有るか否かを判定する(ステップ3903)。無い場合は、ステップ3902へ戻る。有る場合は、TXパケット再送部3104は、受信したACKパケット記載の受信シーケンス番号ACK2913やleft__edge__1~4(2919、2921、2923、2925)やright__edge__1~4(2920、2922、2924、2926)に基づいて、NIF0の送信バッファの管理ポインタを変更する(ステップ3905)。ステップ3905では、受信したACKパケットに記載されている受信シーケンス番号ACK2913が、left__send1104よりも大きい場合、left__send1104は、受信シーケンス番号ACK2913の値に変更される。例えば、末尾ACKであれば、送信バッファ中にACK待ちのデータがなくなる。更に、ステップ3903と同様に、TXパケット再送部3104は、もう一度、NIF0の送信バッファと、NIF1の受信バッファにデータが有るか否かを判定する(ステップ3906)。

[0055] ステップ3906で、NIF1の受信バッファにデータが有る場合は、TXパケット再送部3104は、再送中のデータ箇所の左端と右端を表すポインタleft/right__rts1108-1~SEG__SIZEの値を確認し、再送パケットがあるか否かを判定する(ステップ3908)。再送パケットが無い場合は、ステップ3902へ戻る。再送パケットが有る場合は、パケットを再送して(ステップ3909)、ステップ3902へ戻る。ステップ3906において、NIF1の受信バッファにデータが無いと判定された場合は、ACKパケットを受信したNIF0とは反対側のNIF1で、受信済みデータに対するACKを送信する(ステップ3907)。NIF0の送信バッファの管理ポインタ変更処理(ステップ3905)の前後で、NIF0の送信バッファとNIF1の受信バッファに送受信中のデータが有る状態から無い状態に変化したときに、受信済みデータに対するACKを返信することで、図16に示した受信側から末尾ACKを受信するまで、末尾

ACKを返信しないシーケンス処理が実現される。

[0056] 図20には、図18の受信バッファの管理ポインタ (`prev__left__recv1900`と、`left__recv1102`と、`right__recv1103`) 変更処理 (ステップ3803) を詳細化し、ステップ3805を受信バッファ管理ポインタを用いて表したフローチャート図をあらわす。このフローチャートの各ステップは、例えば、通信装置のTCPモジュール1007、1008が実行する。

[0057] 処理がスタートすると (ステップ1701)、データ付パケットを受信するまで待機する (ステップ1702)。データ付パケットを受信すると、`len`=ペイロード2927の長さ、`left__pkt`=受信パケットの送信シーケンス番号、`right__pkt`=`left__pkt+len`、とする (ステップ1703)。その後、`right__pkt`が`right__recv1103`よりも大きいかなかを判定する (ステップ1704)。大きい場合は、更に、`right__recv1103`が`left__pkt`以上かなかを判定する (ステップ1705)。ステップ1705の判定結果が真の場合は、`right__recv1103`に`right__pkt`を代入する (ステップ1706)。更に、ロスセグメントの有無を判定し (1707)、無い場合は`prev__left__recv1900`に`left__recv1102`を代入し (ステップ2309)、`left__recv1102`に`right__pkt`を代入する (ステップ1708)。有る場合は何もしない。ステップ1705の判定結果が偽の場合は、新規ロスセグメント`left__loss[k]`と`right__loss[k]`を作成して、`left__loss[k]`に`right__recv1103`の値を、`right__loss[k]`に`left__pkt`の値を代入して (1709)、その後で`prev__left__recv1900`に`left__recv1102`を代入し (ステップ2311)、`right__recv1103`に`right__pkt`の値を代入する (1710)。ステップ1704の判定結果が偽の場合は、パケットのセグメント (`left__pkt`~`right__pkt`) と1バイト以上一致する

ロスセグメント (`left__loss[k]~right__loss[k]`) があるか否かを判定し (ステップ1711)、無い場合は何もしない。有る場合は、ロスセグメントを更新した上で (ステップ1712)、ロスセグメントの有無を判定する (ステップ1713)。ロスセグメントが無い場合は、`left__recv1102` に `right__recv1103` の値を代入する (ステップ1714)。その後に、`prev__left__recv1900` に `left__recv1102-1` の値を代入する (ステップ2319)。ロスセグメントが有る場合は、`left__recv1102` に最小のロスセグメントの左端の値を代入する (ステップ1715)。また、`prev__left__recv1900` に `left__recv1102` を代入する (ステップ2320)。`left__recv1102` と `right__recv1103` の変更が完了すると、`left__recv1102` を受信シーケンス `ACK2913` に記載して `ACK` パケットを返信する (ステップ1716)。

[0058] 更に、ステップ1708の後と、ステップ1707で偽と判定された後に、受信バッファへパケットデータを記録し (ステップ3810)、`right__sbuf1106` が `left__send1104` よりも大きい、または、`right__recv1103` と `left__rbuf1101` の差が `len` よりも大きいと判定する (ステップ2313)。ステップ2313は、受信バッファか送信バッファのいずれかに送受信中のデータが残っているか否かを判定するための処理である。ステップ2313で否と判定されたときは、`ACK` 返信をしないまま、ステップ1702へ戻る。本ステップを挿入することで、先頭データを受信したときに `ACK` 返信しない手段が実現される。

[0059] 更に、ステップ1711で否と判定された後に、`prev__left__recv1900` と `left__pkt` が等しく、尚且つ、`left__recv1102` が `right__recv1103` と等しく、尚且つ、`right__recv1103` が `right__pkt` と等しいか否かを判定する (ステッ

プ2321)。ステップ2321で、真(等しい)と判定された時は、prev__left__recv1900を1インクリメントする(ステップ2322)。更に、prev__left__recv1900がright__recv1103と等しいか否かの判定を行い(ステップ2323)、真(等しい)と判定された場合は、prev__left__recv1900にleft__recv1102-1の値を代入する(ステップ2324)。ステップ2321~2324を挿入することで、送信端末からのパケット再送が繰り返されるときに、送信端末がRSTパケットを送信して接続を強制切断する可能性が低くなる。

ステップ2313で否と判定されるケース以外、ACK返信を行ってから(ステップ2325)、ステップ1702へ戻る。

[0060] 図21には、ステップ2325の詳細なフローチャートを示す。

ステップ2325の処理がスタートすると(ステップ2401)、right__sbuf1106とleft__send1104が等しく、尚且つ、right__recv1103とleft__rbuf1101が等しいか否かを判定する(ステップ2402)。ステップ2402は、送信バッファにも受信バッファにも送受信中のデータが残っていないことを確認するための処理である。否と判定され、すなわち、送受信中のデータが残っていると判定された場合は、図12や図13のシーケンス図に示したようなサーバ側に予測コマンドを送信中か否かを判定する(ステップ2403)。否と判定された場合は、prev__left__recv1900を受信シーケンス番号としてACKを返信する(ステップ2404)。ステップ2402で真と判定され、送信バッファにも受信バッファにも送受信中のデータが残っていないと判定された場合や、ステップ2403で真と判定され、サーバ側に予測コマンドを送信中だと判定された場合は、left__recv1102を受信シーケンス番号としてACKを返信する(ステップ2405)。

[0061] 通信装置は、ステップ2402の分岐を用いることで、受信端末からの末尾データに対するACKを受信するまで、送信側に末尾データに対するAC

Kを返信しなくなるため、通信装置に障害が発生しても、図41を用いて説明したような送信側のTCP通信と受信側のTCP通信の状態の不一致が起これなくなる。また、ステップ2403の分岐を用いることで、サーバ側に予測コマンドを送信した後で予測が外れても、サーバに対して末尾データに対するACKが返信されるため、図12を用いて説明したようなサーバ側の処理が止まることが無くなる。

[0062] 図22には、ロスセグメント更新（ステップ1712）のフローチャートを示す。

処理スタート後（ステップ1801）、`left_pkt`が`left_loss[k]`以下であり、尚且つ`left_loss[k]`が`right_pkt`よりも小さいか否かを判定する（ステップ1802）。ステップ1802において、否と判定された場合は、`left_pkt`が`right_loss[k]`よりも小さく、尚且つ`right_loss[k]`が`right_pkt`以下であるか否かを判定する（ステップ1803）。ここで、否と判定された場合は、`left_loss[k]`が`left_pkt`よりも小さく、`right_pkt`が`right_loss[k]`よりも小さいか否かを判定する（ステップ1804）。ステップ1804において真と判定された場合、`right_loss[k]`に`left_pkt`を代入した上で（ステップ1805）、新規ロスセグメント`left_loss[j]`と`right_loss[j]`を作成して、`left_loss[k]`に`right_pkt`の値を、`right_loss[k]`に`right_loss[k]`の値を代入する（1806）。

[0063] ステップ1802にて真と判定された場合、`left_loss[k]`に`right_pkt`の値を代入する（ステップ1808）。ステップ1803にて真と判定された場合、`right_loss[k]`に`left_pkt`の値を代入する（ステップ1807）。ステップ1807／1808が完了すると、`left_loss[k]`が`right_loss[k]`以上であるか否かを判定する（ステップ1809）。ステップ1809にて真と判定された

場合、ロスセグメント `left__loss[k]` と `right__loss[k]` を初期化して、消去する（ステップ1810）。

[0064] 例えば、端末と通信装置の間では標準TCPを使って通信を行い受信データの整列や送信データの再送を行い、各通信装置間ではRTTや廃棄率に依存しない独自のプロトコル（独自TCP）を使って通信を行い受信データの整列や送信データの再送を行う通信装置を用いるときに、送信端末から末尾のデータを受信して確認応答のACKパケットを返信し、その直後に故障した場合、送信端末では送信が完了している一方で、受信端末では受信が完了していない、というケースの発生を防ぐことができる。

[0065] （ポインタの移動の補足説明）

以下、ポインタの移動について補足して説明する。

受信したデータサイズに応じて受信シーケンス番号を記載してACKパケットを返信する通信装置が、パケットを前から順番に受信することが出来ずに、ロスセグメントが発生したときに、どのようにポインタが移動するかを以下に示す。通信装置が送信端末との間のTCP通信に用いる受信バッファのポインタ移動と、通信装置が受信端末との間のTCP通信に用いる送信バッファのポインタ移動を示す。

[0066] 通信装置が、送信端末から送信シーケンス番号が `right__recv1103` よりも大きなデータ付パケットを受信すると、`right__recv1103` を、受信パケットの送信シーケンス番号とデータ長の和に変更して、`left__loss[0]` を `left__recv1102` に、`right__loss[0]` を受信パケットの送信シーケンス番号にしたロスセグメント記録ポインタを新たに作成する。更に、受信シーケンス番号を `left__recv1102` にして、TCPオプションヘッダ2916内の `left__edge__1` (2919) と `right__edge__1` (2920) に、`right__loss[0]` と `right__recv1103` の値を記載した部分的確認応答（Selective ACK (SACK)）用のACKパケットを送信端末へ送る。その後、更に、通信装置が、送信端末から送信シーケ

ス番号が `right__recv1103` よりも大きなデータ付パケットを受信すると、`left__loss[1]` を `right__recv1103` に、`right__loss[1]` を受信パケットの送信シーケンス番号にしたロスセグメント記録ポイントを新たに作成して、`right__recv1103` を、受信パケットの送信シーケンス番号とデータ長の和に変更する。更に、受信シーケンス番号を `left__recv1102` にして、TCP オプションヘッダ 2916 内の `left__edge__1` (2919) と `right__edge__1` (2920) に、`right__loss[0]` と `left__loss[1]` の値を記載し、TCP オプションヘッダ 2916 内の `left__edge__2` (2921) と `right__edge__2` (2922) に、`right__loss[1]` と `right__recv1103` の値を記載した部分的確認応答 SACK 付の ACK パケットを送信端末へ送る。その後、通信装置が、送信シーケンス番号が `left__recv1102` と等しく、送信シーケンス番号とデータ長の和が `right__loss[0]` と等しいデータ付再送パケットを、送信端末から受信すると、`left__recv1102` を `left__loss[1]` の所まで移動させて、ロスセグメント記録用ポイント `left__loss[0]` と `right__loss[0]` を初期化して無くして、受信シーケンス番号を `left__recv1102` にした確認応答用の ACK パケットを送信端末へ送る。`left__recv1102` が `left__rbuf1101` よりも大きくなると、`left__rbuf1101` から `left__recv1102` まで書かれているデータを、`right__sbuf1106` を先頭に送信バッファへ移動させて、`left__rbuf1101` と `right__sbuf1106` を、移動させたデータサイズ分、右側へ移動させる。

[0067] 次に、受信したデータサイズに応じて受信シーケンス番号を記載して ACK パケットを返信する通信装置が、受信端末から ACK を受信したときに、どのようにポイントが移動するかを示す。通信装置が送信端末との間の TCP 通信に用いる受信バッファのポイント移動と、通信装置が受信端末との間

のTCP通信に用いる送信バッファのポインタ移動を示す。

[0068] 通信装置が、すでにある程度の量のデータを受信端末へ送っており、`right_sbuf1106`と`right_send1105`が等しく、`right_send1105`が`left_send1104`よりも大きいときを初期状態とする。通信装置が、受信端末から、受信シーケンス番号が`left_send1104`よりも大きく`right_send1105`よりも小さな確認応答用のACKパケットを受信すると、`left_send1104`の値がACKパケット記載の受信シーケンス番号に変更される。更に、受信シーケンス番号が`left_send1104`の値と等しく、尚且つ、受信シーケンス番号がACKパケットのそれと等しく、TCPオプションヘッダ2916内の`left_edge_1`(2919)に`left_send1104`よりも大きく`right_send1105`よりも小さな値を記載し、TCPオプションヘッダ2916内の`right_edge_1`(2920)に`right_send1105`の値を記載した部分的確認応答(Selective ACK(SACK))用の重複ACKパケットを受信すると、`left_rts[0]`を`left_send1104`に、`right_rts[0]`を`left_edge_1`(2919)にした再送セグメント記録ポインタを新たに作成する。通信装置は、再送セグメントが作成されると、`left_rts[0]`から`right_rts[0]`までのデータを再送して、再送セグメント記録ポインタを初期化して消去する。

実施例 2

[0069] 図23、図24、図25を用い、上述の実施例の図面を適宜参照して、パケットのTCPフラグにPSHフラグがセットされているか否かで末尾データか否かを判断して、末尾データに対するACKパケットが受信端末側から到着するまで、受信済みデータの1パケット前、または1バイト前に対するACKを返信する通信装置800の実施例を示す。

本実施例では、実施例1で示した図3、図4のブロック構成を持つ装置と同様の構成を用いる。図18や図19で示した受信履歴更新部3106やT

Xパケット再送部3104で行う処理に、幾つかの処理を追加することで、本実施例の通信装置800を実現する。

[0070] 図23には、受信履歴更新部3106がペイロード付きデータを受信した時の処理手順を示したフローチャート図を示す。図18と同様の処理については、同じ符号を付し、説明を省略する。

新たに、PSHデータ受信中か否かを判定するための状態変数PSH__recvを定義して、図18に示したステップ3801の後に、PSH__recvに0を代入する処理（ステップ4001）を挿入する。更に、ステップ3804の後に、受信パケットのTCPフラグにPSHフラグがセットされているか否かを判定する処理（ステップ4002）を挿入する。ステップ4002において、真と判定された場合は、PSH__recvに1を代入して（ステップ4003）、ステップ3805に移る。ステップ4002において、否と判定された場合は、NIFOにおいて、従来どおり、受信済みデータに対するACKを返信してから（ステップ4004）、ステップ3802へと戻る。本処理方式により、PSHパケットを受信したときだけ、実施例1に示したACK返信の機能が有効になる。

[0071] 図24には、TXパケット再送部3104がACKパケットを受信した時の処理手順を示したフローチャート図を示す。図19と同様の処理については、同じ符号を付し、説明を省略する。

図19に示したステップ3905の後に、PSH__recvが1であるか否かを判定する処理（ステップ4101）を追加する。ステップ4101で真（PSH__recvが1である）と判定した場合は、ステップ3906へ戻る。ステップ4101で偽と判定した場合は、ステップ3906を飛ばして、ステップ3908へと進む。更に、ステップ3907の後に、PSH__recvに0を代入する処理（ステップ4102）を追加する。本処理方式により、PSHパケットを受信したときだけ、実施例1に示した、末尾データに対するACKパケットが受信端末側から到着するまで、末尾ACKを返信しない機能が有効になる。

[0072] 図25には、装置800が、図23、図24の処理方式を用いて、末尾データか否かを、パケット記載のTCPフラグflag 2914にPSHフラグがセットされているか否かで判定を行い、PSHデータ受信中のときだけ、末尾データに対するACKパケットが受信端末側から到着するまで、末尾ACKを返信しない処理を行った場合のシーケンス図を示す。通信装置#1(800)が、送信端末803からPSHフラグ付パケット810を受け取ると、末尾のデータと判断してACKを返信せずに、そのまま通信装置#2へ送信する(811)。通信装置#2(801)は、同様に、末尾のデータと判断してACKを返信せずに、そのまま受信端末(804)へと送信する(812)。通信装置#1(800)は、再送パケット(813)を受け取っても、受信シーケンス番号に受信済みデータサイズよりも1~1459小さな値をセットしたACKを返信する。通信装置#2(801)・通信装置#1(800)は、受信シーケンス番号に受信済みデータサイズをセットした末尾のデータに対するACKパケット(816、815)を受信端末側から受け取ると、送信端末側に向けて末尾のデータに対するACKパケット(815、814)を送信する。

上記の方式により、送信端末では、送信が完了したと判断しているのにもかかわらず、受信端末では、受信が完了していないと判断している状態の不一致を防ぐことができる。

実施例 3

[0073] 図26~図36を用い、上述の実施例の図面を適宜参照して、送信端末と受信端末の間に2つの通信装置を設置し、通信装置間では上記特許出願1に記載の、又はそれをさらに改良した独自TCPによる通信を行うための実施例を示す。

[0074] 図26には、図1をベースにNIF1側のTCPモジュールを上記の独自TCPの処理を行うモジュール2501/2502に変更したシステム図を示す。本システムでは、WAN907内で独自TCPを用いて通信が行われる。

通信装置 910/920 (例えば、通信装置) は、図 3 に示したブロック図と同様なブロック図で実装することができ、NIF TCP 1008 が独自 TCP モジュール 2501/2502 に変わる。

[0075] 受信端末側の通信装置は、送信端末側の通信装置に対して廃棄箇所を全て逐一フイーロバック通知すると共に、送信端末側の通信装置は、受信端末側の通信装置からフイーロバック通知された廃棄箇所を再送し、尚且つ、基準時刻以降の再送帯域又は廃棄帯域 (再送/廃棄帯域と記すこともある) と、基準時刻以前の送信帯域とに基づいて、特定の宛先に対するデータ送信帯域とデータ再送帯域の総和を制御することで、RTT や廃棄率に依存しない通信帯域を確保する。

[0076] 図 27～図 30 には、独自 TCP の 3 つの特徴を表した説明図を示す。なお、独自 TCP は、以降、Radial (RTT And Discard Independent Congestion Control) - TCP と記載することもある。

図 27 には、左側に、標準的な TCP の例として従来型 TCP の帯域制御方式を、特許出願 1 記載の右側に独自 TCP の帯域制御方式を示す。従来型 TCP は、送信端末 2601 において、RTT 毎の送信量を定めたウインドウサイズ 2605 を制御することで (2607)、送信帯域を制御する。送信帯域は、ウインドウサイズ/RTT で表される。この方式による帯域制御は、RTT が増加すると回線が空いていても送信可能な時間 2610 が増加して、回線帯域の利用率が減少する、という課題があった。一方、独自 TCP では、一定時間毎の送信量を定めたウインドウサイズ 2606 を制御することで (2608)、送信帯域を制御する。送信帯域は、トークンサイズ/インターバル時間で表される。独自 TCP を用いることで、回線が空いていても送信可能な時間を無くすることが出来るため、回線帯域の利用率が増加する (2609)。RTT に依存しない帯域制御を実現することができる。

[0077] 図 28 には、左側に従来型 TCP の再送制御方式を、右側に独自 TCP の再送制御方式を示す。従来型 TCP では、ACK パケットの TCP オフショ

ンヘッダ2916内のleft__edge__1~4 (2919、2921、2923、2925)、right__edge__1~4 (2920、2922、2924、2926) には、どこからどこまで部分的に受信済であるかを最大4箇所記載して、部分的確認応答 (Selective ACK (SACK)) 用に用いる。一方、独自TCPでは、TCPオプションヘッダ2916内のleft__edge__1~4 (2919、2921、2923、2925)、right__edge__1~4 (2920、2922、2924、2926) には、どこからどこまで部分的に再送してほしいかを最大4箇所記載して、部分的未確認応答 (Negative ACK (NACK)) 用に用いる。

- [0078] 従来型TCPでは、送信端末2701から受信端末2702へ送った12個のデータパケットA~L (2705) のうち、B、D、F、H、Jが途中で廃棄されたとすると、TCPオプションヘッダ2916に書き込める受信済み箇所が最大4箇所までである制約から、I以降に送ったパケットに対する確認応答をこの時点では送信端末2701に送ることができない (2709)。送信端末は、A~Iまでの部分的確認応答を記載した確認応答パケット (2709) を用いて、パケットA~Iのうち廃棄されたパケットB、D、F、Hを再送する (2706)。受信端末2702は、再送パケット (2706) を受信してから、パケットI以降の部分的確認応答を記載した確認応答を返信する (2712)。送信端末2701は、パケットI以降の部分的確認応答を記載した確認応答 (2712) を受信してから、パケットI以降に廃棄されたパケットJを再送することができる (2707)。一方、独自TCPでは、送信端末2703から受信端末2704へ送った12個のデータパケットA~L (2708) のうち、B、D、F、H、Jが途中で廃棄されたとしても、A~Jまでの再送してほしい箇所をTCPオプションヘッダ2916内のleft__edge__1 (2919とright__edge__1 (2920) へ逐一書き込んでから、部分的未確認応答用 (NACK) のACKパケットを返信する (2711)。各再送要求箇所は、1つのNA

C Kパケットにだけ書き込まれる。送信端末2703は、部分的未確認応答用(NACK)のACKパケット(2711)を受信すると、TCPオプションヘッダ2916に記載された再送要求箇所B、D、F、H、Jを再送する(2710)。ロスが大量に発生したときでも、一度で再送が完了するため、通信時間が短縮し(2712)、帯域が向上する。つまり、送信端末と受信端末の間に2台の通信中継装置(通信装置)を設置し、受信端末側の通信装置が送信端末側の通信装置に対して廃棄箇所を全て逐一フィードバック通知する。例えば、送信端末側の通信装置は、受信端末側の通信装置からフィードバック通知された廃棄箇所を再送すると共に、基準時刻以降の再送帯域や廃棄帯域と、基準時刻以前の送信帯域とに基づいて、特定の宛先に対するデータ送信帯域とデータ再送帯域の総和を増減させる。それにより、廃棄率に依存しない通信を実現することができる。

[0079] なお、図29に示すように、部分的未確認応答用(NACK)のACKパケットによって再送されたパケットが廃棄されるケースもある。送信端末3000からパケットA~Dが送信され(3002)、パケットBが廃棄され、パケットBをTCPオプションヘッダ2916内のleft__edge__1(2919とright__edge__1(2920)に記載した部分的未確認応答用(NACK)のACKパケット(3003)によって再送されたパケットB(3004)が再び廃棄されたとする。この場合、受信端末3001は、例えば2RTT待っても再送パケットB(3004)が到達しなければ、再び、パケットBの再送を要求する部分的未確認応答用(NACK)のACKパケット(3005)を送信する(3006)。本再送方式は、受信バッファ管理ポイントのleft/right__loss1107と関連付けて、図7に示すACK返信時刻acked__time1901を記録することで実現する。ACK返信時刻acked__time1901と現在時刻との差分が2RTTを超過したときに、超過した廃棄箇所left/right__loss1107の再送を要求する部分的未確認応答用(NACK)のACKパケットを再送する。

[0080] 図30には、左側に従来型TCPの輻輳制御方式を、右側に独自TCPの輻輳制御方式を示す。従来型TCPは、パケット廃棄が1回起きただけでも、制御帯域を大幅に減少させる(2801)。通信回線では、回線利用率が100%に到達していなくても、バッファ等において待ち行列理論から導かれる一定の確率で廃棄が生じる(2806)。このため、従来型TCPは、回線利用率が100%に到達する前に、帯域が減少してしまい(2804)、回線帯域を100%使い切ることが出来ない場合があった。一方、独自TCPは、廃棄/再送率が一定に推移していれば、待ち行列理論から導かれる一定の確率で廃棄が生じているに過ぎないと判断して帯域を増加させていく。廃棄/再送率が増加し始めたら、例えば回線帯域の使いすぎが発生したと判断して、直近の廃棄/再送帯域と、それよりも1RTT前の制御帯域とに基づいて、1RTT前の制御帯域よりも小さくなるように制御帯域を減少させる。制御帯域が回線帯域付近で上下するように制御するため、回線帯域を100%近く使い切ることができる。

なお、ここで、送信帯域とは、シェーパ3209へパケットを振り分ける振分部3208において、シェーパ毎に観測している入力帯域を指す。また、制御帯域とは、シェーパ3209からの実際の出力帯域を指す。

[0081] 図31には、NIF1側のTCPモジュール2501/2502ブロックに実装する独自TCPを実現するためのブロック図を示す。NIF0側のTCPモジュール1007ブロックに実装する標準TCPを実現するためのブロック図は、図4と同様のものを用いることができる。

まず、標準TCPを実現するTCPブロック1007(913/924)について説明すると、図4に示すように受信処理を行うRX部3102と、送信処理を行うTX部3101を有する。RX部3102は、受信パケットをTCP制御パケットとデータ付パケットとACK/部分的確認応答用SACKパケットに分離するパケット解析部3108と、受信したTCP制御パケットに基づいて状態テーブル1001内のTCP状態1003を変更するTCP制御部3107と、受信したデータパケットの送信シーケンス番号S

EQ2912と受信シーケンス番号ACK2913とに基づいて、状態テーブル1001内のバッファ管理ポインタ1005を変更してACKパケットや部分的確認応答SACK付ACKパケットを返信する受信履歴更新部3106、とを有する。受信履歴更新部3106のバッファ管理ポインタ1005の変更では、実施例1に記載の方法を用いる。

[0082] TX部3101は、状態テーブル1001内のTCP状態1004を用いてTCP制御パケットを送信するTCP制御部3103と、受信したACKパケットに基づいて状態テーブル1001内のバッファ管理ポインタ1005を変更し、受信した部分的確認応答SACK付ACKパケットを用いて送信バッファ1015からデータを読み出してパケットを再送するTXパケット再送部3104と、送信バッファ1015からデータを読み出してパケットを送信して、状態テーブル1001内のバッファ管理ポインタ1005を変更する送信履歴更新部3105と、ACK/SACKパケット、TCP制御パケット、再送パケット、データパケットをFIFOで集約して出力するマルチプレクサ3109およびバッファ3110～3113とを有する。

[0083] 独自TCPを実現するTCPブロック2501/2502は、標準TCPを実現するTCPブロック1007をベースに、図31に示すように、RX部3102内の一部のブロックからの出力を変更し、TX部3101内に幾つかのブロックを追加することで実現される。まず、パケット解析部3108は、ACK/部分的確認応答用SACKパケットを、ACK・部分的未確認応答用(NACK)のACKパケット3210に変更し、TXパケット再送部3104に出力する。同様に、受信履歴更新部3106のACK/部分的確認応答用SACKパケットを、ACK・部分的未確認応答用(NACK)のACKパケット3210に変更し、バッファ3110に出力する。ACK/SACKパケットをACK/NACKパケットに変更し、変更されたパケットを送ることで、TX部3101では、廃棄箇所の即時再送を行うことができる。

[0084] TX部3101は、図4と異なり、送信パケットと再送パケットのヘッダ

情報を用いてどのシェーパに割り振るかを判定するシェーパ判定部3201と、コネクション情報毎に宛先シェーパを定めたコネクション・シェーパテーブル3202と、シェーパ判定部3201の判定結果に基づいてパケットを割り振る振り分け部3208と、シェーパA～C（3209）とを有する。この構成により、RTT非依存なトークンに基づく帯域制御が可能となる。

[0085] さらに、TX部3101は、現在時刻を出力するタイマ3203と、インターバル時間を定めたインターバル格納部3204と、シェーパ毎の送信帯域を制御する送信帯域制御部3206と、シェーパ毎に送信帯域や再送帯域の統計情報を記録するシェーパ毎の送信・再送帯域テーブル3205と、インターバル時間あたりの送信可能量を制御するトークン更新部3207とを追加する構成である。

[0086] 送信帯域制御部3206は、シェーパ毎の送信・再送帯域テーブル3205記載の制御帯域が変更されると、新たな変更済み制御帯域をトークン更新部3207へ通知する。トークン更新部3207は、トークンバケツアルゴリズムを用いて、各シェーパ3209からパケットが送信可能か否かを判定し、送信可能と判定したシェーパ3209からパケットを送信するように集約部3109へ指示する。この構成により、再送率に基づく輻輳制御が可能となる。

[0087] トークンバケツアルゴリズムでは、トークンバケツに単位時間毎に予め定められたトークンが蓄積されていき、送信パケット長に到達するとパケットの送信が許可される。更に、パケット送信と同時に、送信パケット長と同じ分のトークンが、トークンバケツから減らされる。トークン更新部3207は、送信帯域制御部3206から通知された制御帯域に基づいて、トークンバケツに加算するトークンの値を決定する。

以上の、送信帯域制御部3206の制御により、図27のトークンサイズに基づく帯域制御を実現可能となる。

[0088] 図32には、TCPコネクション毎に使用するシェーパを定めるためのコ

ネクション・シェーパテーブル 3202 のフォーマットを表す。コネクション・シェーパテーブル 3202 には、送信元 IP / サブネット、宛先 IP / サブネット、送信元ポート番号、及び、宛先ポート番号を含むコネクション情報毎に、対応するシェーパの識別情報を記載する。シェーパ判定部 3201 は受け取ったパケットの IP ヘッダ記載の IP アドレス 2907 / 2908、TCP ヘッダ記載のポート番号 2910 / 2911 と一致するコネクション情報を持つコネクション・シェーパテーブル 3202 のエントリを探し、そのエントリに記載されたシェーパにパケットが出力されるように振り分け部 3208 へ指示する。

[0089] 図 33 には、送信帯域制御部 3206 が制御帯域を更新する際の概念的なフローチャートを表す。

処理がスタートすると（ステップ 4201）、送信帯域制御部 3206 が（以下同じ）、パケット再送率（＝再送帯域 / 制御帯域）の増加率が一定値を超過したか否かを判定する（ステップ 4202）。超過した場合は、現在の再送帯域と、昔の制御帯域を用いて、制御帯域を更新する（ステップ 4203）。超過しなかった場合は、制御帯域を増加させる（ステップ 4204）。

[0090] 図 34 には、シェーパ毎の送信・再送帯域テーブル 3205 のフォーマットを表す。シェーパ毎の送信・再送帯域テーブル 3205 は、シェーパ毎に、旧基準時刻前の送信帯域・制御帯域と、基準時刻前の送信帯域・再送帯域・制御帯域、基準時刻と、基準時刻後の送信ビット積算値・再送ビット積算値・制御帯域とを記録する。送信ビット積算値・再送ビット積算値は、例えば、シェーパにパケットを振り分ける振分部 3208 が送信ビット数、再送ビット数を送信帯域制御部 3206 に通知し、送信帯域制御部 3206 がシェーパ毎の送信・再送帯域テーブル 3205 に書き込むことができる。

[0091] 図 35 には、シェーパ毎の送信・再送帯域テーブル 3205 が保持する値の意味を説明するための概念図である。上から下に向かって時間の経過を表す。図 35 の 4301 に示すように、基準時刻後の制御帯域は、現在時刻に

おける制御帯域を表す（本実施例では、`token`と表す）。基準時刻後の送信帯域は、現在時刻における送信帯域を表し（本実施例では、`snd`と表す）、基準時刻後の送信ビット積算値をインターバルで除算することで求められる。基準時刻後の再送帯域は、現在時刻における再送帯域を表し（本実施例では、`rts`と表す）、基準時刻後の再送ビット積算値をインターバルで除算することで求められる。また、図35の4302に示すように、基準時刻前の制御帯域・送信帯域・再送帯域は、基準時刻の直前まで（現在の基準時刻からインターバル時間前まで）の制御帯域・送信帯域・再送帯域を表す（本実施例では、`old_token`、`old_snd`、`old_rts`と表す）。また、図35の4303に示すように、現在使用している基準時刻の一つ前に使用していた基準時刻を旧基準時刻と呼び、旧基準時刻前の制御帯域・送信帯域・再送帯域は、旧基準時刻の直前まで（旧基準時刻からさらにインターバル時間前まで。現在の基準時刻からすると2インターバル時間前）の制御帯域・送信帯域・再送帯域を表す（本実施例では、`old_old_token`、`old_old_snd`、`old_old_rts`と表す）。図35の4302に示すように、基準時刻前の再送率`old_rts_ratio`は、 $\text{old_rts} / \text{old_old_snd}$ で求められる。また、図35の4301に示すように、基準時刻後の現在の再送率`rts_ratio`は、現在の再送帯域`rts`と、基準時刻前の送信帯域とに基づき、 $\text{rts} / \text{old_snd}$ で求められる。

[0092] 図36には、送信帯域制御部3206がシェーパ毎の送信・再送帯域テーブル3205記載の値を用いて、制御帯域を変更するフローチャート図を表す。

処理がスタートすると（ステップ3501）、送信帯域制御部3206が（以下同じ）、初めにタイマ3203の出力する現在時刻と、送信・再送帯域テーブル3205記載の基準時刻との差が、インターバル3204以上であるか否かを判定する（ステップ3502）。インターバルとして、計測したRTT等を使用してもよい。ステップ3502において真と判断された場

合、制御帯域（基準時刻後） $token$ の値を tmp に退避させておく（ステップ3503）。更に、再送ビット積算値（基準時刻後）／インターバル／送信帯域（基準時刻前）で求められる再送率（基準時刻後） rts_ratio が、再送帯域（基準時刻前）／送信帯域（旧基準時刻前）で求められる旧再送率（基準時刻前） old_rts_ratio の K 倍（ K ：予め定められた1以上の定数）よりも大きいか否かを判定する（ステップ3504）。大きい場合は、再送率が増加していると判断して、制御帯域（基準時刻後） $token$ の値を、制御帯域（基準時刻前） old_token の値よりも小さくなるように、再送帯域 rts を用いて減少させる。例えば、制御帯域（基準時刻後） $token = 制御帯域（基準時刻前）old_token - 再送帯域（基準時刻後）rts$ とする（ステップ3505）。ステップ3504で偽と判定された場合は、制御帯域（基準時刻後） $token$ を増加させる（ステップ3506）。ステップ3505・3506が終了すると、送信帯域（基準時刻前） $old_snd = 送信帯域（基準時刻後）snd$ 、再送帯域（基準時刻前） $old_rts = 再送帯域（基準時刻後）rts$ 、基準時刻＝基準時刻＋インターバル、送信ビット積算値（基準時刻後）＝0、再送ビット積算値（基準時刻後）＝0、制御帯域（旧基準時刻前） $old_old_token = 制御帯域（基準時刻前）old_token$ 、送信帯域（旧基準時刻前） $old_old_snd = 送信帯域（基準時刻前）old_snd$ 、制御帯域（基準時刻前） $old_token = tmp$ 、とし、各値を送信・再送帯域テーブル3205に記憶する（ステップ3507）。

- [0093] ステップ3502で、偽と判定された場合は、ステップ3504と同様に、再送ビット積算値（基準時刻後）／インターバル／送信帯域（基準時刻前）で求められる再送率（基準時刻後） rts_ratio が、再送帯域（基準時刻前）／送信帯域（旧基準時刻前）で求められる旧再送率（基準時刻前） old_rts_ratio の K 倍（ K ：予め定められた1以上の定数）よりも大きいか否かを判定する（ステップ3508）。大きい場合は、例え

ば再送率が増加していると判断して、ステップ3505と同様に、制御帯域（基準時刻後）`token`の値を、制御帯域（基準時刻前）`old_token`の値よりも小さくなるように、再送帯域`rts`を用いて減少させる。例えば、制御帯域（基準時刻後）`token`＝制御帯域（基準時刻前）`old_token`－再送帯域（基準時刻後）`rts`とする（ステップ3509）。

[0094] 以上の送信帯域制御部3206の制御帯域を更新する方法により、図30に示す直近の廃棄／再送帯域と、それよりも1RTT前の制御帯域とに基づいて、1RTT前の制御帯域よりも小さくなるように制御帯域を減少させることが可能となる。制御帯域が回線帯域付近で上下するように更新されるため、回線帯域を100%近く使い切ることができる。

以上記載した手段を用いることで、標準的なTCP通信しかできない端末間でも、RTTや廃棄率に依存しない通信を実現することができる。

実施例 4

[0095] 図37～図40を中心に、上述の実施例の図面を適宜参照して、通信装置を端末間に2台設置し、端末間のTCP通信を、端末と通信装置間のTCP通信、通信装置間のTCP通信、通信装置と端末間のTCP通信、という3つのTCP通信に分割して、各通信装置が中継する2つのTCP通信のうち、端末と通信装置間のTCP通信では標準TCPを、通信装置間のTCP通信では上記特許出願1に記載の又はそれをさらに改良した独自TCPを用いつつ、実施例1に記載のACK返信方法を用いる実施例を説明する。各通信装置が中継する2つのTCP通信では、送信データの再送と受信データの整列が独立して実行される。

[0096] 本実施例では、図26に記載のシステム、図3に記載のブロックを備えた通信装置（例えば通信装置）を用い、通信装置のTCPモジュールは、端末側は図4に記載のブロックを備えた標準TCPモジュールを用い、対向する通信装置側は図31に記載のブロックを備えた独自TCPモジュールを用いる（図26参照）。更に、受信履歴更新部3106において、実施例1にお

いて図22、図14～図21、図18、図19を用いて説明した手法を用いてバッファ管理ポインタ更新と、ACKパケット返信を行う。

[0097] また、図37に示すように、図18のフローチャート図が示す処理に対し、ACKを返信する処理（ステップ3806）の前に、TCPオプショナルのright/left/edge_1～4（2919～2926）に、管理ポインタright/left/loss1107が指示する廃棄箇所を記載する処理（ステップ4401）を追加することで、廃棄箇所を記載した部分的未確認応答（Negative ACK（NACK））用のACKパケットの返信が実現される。他の各ステップは、図18と同様であり、同じ符号を付して説明を省略する。

[0098] 更に、図38に示すように、図19のフローチャート図が示す処理に対し、送信バッファの管理ポインタを変更する処理（ステップ3905）の後で、TCPオプショナルのright/left/edge_1～4（2919～2926）に記載されている値を、再送信管理ポインタright/left/loss1108に記載する処理（ステップ4501）を追加することで、廃棄箇所を記載した部分的未確認応答（Negative ACK（NACK））用のACKパケットを受信したときに、廃棄箇所の即時再送が実現される。

[0099] 受信履歴更新部3106は、図37に示すように、図18のフローチャート図が示す処理に対して、ACKを返信する処理（ステップ3805）の前に、TCPオプショナルのright/left/edge_1～4（2919～2926）に、管理ポインタright/left/loss1107が指示する廃棄箇所を記載する処理（ステップ4401）を追加することで、廃棄箇所を記載した部分的未確認応答（Negative ACK（NACK））用のACKパケット返信を実現する。

[0100] 更に、受信履歴更新部3106は、廃棄箇所の管理ポインタright/left/loss1107毎に、TCPオプショナルのright/left/loss1107毎に、ACKパケットを返信した時刻を記録しておき、現在時刻とACKパケット

の返信時刻との差が一定時間以上に達するまで、管理ポインタ `right/left__loss1107` を TCP オプションフィールドに記載した ACK パケットを返信しないことで、図 29 に示した、一定時間経過しても再送パケットが到着しない場合に、再度、廃棄箇所を記載した ACK パケットを返信することが可能となる。

- [0101] 更に、TX パケット再送部 3104 は、図 38 に示すように、図 19 のフローチャート図が示す処理に対して、送信バッファの管理ポインタを変更する処理（ステップ 3905）の後で、TCP オプションフィールドの `right/left__edge__1~4`（2919~2926）に記載されている値を、再送箇所管理ポインタ `right/left__rts1108` に記載する処理（ステップ 4501）を追加し、再送箇所管理ポインタ `right/left__rts1108` に記載された範囲のデータを再送することで、廃棄箇所を記載した部分的未確認応答（Negative ACK（NACK））用の ACK パケットを受信したときに、廃棄箇所を即時再送することが可能となる。

以上の受信履歴更新部 3106 と、TX パケット再送部 3104 の制御により、図 28 の全廃棄箇所の即時再送が実現可能となる。

- [0102] 図 39 には、本実施例を用いた場合に、通信装置間の独自 TCP を用いた通信において、ロスセグメントが発生したときに、どのようにポインタが移動し、どのような ACK パケットが返信されるかの一例を示したシーケンス図を示す。通信装置 #2（3602）（上述の通信装置 920 に相当）が通信装置 #1（3601）（上述の通信装置 910 に相当）との間の独自 TCP 通信に用いる受信バッファ 3603 のポインタ移動と、通信装置 3602 が受信端末 3603 との間の標準 TCP 通信に用いる送信バッファ 3604 のポインタ移動を示す。

- [0103] 通信装置 #2（3602）が、通信装置 #1（3601）から送信シーケンス番号が `right__recv1103` よりも大きなデータ付パケット 3605 を受信すると、`right__recv1103` を、受信パケットの送

信シーケンス番号とデータ長の和に変更して、`left_loss[0]`を`left_recv1102`に、`right_loss[0]`を受信パケットの送信シーケンス番号にしたロスセグメント記録ポインタを新たに作成する（3606）。更に、受信シーケンス番号を`prev_left_recv1102`にして、TCPオプションヘッダ2916内の`left_edge_1`（2919）と`right_edge_1`（2920）に、`left_loss[0]`と`right_loss[0]`の値を記載した（3604）部分的未確認応答（Negative ACK（NACK））用のACKパケット3608を通信装置#1（3601）へ送る。その後、更に、通信装置#2（3602）が、通信装置#1（3601）から送信シーケンス番号が`right_recv1103`よりも大きなデータ付パケット3609を受信すると、`left_loss[1]`を`right_recv1103`に、`right_loss[1]`を受信パケットの送信シーケンス番号にしたロスセグメント記録ポインタを新たに作成して、`right_recv1103`を、受信パケットの送信シーケンス番号とデータ長の和に変更する（3610）。更に、受信シーケンス番号を`prev_left_recv1102`にして、TCPオプションヘッダ2916内の`left_edge_1`（2919）と`right_edge_1`（2920）に、`left_loss[1]`と`right_loss[1]`の値を記載した（3605）部分的未確認応答NACK用のACKパケット3612を通信装置#1（3601）へ送る。その後、通信装置#2（3602）が、送信シーケンス番号が`left_recv1102`と等しく、送信シーケンス番号とデータ長の和が`right_loss[0]`と等しいデータ付再送パケット3613を、通信装置#1（3601）から受信すると、`left_recv1102`を`left_loss[1]`の所まで移動させて、ロスセグメント記録用ポインタ`left_loss[0]`と`right_loss[0]`を初期化して無くして（3614）、受信シーケンス番号を`prev_left_recv1102`にした確認応答用のACKパケット3618を通信装置#1（3601）へ送る。le

`left__recv1102`が`left__rbuf1101`よりも大きくなると、`left__rbuf1101`から`left__recv1102`まで書かれているデータを、`right__sbuf1106`を先頭に送信バッファ3615へ移動させて(3616)、`left__rbuf1101`と`right__sbuf1106`を、移動させたデータサイズ分、右側へ移動させる(3617、3619)。

[0104] 図40には、本実施例を用いた場合に、通信装置#1(3600)が通信装置#2(3602)からNACKを受信したときに、どのようにポインタが移動するかを示したシーケンス図を示す。通信装置#1(3600)が送信端末3601との間のTCP通信に用いる受信バッファ3603のポインタ移動と、通信装置#1(3600)が通信装置#2(3602)との間の独自TCP通信に用いる送信バッファ3604のポインタ移動を示す。

[0105] 通信装置#1(3600)が、すでにある程度の量のデータを通信装置#2(3602)へ送っており、`right__sbuf1106`と`right__send1105`が等しく、`right__send1105`が`left__send1104`よりも大きいときを初期状態とする(3604)。通信装置#1(3600)が、通信装置#2(3602)から、受信シーケンス番号が`left__send1104`よりも大きく`right__send1105`よりも小さな確認応答用のACKパケット3605を受信すると、`left__send1104`の値がACKパケット3605記載の受信シーケンス番号ACKに変更される(3606)。更に、受信シーケンス番号が`left__send1104`の値と等しく、尚且つ、受信シーケンス番号がACKパケット3605と等しく、TCPオプションヘッダ2916内の`left__edge__1`(2919)に`left__send1104`の値を記載し、TCPオプションヘッダ2916内の`right__edge__1`(2920)に`left__send1104`よりも大きく`right__send1105`よりも小さな値を記載した(3611)部分的未確認応答NACK用の重複ACKパケット(3607)を受信すると、`left__rts[0]`を`le`

`left__edge__1` (2919) に、`right__rts[0]` を `right__edge__1` (2920) にした再送セグメント記録ポイントを新たに作成する (3608)。通信装置 #1 (3600) は、再送セグメントが作成されると、`left__rts[0]` から `right__rts[0]` までのデータを再送して (3609)、再送セグメント記録ポイントを初期化して消去する (3610)。

[0106] 上記に記載の手段を用いることで、標準的な TCP 通信しかできない端末間でも、RTT や廃棄率に依存しない通信を実現することが出来、尚且つ、通信装置のいずれか一方が故障しても、送信端末では送信が完了している一方で、受信端末では受信が完了していない、というケースを発生させなくすることが可能となる。

[変形例]

[0107] 以下、図 42 ないし図 48 を用いて図 30 ないし図 36 に関連する通信装置の送信制御部における帯域制御に関する他の変形例を説明する。以下、送信帯域制御部 3206 が、所定の期間内における現時点における再送状況と、過去の所定の期間内における送信状況、特に再送状況とに基づいて、シェーパ毎のバッファ 3209 や再送用のバッファ 3110 に保持されるパケットをネットワークに送出する量である制御帯域を増減する量を制御する例である。

[0108] 図 42 は、シェーパ毎の送信・再送帯域テーブル 3205 が保持する値の意味を説明するための図 35 の変形例である。上から下に向かって時間の経過を表す。図 42 の 4201 と 4202 に示すように、基準時刻後の制御帯域は、現在時刻における制御帯域に対応し（本実施例では、`token` と表す）。基準時刻後の送信帯域は、現在時刻における送信帯域（本実施例では、`snd` と表す）に対応する。送信帯域制御部 3206 は、基準時刻後の送信ビット積算値を現在時刻と基準時刻の差で除算することで求める場合と（4202）、基準時刻後の送信ビット積算値をインターバルで除算した値に

、基準時刻後の送信帯域 old_snd に $1 - (\text{現在時刻} - \text{基準時刻}) / \text{インターバル}$ を乗算した値を加算することで求める場合がある (4201)。

[0109] 基準時刻後の再送帯域は、現在時刻における再送帯域に対応する (本実施例では、 rts と表す)。この場合は、送信帯域制御部 3206 は、基準時刻後の再送ビット積算値を現在時刻と基準時刻の差で除算することで求める場合と (4202)、基準時刻後の再送ビット積算値をインターバルで除算した値に、基準時刻後の再送帯域 old_rts に $1 - (\text{現在時刻} - \text{基準時刻}) / \text{インターバル}$ を乗算した値を加算することで求める場合がある (4201)。また、図 42 の 4203 に示すように、基準時刻前の制御帯域・送信帯域・再送帯域は、基準時刻の直前まで (現在の基準時刻からインターバル時間前まで) の制御帯域・送信帯域・再送帯域に対応する (本実施例では、 old_token 、 old_snd 、 old_rts と表す)。

[0110] また、図 42 の 4204 に示すように、現在使用している基準時刻の一つ前に使用していた基準時刻を旧基準時刻と呼び、旧基準時刻前の制御帯域・送信帯域・再送帯域は、旧基準時刻の直前まで (旧基準時刻からさらにインターバル時間前まで。現在の基準時刻からすると 2 インターバル時間前) の制御帯域・送信帯域・再送帯域に対応する (本実施例では、 old_old_token 、 old_old_snd 、 old_old_rts と表す)。図 42 の 4203 に示すように、基準時刻前の再送率 old_rts_ratio は、 old_rts / old_old_snd で求められる。また、図 42 の 4201 または 4202 に示すように、基準時刻後の現在の再送率 rts_ratio は、現在の再送帯域 rts と、基準時刻前の送信帯域とに基づき、 rts / old_snd で求められる。現在時刻から基準時刻までを 1 区間前、基準時刻から旧基準時刻までを 2 区間前、旧基準時刻以前を 3 区画前、と区別する。インターバルには、計測した RTT を使用してもよいし、固定値を用いてもよい。

[0111] 図 43、44 等を用いて、図 33 の変形例として、送信帯域制御部 3206 が、1 区間前の再送帯域と、2 区間前以上前の帯域の使用状況と、に基づ

いて、制御帯域を増減する制御をする例を示す。

図43には、送信帯域制御部3206がシェーパ毎の送信・再送帯域テーブル3205記載の値を用いて、制御帯域を更新する例を示す。

処理がスタートすると（ステップ4301）、送信帯域制御部3206が（以下同じ）、1区間前のパケット再送帯域 rt_s が閾値を超過したか否かを判定する（ステップ4302）。超過した場合は、送信帯域制御部3206は、1区間前の再送帯域と、2区間以上前の送信/制御帯域を用いて、制御帯域を減少させる（ステップ4303）。ステップ4303の結果、通信装置から、一のシェーパに対応するWAN側へのTX部3101からのデータ流出量は、減少する。

[0112] 一方、超過しなかった場合は、ステップ4304に進み、送信帯域制御部3206は、制御帯域を増加させる制御を行う。はじめに、一定期間、再送帯域の閾値超過が無いか否かを判定する（ステップ4304）。例えば、一定期間とは、過去における2区間以上前の所定の一又は複数の区間に対応する。過去における所定の区間毎の再送帯域は、図34に示す送信再送帯域テーブル3205に、旧基準時刻前の再送帯域の欄を、区間ごとに設け、送信帯域制御部3206は、再送状況を管理する。ステップ4304では、送信帯域制御部3206は、旧基準時刻前の再送帯域の欄の所定の一以上の区間の再送帯域の情報を参照し、一定期間、再送帯域の閾値超過が無いか否かを判定する。この場合は、ステップ4302における閾値と、ステップ4306における閾値は異なってもよい。

[0113] または、ステップ4303において最後に帯域減少させた時刻として保持しておき、送信帯域制御部は、最後に帯域減少させた時刻と、現時刻とを比較して、所定の期間経過したかを判定してもよい。帯域減少させる処理（ステップ4303）は、図43の処理時点で、再送帯域が閾値を超えていた場合に行われるため、帯域減少させた時刻を送信再送帯域テーブルに保持しておくことで、ステップ4302の閾値と同じ情報を用いて、ステップ4304の判定をすることができる。

- [0114] ステップ4304の結果、YESの場合、送信帯域制御部3206は、制御帯域を指数的に増加させる（ステップ4305）。ステップ4305の結果、通信装置から、一のシェーパに対応するWAN側へのTX部3101からのデータ流出量は、増加する。一方、NOの場合は、送信帯域制御部3206は、制御帯域を線形的に増加させる（4306）。ステップ4605、あるいは4606により増加された新たな制御帯域に基づいて、各シェーパ毎のバッファ3209に保持されるパケットを、ネットワークへ送出する。
- [0115] ステップ4306による帯域の増加は、ステップ4305における帯域の増加よりも、緩やかな増加の制御となり、通信装置から一のシェーパに対応するWAN側へのTX部3101からのデータ流出量は、ステップ4305より比べ緩やかに増加する。ステップ4304～4306により、再送帯域が閾値を下回ってから一定時間経つまでは、他通信との競合が発生する可能性があるとして判断し、線形的にゆっくりと帯域を増加させる。一方、再送帯域が閾値を下回っている状態が一定時間続くと、他通信との競合が発生していないと判断し、指数的に急激に帯域を増加させる。このように、2区間以上の一定期間の再送の発生状況により、制御帯域の増加する割合を制御することにより、他の通信帯域を圧迫しないような帯域制御が可能となる。
- [0116] なお、ステップ4305とステップ4306の帯域の増加のさせ方は、指数的な増加、線形的な増加に限定されない。例えば、ステップ4302の結果、一定期間における再送帯域が閾値以下の場合に増加させる制御帯域量が、一定期間における再送帯域が閾値を超過した場合に増加させる制御帯域量の2倍や3倍としてもよい。つまり、ステップ4305と4306では、送信帯域制御部3206は、制御帯域を増加させる場合、2区間以上前の再送帯域に基づいて、制御帯域の増加量を制御し、過去の再送帯域が一定期間、閾値を下回っている場合に比べ、閾値に達した場合の制御帯域の増加量を抑える。また、送信帯域制御部3206は、一定期間における再送帯域が閾値を上回った回数によって、帯域の増加を緩やかにするか急にすることを制御し

てもよい。また、送信帯域制御部は、一定期間、帯域を減少させた回数によって、帯域の増加を緩やかにするか急にするかを制御してもよい。

図43に示す処理により、2区間以上に亘る送信／制御／再送帯域に基づく帯域制御と、他の通信帯域を圧迫しないような帯域制御の共存が可能となる。

[0117] 図44には、図43をより詳細に説明した、送信帯域制御部3206がシェーパ毎の送信・再送帯域テーブル3205記載の値を用いて、制御帯域を設定する際のフローチャート図を表す。

処理がスタートすると（ステップ4401）、送信帯域制御部3206が（以下同じ）、基準時刻後の再送帯域 rt_s が、予め定めた閾値 thr_R よりも大きいかな否かを判定する（ステップ4402）。大きい場合、送信帯域制御部3206は、基準時刻前の制御帯域 old_token よりも小さくなるように、基準時刻後の再送帯域 rt_s を用いて、新たな制御帯域（基準時刻後） $token$ を決定する（ステップ4408）。例えば、 $old_token - rt_s$ を新たな制御帯域（基準時刻後） $token$ とする（ステップ4408）。ステップ4402にて否と判定した場合、送信帯域制御部3206は、現在時刻と基準時刻との差がインターバル以上かな否かを判定する（ステップ4403）。否の場合は、ステップ4402に戻る。

[0118] ステップ4403にて真と判定した場合、送信帯域制御部3206は、制御帯域（基準時刻後） $token$ の値を tmp に退避させておく（ステップ4404）。その後、送信帯域制御部3206は、制御帯域（基準時刻後） $token$ を増加させる。一定期間、送信帯域制御部3206は、再送帯域の閾値超過が無いか否かを判定する（ステップ4405）。YESの場合は、送信帯域制御部3206は、制御帯域を指数的に増加させる（ステップ4410）。NOの場合は、送信帯域制御部3206は、制御帯域を線形的に増加させる（ステップ4409）。

[0119] その後、送信帯域制御部3206は、制御帯域（旧基準時刻前） $old_old_token =$ 制御帯域（基準時刻前） old_token 、制御帯

域（基準時刻前） $old_token = tmp$ 、とし、各値を送信・再送帯域テーブル3205に記憶する（ステップ4406）。ステップ4406またはステップ4408を実施した後、送信帯域制御部3206は、送信帯域（旧基準時刻前） $old_old_snd = 送信帯域（基準時刻前）old_snd$ 、送信帯域（基準時刻前） $old_snd = 送信帯域（基準時刻後）snd$ 、再送帯域（基準時刻前） $old_rts = 再送帯域（基準時刻後）rts$ 、基準時刻＝基準時刻＋インターバル、送信ビット積算値（基準時刻後）＝0、再送ビット積算値（基準時刻後）＝0、基準時刻＝現在時刻として、各値を送信・再送帯域テーブル3205に記憶する（ステップ4407）。ステップ4407の後、ステップ4402に戻る。ステップ4405と4409と4410により、再送帯域が閾値を下回ってから一定時間経つまでは、他通信との競合が発生する可能性があるとして判断し、線形的にゆっくりと帯域を増加させる。一方、再送帯域が閾値を下回っている状態が一定時間続くと、他通信との競合が発生していないと判断し、指数的に急激に帯域を増加させる。これにより、他の通信帯域を圧迫しないような帯域制御が可能となる。更に、図44に示す処理により、2区間以上に亘る送信／制御／再送帯域に基づく帯域制御と、他の通信帯域を圧迫しないような帯域制御の共存が可能となる。

[0120] 図45、46、47を用いて、送信帯域制御部3206が、複数の区間における再送率の変化率に基づいて、制御帯域を制御する例を説明する。

図45には、図33の変形例として、送信帯域制御部3206がシェーパ毎の送信・再送帯域テーブル3205記載の値を用いて、制御帯域を更新する処理を表す。

処理がスタートすると（ステップ4501）、送信帯域制御部3206が（以下同じ）、パケット再送率の変化率（ $rts_ratio / old_rts_ratio$ ）が閾値を超過したか否かを判定する（ステップ4502）。超過した場合は、1区間前の再送帯域と、2区間以上前の送信／制御帯域を用いて、制御帯域を減少させる（ステップ4503）。超過しなかった場

合は、制御帯域を増加させる（ステップ4504）。これにより、2区間以上に亘る送信／制御／再送帯域、および再送率の変化率に基づく、帯域制御が可能となる。

[0121] 図46には、図45の変形例として、送信帯域制御部3206がシェーパ毎の送信・再送帯域テーブル3205記載の値を用いて、制御帯域を更新する際、特に帯域を増加させる場合を中心に、フローチャート図を用いて説明する。

処理がスタートすると（ステップ4601）、ステップ4602、4603は、図45のステップ4502及び4503に対応する。ステップ4602の結果、パケットの再送率の変化率（ $rts_ratio / old_rts_ratio$ ）が閾値を超過しなかった場合は、送信帯域制御部3206は、過去の一定期間、再送率の変化率の閾値超過が無いか否かを判定する（ステップ4604）。

[0122] これは、ステップ4603で、送信帯域制御部3206は、最後に帯域減少させた時刻として保持しておき、送信帯域制御部3206は、最後に帯域減少させた時刻と、現時刻とを比較して、所定の期間経過したかを判定してもよい。帯域減少させる処理（ステップ4603）は、図46の処理時点で、再送帯域が閾値を超えていた場合に行われるため、帯域減少させた時刻を管理しておくことで、ステップ4602の閾値と同じ情報を用いて、ステップ4604の判定をすることができる。

[0123] あるいは、一定期間とは、過去における2区間以上前の所定の一又は複数の区間に対応する。過去における所定の区間毎の再送帯域は、図34に示す送信再送帯域テーブル3205に、パケット再送率の欄を過去の区間ごとに設け、送信帯域制御部3206は、パケット再送の変化率を管理してもよい。ステップ4604では、送信帯域制御部3206は、送信再送帯域管理テーブル3205を参照し、一定期間における再送の変化率が閾値を超過していないか判定する。この場合、ステップ4604の閾値とステップ4602の閾値は異なる値でもよい。

[0124] 一定期間、再送率の変化率の閾値超過が無い場合は、送信帯域制御部 3206 は、制御帯域を指数的に増加させる（ステップ 4606）。ステップ 4604 にて否と判定した場合は、送信帯域制御部 3206 は、制御帯域を線形増加させる（ステップ 4605）。ステップ 4605、及び 4606 により増加させた場合、新たな制御帯域を、テーブル 3205 に格納し、新たな制御帯域に基づいて、各シェーパ毎のバッファ 3209 に保持されるパケットを、ネットワークに送出する。

[0125] 図 43 のステップ 4305 や 4306 と同様に、ステップ 4606 による帯域の増加は、ステップ 4605 における帯域の増加よりも、緩やかな増加の制御となり、ステップ 4606 の結果、通信装置から、一のシェーパに対応する WAN 側への TX 部 3101 からのデータ流出量は、ステップ 4605 より比べ緩やかに増加する。

これにより、2 区間以上に亘る送信／制御／再送帯域、および再送率の変化率に基づく帯域制御と、他の通信帯域を圧迫しないような帯域制御を共存させることが可能となる。つまり、廃棄率の増加率が閾値を下回ってから一定時間経つまでは、他通信との競合が発生する可能性があると判断し、線形的にゆっくりと帯域を増加させる。一方、廃棄率の増加率が閾値を下回っている状態が一定時間続くと、他通信との競合が発生していないと判断し、指数的に急激に帯域を増加させる。これにより、他の通信帯域を圧迫しないような帯域制御が可能となる。

[0126] 図 43 と同様に図 46 も、2 区間前以上の一定期間の再送の発生状況により、制御帯域の増加する割合を制御することにより、他の通信帯域を圧迫しないような帯域制御が可能となる。

なお、ステップ 4605 とステップ 4606 の帯域の増加のさせ方は、指数的な増加、線形的な増加に限定されない。例えば、ステップ 4602 の結果、一定期間における再送の変化率が閾値以下の場合に増加させる制御帯域量が、一定期間における再送の変化率が閾値を超過した場合に増加させる制御帯域量の 2 倍や 3 倍としてもよい。つまり、ステップ 4305 と 4306

では、送信帯域制御部3206は、制御帯域を増加させる場合、2区間以上前の再送率の変化率に基づいて、制御帯域の増加量を制御し、過去の再送率の変化率が一定期間、閾値を下回っている場合に比べ、閾値に達した場合の制御帯域の増加量を抑えるようにしてもよい。

送信帯域制御部3206は、ステップ4604の代わりに一定期間における再送率の変化率が閾値を上回った回数によって、帯域の増加を緩やかにするか急にするかを判定してもよい。

[0127] 図47には、図46の具体的な例として、送信帯域制御部3206が制御帯域を更新する処理を示す。

処理がスタートすると（ステップ4701）、送信帯域制御部3206が（以下同じ）、初めにタイマ3203の出力する現在時刻と、送信・再送帯域テーブル3205記載の基準時刻との差が、インターバル3204以上であるか否かを判定する（ステップ4702）。インターバルには、計測したRTTを用いてもよいし、予め定めた値を用いても良い。ステップ4702において真と判断された場合、送信帯域制御部3206は、制御帯域（基準時刻後）`token`の値を`tmp`に退避させておく（ステップ4703）。更に、再送率（基準時刻後）`rts_ratio`が、旧再送率（基準時刻前）`old_rts_ratio`のK倍（K：予め定められた1以上の定数）よりも大きいか否かを判定する（ステップ4704）。大きい場合は、送信帯域制御部3206は、再送率が増加していると判断して、制御帯域（基準時刻後）`token`の値を、制御帯域（基準時刻前）`old_token`の値よりも小さくなるように、再送帯域`rts`を用いて減少させる。例えば、制御帯域（基準時刻後）`token`=制御帯域（基準時刻前）`old_token`-再送帯域（基準時刻後）`rts`とする（ステップ4705）。更に、最後に帯域減少させた時刻として`dec_time`に現在時刻を代入しておく（ステップ4706）。

[0128] ステップ4704で偽と判定された場合は、送信帯域制御部3206は、制御帯域（基準時刻後）`token`を線形的にまたは指数的に増加させる。

現在時刻と、最後に帯域減少させた時刻 dec_time の差が、予め定めた閾値 T よりも大きいか否かを判定する（ステップ4708）。 T は、 RTT と $token$ に比例する動の変数値としてもよい。ステップ4708にて真と判定した場合は、 $token$ を指数的に増加させる（ステップ4709）。たとえば、予め定めた値 E を用いて、 $token = E * token$ としてもよい（ステップ4709）。 E は、 $token$ によって定まる値としてもよい。ステップ4708にて偽と判定した場合は、 $token$ を線形的に増加させる（ステップ4710）。たとえば、予め定めた値 L を用いて、 $token = token + L * MSS / RTT$ としてもよい（ステップ4710）。 MSS は $Maximum\ Segment\ Size$ である。ステップ4706、4709、4710が終了したら、送信帯域（旧基準時刻前） old_old_snd = 送信帯域（基準時刻前） old_snd 、送信帯域（基準時刻前） old_snd = 送信帯域（基準時刻後） snd 、再送帯域（基準時刻前） old_rts = 再送帯域（基準時刻後） rts 、基準時刻 = 基準時刻 + インターバル、送信ビット積算値（基準時刻後）= 0、再送ビット積算値（基準時刻後）= 0、制御帯域（旧基準時刻前） old_old_token = 制御帯域（基準時刻前） old_token 、制御帯域（基準時刻前） old_token = tmp 、とし、各値を送信・再送帯域テーブル3205に記憶する（ステップ4707）。

- [0129] ステップ4702で、偽と判定された場合は、ステップ4704と同様に、送信帯域制御部3206は、再送率（基準時刻後） rts_ratio が、旧再送率（基準時刻前） old_rts_ratio の K 倍（ K ：予め定められた1以上の定数）よりも大きいか否かを判定する（ステップ4711）。大きい場合は、例えば再送率が増加していると判断して、ステップ3505と同様に、制御帯域（基準時刻後） $token$ の値を、制御帯域（基準時刻前） old_token の値よりも小さくなるように、再送帯域 rts を用いて減少させる。例えば、制御帯域（基準時刻後） $token$ = 制御帯域（基準時刻前） old_token - 再送帯域（基準時刻後） rts とす

る（ステップ４７１２）。ステップ４７１２が終了したら、ステップ４７０２に戻る。以上により、２区間以上に亘る送信／制御／再送帯域、および再送率の変化率に基づく帯域制御と、他の通信帯域を圧迫しないような帯域制御を共存させることが可能となる。

[0130] 図４６及び４７によると、送信帯域制御部３２０６は、第一の期間内における現時点で設定されている制御帯域に基づいて、ＴＸ部３１０１から送出されるパケットに占める再送パケットの再送率の変化率を監視する。その監視結果、所定の閾値に達した場合、送出されるパケットの量を減少させるために、制御帯域を小さくし、その制御帯域に従ってバッファ３２０９からパケットを読み出す。また、監視結果、再送率の変化率が、所定の時間経過後、閾値に達しなかった場合、送信帯域制御部３２０６は、送出されるパケットの量を増加させるために制御帯域を大きくし、その制御帯域に従って、その制御帯域に従ってバッファ３２０９からパケットを読み出す。また、制御帯域を大きくする場合、送信帯域制御部３２０６は、過去の第一の期間ごとの制御帯域の遷移に基づいて、どの程度大きくするかを制御する。例えば、制御帯域が減少された時点と現在の時刻との期間が、第一の期間よりも大きく、あらかじめ設けられた期間を経過している場合は、していない場合に比べて、増加させる量を制御する。

[0131] 図４３ないし図４７によると、送信帯域制御部３２０６は、ＲＴＴ等に対応するインターバル毎にＴＸ部３１０１から送出されるパケットの量を制限するための制御帯域を更新する。送信帯域制御部は、再送パケットの送出に応じて、インターバルである第一の期間ごとに再送状況を監視し、再送状況に基づいて、制御帯域を増減すべきかを判断する。監視対象となる再送状況は、例えば、第一の期間で発生した再送パケットの量に対応する再送帯域や、第一の期間で発生した再送パケットの量が、監視した時点を含む第一の期間よりも前の期間で発生した再送パケットの量との比較したものが対象となる。

制御帯域を減少させる場合、送信帯域制御部３２０６は、監視した時点

含む第一の期間よりも、過去の送信状況、再送状況や設定されていた制御帯域に基づいて、減少させる帯域量を決定し、制御帯域を減少させる。

[0132] 一方、制御帯域を増加させる場合、送信帯域制御部 3206 は、制御帯域を減少させた時点と現時点との間隔をみて、間隔が長い場合のときは、短い場合に比べて増加させる制御帯域の量大きくする、制御を行う。

以上の構成や処理により、通信装置は、第一の期間における再送状況と、過去の送信状況によって、きめ細やかな帯域制御を行うことができる。また、帯域の増減をきめ細やかに行うことで、他の通信帯域への圧迫を防ぐことができる。

[0133] また、以上の送信帯域制御部 3206 の制御帯域を更新する方法により、現時点の第一の期間における再送帯域と、それより以前の第一の期間、例えば、1 R T T 前の制御帯域野菜層帯域とに基づいて、1 R T T 前の制御帯域よりも小さくなるように制御帯域を減少させ、あるいは、増加させることが可能となる。制御帯域が回線帯域付近で上下するように更新されるため、回線帯域を有効に使うことができる。

[0134] 以下、図 48 ないし 53 を用いて、図 9 に関するデータ通信を行う通信装置間を中心にシーケンス図を説明する。

図 48 は、図 9 に示すデータ通信を行う前に、TCP コネクションを確立するためのシーケンス図を示す。はじめに、送信端末 103 と通信装置 #1 100 との間で、SYN 4801、SYN-ACK 4802、ACK 4803 のやり取りを行いコネクションを確立する。次に、通信装置 #1 100 と通信装置 #2 101 との間で、SYN 4804、SYN-ACK 4805、ACK 4806 のやり取りを行いコネクションを確立する。最後に、通信装置 #2 101 と受信端末 104 との間で、SYN 4807、SYN-ACK 4808、ACK 4809 のやり取りを行いコネクションを確立する。送信端末 103 は、送信端末 103 と通信装置 #1 100 との間で、コネクションが確立した後に、図 9 と同様にデータ送信を開始する。

[0135] 図 49 は、図 9 に示すデータ通信を行う前に、TCP コネクションを確立

するための別のシーケンス図を示す。送信端末103から、通信装置#1 100と通信装置#2 101を経由して、受信端末104に向けて、SYN(4901、4904、4907)が送られる。次に、受信端末104から、通信装置#2 101と通信装置#1 100を経由して、送信端末103に向けて、SYN-ACK(4908、4905、4902)が送られる。最後に、送信端末103から、通信装置#1 100と通信装置#2 101を経由して、受信端末104に向けて、ACK(4903、4906、4909)が送られる。この後、送信端末103は、図9と同様にデータ送信を開始する。

[0136] 図50は、図9に示すデータ通信を行う前に、TCPコネクションを確立するための別のシーケンス図を示す。送信端末103から、通信装置#1 100と通信装置#2 101を経由して、受信端末104に向けて、SYN(5001、5004、5007)が送られる。次に、受信端末104から、通信装置#2 101と通信装置#1 100を経由して、送信端末103に向けて、SYN-ACK(5008、5005、5002)が送られる。通信装置#2 101は、SYN-ACK5008を受け取ると、ACK5009を返信する。通信装置#1 100は、SYN-ACK5005を受け取ると、ACK5006を返信する。送信端末103は、SYN-ACK5002を受け取ると、ACK5003を返信する。その後、送信端末103は、図9と同様にデータ送信を開始する。

[0137] 図51は、図9に示すデータ通信を行った後に、TCPコネクションを切断するためのシーケンス図を示す。はじめに、送信端末103と通信装置#1 100との間で、FIN5101、FIN-ACK5102、ACK5103のやり取りを行い、コネクションを切断する。次に、通信装置#1 100と通信装置#2 101との間で、FIN5104、FIN-ACK5105、ACK5106のやり取りを行い、コネクションを確立する。最後に、通信装置#2 101と受信端末104との間で、FIN5107、FIN-ACK5108、ACK5109のやり取りを行いコネクションを

切断する。

[0138] 図52は、図9に示すデータ通信を行った後に、TCPコネクションを切断するための別のシーケンス図を示す。送信端末103から、通信装置#1 100と通信装置#2 101を経由して、受信端末104に向けて、FIN (5201、5204、5207) が送られる。次に、受信端末104から、通信装置#2 101と通信装置#1 100を経由して、送信端末103に向けて、FIN-ACK (5208、5205、5202) が送られる。最後に、送信端末103から、通信装置#1 100と通信装置#2 101を経由して、受信端末104に向けて、ACK (5203、5206、5209) が送られる。

[0139] 図53は、図9に示すデータ通信を行った後に、TCPコネクションを切断するための別のシーケンス図を示す。送信端末103から、通信装置#1 100と通信装置#2 101を経由して、受信端末104に向けて、FIN (5301、5304、5307) が送られる。次に、受信端末104から、通信装置#2 101と通信装置#1 100を経由して、送信端末103に向けて、FIN-ACK (5308、5305、5302) が送られる。通信装置#2 101は、FIN-ACK 5308を受け取ると、ACK 5309を返信する。通信装置#1 100は、FIN-ACK 5305を受け取ると、ACK 5306を返信する。送信端末103は、FIN-ACK 5302を受け取ると、ACK 5303を返信する。

[0140] 以上、図9並びに図48ないし図53により、端末間でTCPコネクションを確立後、通信装置が、送信側と受信側とのデータ通信を中継し、送信側の端末にACKを返すタイミングを制御する。データ通信に関するACKを送信側が受信するのに応じて、送信側からTCPコネクションを切断する処理が行われる、例を説明した。

産業上の利用可能性

[0141] 本発明は、例えば、端末間の通信を中継する通信装置及び通信システムに利用可能である。

符号の説明

- [0142] 103、903、904 端末
104、905、906 端末
110 データの流れ
100、910 通信装置
101、920 通信装置
900、901、902、907 ネットワーク
1000 プロキシモジュール

請求の範囲

- [請求項1] 2つの端末間に位置し、端末間のTCP通信を中継する第一の通信装置と第二の通信装置を備え、
- 前記第二の通信装置が、通信中に受信できなかったデータの箇所を検出すると、未受信箇所の識別情報を確認応答パケットに含めて、前記第一の通信装置に対してフィードバック通知し、
- 前記第一の通信装置は、フィードバック通知された確認応答パケットに含まれる未受信箇所のデータを含む再送パケットを前記第二の通信装置に送信し、
- 前記第二の通信装置は、前記未受信箇所のデータを含む再送パケットを受信するまで、前記未受信箇所の識別情報を含む確認応答パケットを、前記第一の通信装置に対して、定期的にフィードバック通知する通信システム。
- [請求項2] 前記第一の通信装置が予め定められたインターバル時間と、該インターバル時間毎に更新される基準時刻とを管理しており、現在時刻と基準時刻の差がインターバル時間を超過した場合に、基準時刻にインターバル時間を加算して該基準時刻を更新し、
- 前記第一の通信装置が、再送／廃棄帯域と、送信帯域と、再送／廃棄帯域と送信帯域の総和を制限するための制御帯域とを、基準時刻以降現在時刻までの第一時間と、基準時刻から該基準時刻よりもインターバル時間前の旧基準時刻までの第二時間と、該旧基準時刻からさらにインターバル時間前までの第三時間とに分けてそれぞれ記録するテーブルを備え、
- 基準時刻以降の前記第一時間の再送／廃棄帯域と、基準時刻以前の前記第二時間の制御帯域又は送信帯域とに基づいて、基準時刻以降の制御帯域を変化させることを特徴とする請求項1に記載の通信システム。
- [請求項3] 基準時刻から旧基準時刻の間の前記第二時間の送信帯域又は制御帯

域と、基準時刻以降現在時刻までの前記第一時間の再送／廃棄帯域との比で求められる第一再送率が、該旧基準時刻からさらにインターバル時間前までの前記第三時間の送信帯域又は制御帯域と、基準時刻から旧基準時刻の間の前記第二時間の再送／廃棄帯域との比で求められる第二再送率よりも大きいときに、

基準時刻以降の制御帯域を、基準時刻以降の再送／廃棄帯域に基づいて、基準時刻から旧基準時刻の間の前記第二時間の制御帯域よりも小さくなるように減少させることを特徴とする請求項 2 に記載の通信システム。

[請求項4] 基準時刻から旧基準時刻の間の前記第二時間の送信帯域又は制御帯域と、基準時刻以降現在時刻までの前記第一時間の再送／廃棄帯域との比で求められる第一再送率が、該旧基準時刻からさらにインターバル時間前までの前記第三時間の送信帯域又は制御帯域と、基準時刻から旧基準時刻の間の前記第二時間の再送／廃棄帯域との比で求められる第二再送率よりも小さいとき、制御帯域を減少させた時点より所定の期間経過しているか判定し、前記判定結果、所定の期間経過している場合は、基準時刻以降の制御帯域を第一の量大きくなるよう制御し、前記判定結果、所定の期間経過していない場合は、第一の量より小さい第二の量だけ大きくなるよう制御する、ことを特徴とする請求項 3 に記載の通信システム。

[請求項5] 前記第一の通信装置が、一定時間毎の送信量を定めたトークンサイズを制御することで制御帯域を制御する請求項 1 に記載の通信システム。

[請求項6] 前記第一の通信装置及び前記第二の通信装置は、プロキシ装置である請求項 1 に記載の通信システム。

[請求項7] 前記通信システムは、前記第一の通信装置がデータ送信端末との第一の TCP 通信を実施し、前記第二の通信装置がデータ受信端末との第二の TCP 通信を実施し、前記第一の通信装置と前記第二の通信装

置の間で第三の T C P 通信を実施し、各 T C P 通信が送信データの再送と受信データの整列を独立して実施し、

前記第一の通信装置は、

第一の T C P 通信においてデータ付パケットを受信した場合は、少なくともひとつ前に受信したデータ付パケットに対する確認応答、又は、最後尾データより手前のデータに対する確認応答を返信し、

第三の T C P 通信において確認応答待ちの送信中データが有り、かつ、第一の T C P 通信において整列待ちの受信中データが無い状態で第三の T C P 通信において確認応答を受信して確認応答待ちの送信中データが無くなったタイミングで、第一の T C P 通信において受信した最後尾データに対する確認応答を返信することを特徴とする請求項 1 に記載の通信システム。

[請求項8] 第三の T C P 通信において確認応答待ちの送信中データが有り、又は、第一の T C P 通信において整列待ちの受信中データが有る状態で第一の T C P 通信においてデータ付パケットを受信した場合に、前記少なくともひとつ前に受信したデータ付パケットに対する確認応答、又は、前記最後尾データより手前のデータに対する確認応答を返信する請求項 7 に記載の通信システム。

[請求項9] 第三の T C P 通信において確認応答待ちの送信中データが無く、かつ、第一の T C P 通信において整列待ちの受信中データも無い状態で、第一の T C P 通信においてデータ付パケットを受信した場合は確認応答の返信を行わない請求項 8 に記載の通信システム。

[請求項10] 前記第一の通信装置は、
第三の T C P 通信に対する送信バッファと、
第一の T C P 通信に対する受信バッファと
を備え、
第三の T C P 通信の前記送信バッファ内に確認応答待ちの送信中データが記憶され、

第一のTCP通信の前記受信バッファ内に整列待ちの受信中データが記憶され、

前記送信バッファを参照して、第三のTCP通信において確認応答待ちの送信中データが有るか否か判定し、前記受信バッファを参照して第一のTCP通信において整列待ちの受信中データが有るか否かを判定する請求項7に記載の通信システム。

[請求項11] 第一のデータ通信と第二のデータ通信の2つのデータ通信を中継する通信装置において、

第二のデータ通信においてデータ付パケットを受信した場合は、少なくともひとつ前に受信したデータ付パケットに対する確認応答、又は、最後尾データより手前のデータに対する確認応答を返信し、

第一のデータ通信において確認応答待ちの送信中データが有り、かつ、第二のデータ通信において整列待ちの受信中データが無い状態で第一のデータ通信において確認応答を受信して確認応答待ちの送信中データが無くなったタイミングで、第二のデータ通信において受信した最後尾データに対する確認応答を返信する通信装置。

[請求項12] 第一のデータ通信において確認応答待ちの送信中データが有り、又は、第二のデータ通信において整列待ちの受信中データが有る状態で第二のデータ通信においてデータ付パケットを受信した場合に、前記少なくともひとつ前に受信したデータ付パケットに対する確認応答、又は、前記最後尾データより手前のデータに対する確認応答を返信する請求項11に記載の通信装置。

[請求項13] 第一のデータ通信において確認応答待ちの送信中データが無く、かつ、第二のデータ通信において整列待ちの受信中データも無い状態で、第二のデータ通信においてデータ付パケットを受信した場合は確認応答の返信を行わない請求項12に記載の通信装置。

[請求項14] 第一のデータ通信に対する送信バッファと、
第二のデータ通信に対する受信バッファと

を備え、

第一のデータ通信の前記送信バッファ内に確認応答待ちの送信中データが記憶され、

第二のデータ通信の前記受信バッファ内に整列待ちの受信中データが記憶され、

前記送信バッファを参照して、第一のデータ通信において確認応答待ちの送信中データが有るか否か判定し、前記受信バッファを参照して第二のデータ通信において整列待ちの受信中データが有るか否かを判定する請求項 11 に記載の通信装置。

[請求項15] 第一のデータ通信において確認応答待ちの送信中データが有り、又は、第二のデータ通信において整列待ちの受信中データが有る状態で、第二のデータ通信において、前回受信したデータ付パケットと同じデータ付パケットを受信した場合は、前回確認応答を返信したデータよりも後ろで、かつ、最後尾データより手前のデータに対する確認応答を返信することを特徴とする請求項 12 に記載の通信装置。

[請求項16] 第一のデータ通信で予測コマンドを送信し、
第一のデータ通信で受信したデータ又は該データに対して加工を施したデータを、第二のデータ通信にて送信した後で、第二のデータ通信において予測コマンドに対するデータ付パケットを受信した場合は、該データ付パケットの最後尾データに対する確認応答を第二のデータ通信で返信することを特徴とする請求項 11 に記載の通信装置。

[請求項17] 第二のデータ通信において、P S Hフラグ付のデータパケットを受信した後、第一のデータ通信において、確認応答を受信して確認応答待ちの送信中データが無くなった状態になったタイミングで、第二のデータ通信において受信したデータパケットのうち、最後尾に位置するデータパケットに対する確認応答を返信することを特徴とする請求項 11 に記載の通信装置。