

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-114740  
(P2012-114740A)

(43) 公開日 平成24年6月14日(2012.6.14)

(51) Int.Cl. F I テーマコード (参考)  
H04L 12/56 (2006.01) H04L 12/56 400B 5K030

審査請求 未請求 請求項の数 7 O L (全 23 頁)

<p>(21) 出願番号 特願2010-262804 (P2010-262804)</p> <p>(22) 出願日 平成22年11月25日 (2010.11.25)</p> <p>(出願人による申告) 平成22年度、総務省「管理型自己組織化技術に基づく多様なサービスを取容する光ネットワーク制御技術の研究開発」産業技術力強化法第19条の適用を受ける特許出願</p>	<p>(71) 出願人 504133110 国立大学法人電気通信大学 東京都調布市調布ヶ丘一丁目5番地1</p> <p>(74) 代理人 100082131 弁理士 稲本 義雄</p> <p>(74) 代理人 100121131 弁理士 西川 孝</p> <p>(72) 発明者 伊賀井 清史 東京都調布市調布ヶ丘一丁目5番地1 国立大学法人電気通信大学内</p> <p>(72) 発明者 大木 英司 東京都調布市調布ヶ丘一丁目5番地1 国立大学法人電気通信大学内</p> <p>Fターム(参考) 5K030 GA04 GA14 LE16 MA04 MB06 MB09 MC02</p>
--	---

(54) 【発明の名称】 情報処理装置および方法、並びにプログラム

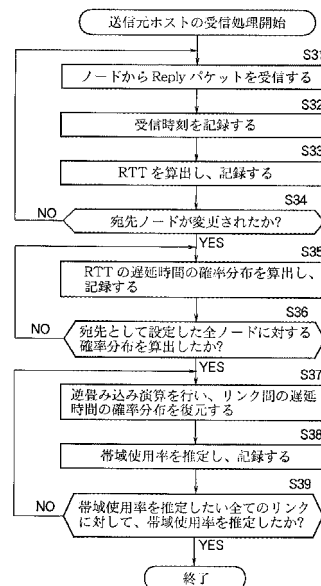
(57) 【要約】

【課題】 通信経路上の各リンクの帯域使用率を算出できるようにする。

【解決手段】 通信経路上の各ノードに対して、複数回、Requestパケットを送信し、Replyパケットを受信する。同一のノードからの複数回のRequestパケットの送信時刻からReplyパケットの受信時刻までの時間(RTT)を計測し、RTTの遅延時間の確率密度関数を作成する。各ノードにおいて、確率密度関数を用いた逆畳み込み演算を行う。逆畳み込み演算を行うことで、通信経路上の所定のリンク間の遅延時間の確率分布を復元することができる。そして、復元された遅延時間の確率分布を用いて、所定のリンク間の帯域使用率が推定される。本発明は、ネットワーク内の通信精度を確保するために、帯域使用率を測定するコンピュータに適用できる。

【選択図】 図6

図6



**【特許請求の範囲】****【請求項 1】**

第 1 のホストと第 2 のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置において、

前記複数のノードまたは前記第 2 のホストのうちの 1 つを宛先ノードに設定し、前記宛先ノードに対して、第 1 のパケットを送信し、その第 1 のパケットに対する前記宛先ノードからの第 2 のパケットを受信し、前記送信から前記受信までの時間を計測する計測手段と、

前記計測を、複数回、同一の宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出する第 1 の算出手段と、

前記第 1 の算出手段により算出された前記確率密度関数を、前記所定のリンクの両端のノードを宛先として算出し、算出された確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出する第 2 の算出手段と、

前記第 2 の算出手段により算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定する推定手段と

を備える情報処理装置。

**【請求項 2】**

前記第 1 のパケットには、前記第 2 のパケットを各ノードにおいて優先的に処理するかどうかを示すフラグが含まれる

請求項 1 に記載の情報処理装置。

**【請求項 3】**

第 1 のホストと第 2 のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置の情報処理方法において、

前記複数のノードまたは前記第 2 のホストのうちの 1 つを宛先ノードに設定し、前記宛先ノードに対して、第 1 のパケットを送信し、その第 1 のパケットに対する前記宛先ノードからの第 2 のパケットを受信し、前記送信から前記受信までの時間を計測し、

前記計測を、複数回、同一の宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出し、

前記所定のリンクの両端のノードを宛先として算出し、

算出された確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出し、

算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定する

ステップを含む情報処理方法。

**【請求項 4】**

コンピュータに、

第 1 のホストと第 2 のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置に、

前記複数のノードまたは前記第 2 のホストのうちの 1 つを宛先ノードに設定し、前記宛先ノードに対して、第 1 のパケットを送信し、その第 1 のパケットに対する前記宛先ノードからの第 2 のパケットを受信し、前記送信から前記受信までの時間を計測し、

前記計測を、複数回、同一の宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出し、

前記所定のリンクの両端のノードを宛先として算出し、

算出された確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出し、

算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定する

ステップを含む処理を実行させるためのコンピュータ読み取り可能なプログラム。

**【請求項 5】**

10

20

30

40

50

第1のホストと第2のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置において、

前記第2のホストを宛先ノードに設定し、前記宛先ノードに対して、T T L (Time To Live) を含む第1のパケットを送信し、その第1のパケットに対する前記ノードからの第2のパケットを受信し、前記送信から前記受信までの時間を計測する計測手段と、

前記計測を、複数回、同一のT T Lを含む前記第1のパケットを前記宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出する第1の算出手段と、

前記第1の算出手段により算出された前記確率密度関数を、前記T T Lの値を増やすことで、前記複数のノードの全てにおいて算出し、複数の確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出する第2の算出手段と、

前記第2の算出手段により算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定する推定手段と

を備える情報処理装置。

#### 【請求項6】

第1のホストと第2のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置の情報処理方法において、

前記第2のホストを宛先ノードに設定し、前記宛先ノードに対して、T T L (Time To Live) を含む第1のパケットを送信し、その第1のパケットに対する前記ノードからの第2のパケットを受信し、前記送信から前記受信までの時間を計測し、

前記計測を、複数回、同一のT T Lを含む前記第1のパケットを前記宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出し、

算出された前記確率密度関数を、前記T T Lの値を増やすことで、前記複数のノードの全てにおいて算出し、複数の確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出し、

算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定する

ステップを含む情報処理方法。

#### 【請求項7】

コンピュータに、

第1のホストと第2のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置に、

前記第2のホストを宛先ノードに設定し、前記宛先ノードに対して、T T L (Time To Live) を含む第1のパケットを送信し、その第1のパケットに対する前記ノードからの第2のパケットを受信し、前記送信から前記受信までの時間を計測し、

前記計測を、複数回、同一のT T Lを含む前記第1のパケットを前記宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出し、

算出された前記確率密度関数を、前記T T Lの値を増やすことで、前記複数のノードの全てにおいて算出し、複数の確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出し、

算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定する

ステップを含む処理を実行させるためのコンピュータ読み取り可能なプログラム。

#### 【発明の詳細な説明】

#### 【技術分野】

#### 【0001】

本発明は情報処理装置および方法、並びにプログラムに関し、特に、ネットワークの帯域使用率を推定する際に用いて好適な情報処理装置および方法、並びにプログラムに関する。

#### 【背景技術】

10

20

30

40

50

## 【 0 0 0 2 】

近年、ネットワークのトラフィックが増大し、通信効率の維持を目的として、ネットワークの通信制御が行われつつある。ネットワークの通信制御を行うためには、ネットワークの性能評価が必要である。ネットワークの性能評価の手法として、エンドホスト間の通信経路のリンクの帯域使用率を測定する手法がある。リンクの帯域使用率とは、そのリンクの物理帯域のうち、どれくらいの帯域が使用されているかの割合を示す値である。このリンク帯域使用率が大きいほど、そのリンクには負荷が掛かっていることになる。

## 【 0 0 0 3 】

負荷が掛かっているリンクがネットワーク内に存在すると、ネットワーク自体の性能が低下してしまう。ネットワーク内のエンドホスト間において通信経路の候補が複数ある場合、もっとも負荷が小さい経路を選択することで、負荷が掛かっているリンクを避けて通信を行うことが可能となる。このようなことを可能とするために、上記したように、リンク帯域使用率を測定するといったネットワークの性能評価が行われている。(非特許文献 1 乃至 5 参照)

## 【 先行技術文献 】

## 【 非特許文献 】

## 【 0 0 0 4 】

【 非特許文献 1 】 滝田英勝, 杉崎義雄, 山口実靖, 浅谷耕一, “ RTT を用いたボトルネックリンクの帯域使用率推定法 ”, 信学技報, IA2008-15, pp.13-18 .

【 非特許文献 2 】 M. Jain and C. Dovrolis, “ End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Through-put ”, In Proceedings of the 2002 SIGCOMM conference, 2002.

【 非特許文献 3 】 Vinay Ribeiro, Rudolf Riedi, Richard Bara-niuk, Jiri Navratil, and Les Cottrell “ pathChirp:Efficient Available Bandwidth Estimation for Network Paths, ”, Passive and Active Measurement Workshop, 2003.

【 非特許文献 4 】 Ningning Hu and Peter Steenkiste, “ Estimating Available Bandwidth Using Packet Pair Probing ”, 2002 .

【 非特許文献 5 】 “ Tobi Oetiker's MRTG - The Multi Router Traffic Grapher ”, <http://oss.oetiker.ch/mrtg/>

## 【 発明の概要 】

## 【 発明が解決しようとする課題 】

## 【 0 0 0 5 】

提案されている従来のエンドホスト間の帯域使用率を求める手法として、通信経路中の各ノードに測定用のソフトウェアを組み込み、各ノードが、トラフィックの流量を監視し、解析する手法がある。この手法によれば、帯域使用率を正確に測れる利点があるが、各ノードに測定用のソフトウェアを組み込まなくてはならず、その手間がかかる、費用がかかるといった問題があった。また、各ノードがトラフィックの流量を監視し、解析するために、各ノードに、そのような処理を行うための負荷を与えてしまうことになる。

## 【 0 0 0 6 】

また提案されている従来の手法として、Pingを用いた方式がある。Pingは、ネットワーク中のノードに通じるかを判定するものであり、相手のノードにRequestを送り、相手はReplyを送り手にReplyを返すものである。このPingを、通信経路のエンドホストからもう一方のエンドホストに送ることで、RTT(Round-Trip-Time)の最小値の出現頻度を得て、帯域使用率を推定する方式が提案されている。この方式によれば、Pingを送るといった簡易な方法で、帯域使用率を測定することができるが、1ホップの通信経路でしか、帯域使用率を測定することができない。

## 【 0 0 0 7 】

本発明は、このような状況に鑑みてなされたものであり、簡便な手法で、帯域使用率を求め、ネットワークの効率的な利用を促すことができるようにするものである。

## 【 課題を解決するための手段 】

## 【0008】

本発明の一側面の第1の情報処理装置は、第1のホストと第2のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置において、前記複数のノードまたは前記第2のホストのうちの1つを宛先ノードに設定し、前記宛先ノードに対して、第1のパケットを送信し、その第1のパケットに対する前記宛先ノードからの第2のパケットを受信し、前記送信から前記受信までの時間を計測する計測手段と、前記計測を、複数回、同一の宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出する第1の算出手段と、前記第1の算出手段により算出された前記確率密度関数を、前記所定のリンクの両端のノードを宛先として算出し、算出された確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出する第2の算出手段と、前記第2の算出手段により算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定する推定手段とを備える。

10

## 【0009】

前記第1のパケットには、前記第2のパケットを各ノードにおいて優先的に処理するかどうかを示すフラグが含まれるようにすることができる。

## 【0010】

本発明の一側面の第1の情報処理方法は、第1のホストと第2のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置の情報処理方法において、前記複数のノードまたは前記第2のホストのうちの1つを宛先ノードに設定し、前記宛先ノードに対して、第1のパケットを送信し、その第1のパケットに対する前記宛先ノードからの第2のパケットを受信し、前記送信から前記受信までの時間を計測し、前記計測を、複数回、同一の宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出し、前記所定のリンクの両端のノードを宛先として算出し、算出された確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出し、算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定するステップを含む。

20

## 【0011】

本発明の一側面の第1のプログラムは、コンピュータに、第1のホストと第2のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置に、前記複数のノードまたは前記第2のホストのうちの1つを宛先ノードに設定し、前記宛先ノードに対して、第1のパケットを送信し、その第1のパケットに対する前記宛先ノードからの第2のパケットを受信し、前記送信から前記受信までの時間を計測し、前記計測を、複数回、同一の宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出し、前記所定のリンクの両端のノードを宛先として算出し、算出された確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出し、算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定するステップを含む処理を実行させるためのコンピュータ読み取り可能なプログラムである。

30

## 【0012】

本発明の一側面の第1の情報処理装置および方法、並びにプログラムにおいては、設定された宛先ノードに対して、第1のパケットが送信され、その第1のパケットに対する宛先ノードからの第2のパケットが受信され、送信から受信までの時間が計測される。その計測が、複数回、同一の宛先ノードに対して行われ、送信から受信までの時間の確率密度関数が算出される。複数のノードにおける確率密度関数に対して逆畳み込み演算が行われ、所定のリンクの確率分布が算出される。そして、その所定のリンクの確率分布から所定のリンクにおける帯域使用率が推定される。

40

## 【0013】

本発明の一側面の第2の情報処理装置は、第1のホストと第2のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報

50

処理装置において、前記第2のホストを宛先ノードに設定し、前記宛先ノードに対して、T T L (Time To Live) を含む第1のパケットを送信し、その第1のパケットに対する前記ノードからの第2のパケットを受信し、前記送信から前記受信までの時間を計測する計測手段と、前記計測を、複数回、同一のT T Lを含む前記第1のパケットを前記宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出する第1の算出手段と、前記第1の算出手段により算出された前記確率密度関数を、前記T T Lの値を増やすことで、前記複数のノードの全てにおいて算出し、複数の確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出する第2の算出手段と、前記第2の算出手段により算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定する推定手段とを備える。

10

**【0014】**

本発明の一側面の第2の情報処理方法は、第1のホストと第2のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置の情報処理方法において、前記第2のホストを宛先ノードに設定し、前記宛先ノードに対して、T T Lを含む第1のパケットを送信し、その第1のパケットに対する前記ノードからの第2のパケットを受信し、前記送信から前記受信までの時間を計測し、前記計測を、複数回、同一のT T Lを含む前記第1のパケットを前記宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出し、算出された前記確率密度関数を、前記T T Lの値を増やすことで、前記複数のノードの全てにおいて算出し、複数の確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出し、算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定するステップを含む。

20

**【0015】**

本発明の一側面の第2のプログラムは、第1のホストと第2のホストとの通信経路上に、複数のノードが存在するネットワーク内の所定のリンクの帯域使用率を推定する情報処理装置に、前記第2のホストを宛先ノードに設定し、前記宛先ノードに対して、T T Lを含む第1のパケットを送信し、その第1のパケットに対する前記ノードからの第2のパケットを受信し、前記送信から前記受信までの時間を計測し、前記計測を、複数回、同一のT T Lを含む前記第1のパケットを前記宛先ノードに対して行うことで、前記送信から前記受信までの時間の確率密度関数を算出し、算出された前記確率密度関数を、前記T T Lの値を増やすことで、前記複数のノードの全てにおいて算出し、複数の確率密度関数に対して逆畳み込み演算を行うことで、所定のリンクの確率分布を算出し、算出された前記所定のリンクの確率分布から前記所定のリンクにおける帯域使用率を推定するステップを含む処理を実行させるコンピュータが読み取り可能なプログラムである。

30

**【0016】**

本発明の一側面の第2の情報処理装置および方法、並びにプログラムにおいては、設定された宛先ノードに対して、T T Lを含む第1のパケットが送信され、その第1のパケットに対するノードからの第2のパケットが受信され、送信から受信までの時間が計測される。そのような計測が、複数回、同一のT T Lを含む第1のパケットを宛先ノードに対して行うことで行われ、送信から受信までの時間の確率密度関数が算出される。確率密度関数は、T T Lの値が増やされることで、複数のノードの全てにおいて算出され、複数の確率密度関数に対して逆畳み込み演算が行われることで、所定のリンクの確率分布が算出される。そして算出された所定のリンクの確率分布から所定のリンクにおける帯域使用率が推定される。

40

**【発明の効果】****【0017】**

本発明の一側面によれば、ネットワーク内の所定のリンク間の帯域使用率を推定することが可能となる。

**【図面の簡単な説明】****【0018】**

50

- 【図 1】 1 ホップのネットワークについて説明するための図である。  
 【図 2】 複数ホップのネットワークについて説明するための図である。  
 【図 3】 リンク間の帯域使用率の算出について説明するための図である。  
 【図 4】 エンドホスト、ノードの機能について説明するための図である。  
 【図 5】 送信元ホストの送信処理について説明するためのフローチャートである。  
 【図 6】 送信元ホストの受信処理について説明するためのフローチャートである。  
 【図 7】 中継ノードの処理について説明するためのフローチャートである。  
 【図 8】 宛先ノードの処理について説明するためのフローチャートである。  
 【図 9】 送信元ホストの他の処理について説明するためのフローチャートである。  
 【図 10】 中継ノードの他の処理について説明するためのフローチャートである。  
 【図 11】 記録媒体について説明するための図である。

10

【発明を実施するための形態】

【0019】

以下に、本発明の実施の形態について図面を参照して説明する。

【0020】

本発明は、ネットワークのリンクの帯域使用率を測定するのに適用できる。以下に説明する本発明は、通信経路中のエンドホストと各ノード間でPingを送り、エンドホストと各ノード間のRTTの遅延時間の確率密度関数を得て、逆畳みこみを利用することで、リンクの帯域使用率を推定する。

【0021】

20

Pingとは、ネットワーク上の所定のノードに対する応答要求(Request パケット)を送り、要求に対して当該ノードが応答確認(Replyパケット)を返す一連の動作のことである。以下の説明においては、Pingを用いた方式を例にあげて説明するが、所定のノードに対して、パケットを送信し、その送信に対応した応答を受信する仕組みであれば、本発明に適用できるため、本発明の適用範囲が、IP ネットワークにおけるPingに限定されることを意味するわけではない。

【0022】

パケットとは、2点間のホストで通信を行うためのデータの転送単位のことであるとする。ここでは、パケットとの記載をするが、本発明に適用できるのは、ネットワークにおけるパケットだけではなく、そのようなパケットだけに限定されることを意味する記載ではない。RTTは、送信側のエンドホストが、Request パケットを送ってから、Replyパケットを受信するまでの時間である。

30

【0023】

ネットワークのリンクの帯域使用率について説明する。まず、ネットワークとして、図1に示したようなネットワークを例にあげて説明する。図1に示したネットワークは、エンドホスト11がノード12と接続され、ノード12がエンドホスト13と接続されている。ノード12は、キュー21を備える。このようなネットワークにおいて、エンドホスト11からエンドホスト13に至る通信経路には、1つのノード12しか存在しておらず、このようなネットワークを1ホップからなる通信経路と称する。

【0024】

40

1ホップからなる通信経路においては、RTTの遅延時間は、ノード12で起こることになる。すなわち、RTTは、次式(1)で表すことができる。

$$RTT = \text{Const} + T_{\text{que}} \quad \dots (1)$$

式(1)において、Constは、次式(2)で表される値であり、T<sub>que</sub>は、キュー21におけるキューイング遅延である。このキューイング遅延とは、出力インタフェースであるキュー21に、パケットが格納されている時間である。

$$\text{Const} = T_{\text{trs}} + T_{\text{fw}} + T_{\text{prop}} \quad \dots (2)$$

式(2)において、T<sub>trs</sub>は処理遅延を表し、T<sub>fw</sub>は転送遅延を表し、T<sub>prop</sub>は伝搬遅延を表す。これらのT<sub>trs</sub>、T<sub>fw</sub>、T<sub>prop</sub>は、ノード12の処理能力などに依存し、ノード固有の値となる。よって、Const、すなわち、constant(一定値)は、通信経路固有の値

50

となる。

【0025】

式(1)において、Constが一定値であるとする、RTTの変動に係わってくるのは、T<sub>que</sub>ということになる。このT<sub>que</sub>は、上記したように、キュー21におけるキューイング遅延である。RTTが小さくなるのは、キューイング遅延が小さいときであり、RTTが大きくなるのは、キューイング遅延が大きいときである。すなわち、RTTの最小値RTT<sub>min</sub>は、ノード12のキュー21にパケットがないときに観測され、このとき、リンクの使用率は0となる。また、RTTの最小値RTT<sub>min</sub>以外のRTT<sub>other</sub>は、ノード12のキュー21にパケットがあるときであり、このときのリンクの使用率は1である。

【0026】

所定の時刻におけるRTT<sub>min</sub>とRTT<sub>other</sub>が測定された回数を、それぞれN<sub>min</sub>とN<sub>other</sub>とした場合、帯域使用率Uは、次式(3)で表される。

$$\text{帯域使用率 } U = N_{\text{other}} / (N_{\text{min}} + N_{\text{other}}) \quad \dots (3)$$

遅延時間の確率密度関数における、最小値の出現確率が、そのリンクの使用できる帯域率Aとなる。すなわち、使用できる帯域率Aと帯域使用率Uの関係は次式(4)となる。

$$U = 1 - A \quad \dots (4)$$

【0027】

図1に示したように、1ホップの通信経路の場合、エンドホスト11とエンドホスト13の2点間の通信経路における帯域使用率は、1ホップ、すなわち、ノード12の影響のみを考慮すれば良く、上記したようにして、帯域使用率Uなどを求めることができる。しかしながら、通常、図2に示すように、エンドホストからエンドホストまでの間には、複数のノードが存在し、複数のリンクが存在することになる。

【0028】

図2に示したネットワークには、エンドホスト111とエンドホスト113との2点間を結ぶ通信経路上に、ノード112-1乃至112-Nが存在している。また、ノード112-1とノード112-2は、直接接続されているとともに、ノード131-1を介しても接続されている。また各ノードは、キュー121-1乃至121-N, 141-1を備えている。以下の説明において、ノード112-1乃至112-Nを個々に区別する必要がない場合、単にノード112(ノード131-1も含む)と記載する。他の記載も同様に記載する。

【0029】

図2に示したネットワークには、複数のノード112が存在し、複数のリンクが存在する。このような場合、RTTは、各ノード112のキューイング遅延の影響を受けることになる。そして、エンドホスト111とエンドホスト113との2点間のRTT、すなわち通信経路全体の遅延時間の確率密度関数の最小値の出現頻度は、各リンクにおける遅延時間の確率密度関数の最小値の出現頻度の積となる。エンドホスト111とエンドホスト113との2点間のRTTは求めることができ、帯域使用率を推定することはできるが、個々のリンクの帯域使用率を推定することができない。

【0030】

個々のリンクの帯域使用率を推定できないと効率の良い通信経路を推定することができない。例えば、ノード112-1とノード112-2の間のリンク(リンクAとする)の帯域使用率が高く、ノード112-1からノード131-1を介してノード112-2へ行く経路のリンク(リンクBとする)の帯域使用率が低いような場合を考える。このような場合、リンクAが選択されて通信が行われるよりも、リンクBが選択されて通信が行われる方が、効率的に通信を行えるが、リンクAとリンクBの帯域使用率をそれぞれ別個に推定できないため、リンクAが選択されて通信が行われる可能性がある。効率的に通信が行われるようにするためには、個々のリンクの帯域使用率を推定することが好ましい。

【0031】

そこで、以下に説明するように帯域使用率を推定することで、個々のリンクの帯域使用率を推定することを可能とする。ここでは、図2に示したネットワークモデルのうち、図

10

20

30

40

50



3 に示したネットワークモデルを例にあげ、個々のリンクの帯域使用率の推定について説明を加える。図 3 に示したネットワークモデルは、エンドホスト 1 1 1、ノード 1 1 2 - 1 乃至 1 1 2 - 3 から構成されている。

【 0 0 3 2 】

図 3 に示したネットワークモデルにおいては、エンドホスト 1 1 1 からノード 1 1 2 - 1 を介してノード 1 1 2 - 2 へのリンクと、ノード 1 1 2 - 2 からノード 1 1 2 - 3 へのリンクが存在している。エンドホスト 1 1 1 とノード 1 1 2 - 2 のリンクの遅延時間の確率密度関数を  $f(x)$  とし、ノード 1 1 2 - 2 とノード 1 1 2 - 3 のリンクの遅延時間の確率密度関数を  $g(x)$  とした場合、この 2 つのリンクを通過するエンドホスト 1 1 1 とノード 1 1 2 - 3 における遅延時間の確率密度関数は、 $h(x)$  となり、この  $h(x)$  は、次式 (5) で表すことができる。

10

【数 1】

数 1

$$h(x) = (f * g)(x) = \int_{-\infty}^{+\infty} f(x) g(x - \tau) d\tau \quad \dots\dots (5)$$

20

【 0 0 3 3 】

式 (5) より、確率密度関数  $h(x)$  は、確率密度関数  $f(x)$  と確率密度関数  $g(x)$  との畳み込み演算により算出できることがわかる。さらに、離散値定義される関数の場合、式 (5) のように積分ではなく、次式 (6) に示すような総和で定義することができる。

【数 2】

数 2

$$h(m) = \sum_n f(n) g(n - m) \quad \dots\dots (6)$$

30

【 0 0 3 4 】

式 (6) を、具体的に記載すると、以下のようなになる。

40

$$h(0) = f(0)g(0) \quad \dots\dots (7 - 1)$$

$$h(1) = f(0)g(1) + f(1)g(0) \quad \dots\dots (7 - 2)$$

$$h(2) = f(0)g(2) + f(1)g(1) + f(2)g(0) \quad \dots\dots (7 - 3)$$

$$h(3) = f(0)g(3) + f(1)g(2) + f(2)g(1) + f(3)g(0) \quad \dots\dots (7 - 4)$$

.....

$$h(n) = f(0)g(n) + f(1)g(n-1) + \dots\dots + f(n-1)g(1) + f(n)g(0) \quad \dots\dots (7 - n)$$

【 0 0 3 5 】

ここで、再度図 3 を参照する。エンドホスト 1 1 1 が、図 3 に示したネットワークの帯域使用率を推定する場合を考える。確率密度関数  $f(x)$  は、エンドホスト 1 1 1 を一方のエンドホストとし、ノード 1 1 2 - 2 を他方のエンドホストとし、エンドホスト 1 1 1

50

からノード 1 1 2 - 2 に対してPingを送る ( R T Tを測定する ) ことで求めることができる。また同様に、確率密度関数  $h(x)$  は、エンドホスト 1 1 1 を一方のエンドホストとし、ノード 1 1 2 - 3 を他方のエンドホストとし、エンドホスト 1 1 1 からノード 1 1 2 - 2 に対してPingを送る ( R T Tを測定する ) ことで求めることができる。

【 0 0 3 6 】

しかしながら、ノード 1 1 2 - 2 を一方のエンドホストとし、ノード 1 1 2 - 3 を他方のエンドホストとし、ノード 1 1 2 - 2 とノード 1 1 2 - 3 とのリンクの確率密度関数  $g(x)$  を、エンドホスト 1 1 1 が求めることはできない。このことは、例えば、図 2 に示したネットワークモデルにおいて、エンドホスト 1 1 1 とエンドホスト 1 1 3 との 2 点間の通信経路上に存在するノード 1 1 2 のうちの所定の 2 台のノード 1 1 2 のリンクにおける確率密度関数をエンドホスト 1 1 1 が求めることはできないことを示している。

10

【 0 0 3 7 】

ここで、式 ( 7 ) を再度参照する。式 ( 7 - 1 ) を変形すると、次式 ( 8 - 1 ) になる。

$$g(0) = h(0) / f(0) \dots (8-1)$$

(但し、 $f(0)$  は、0 ではない。他の式においても同様とする)

また、同様に、式 ( 7 - 2 ) を変形すると、次式 ( 8 - 2 ) になる。

$$g(1) = (h(1) - f(1)g(0)) / f(0) \dots (8-2)$$

【 0 0 3 8 】

式 ( 8 - 1 ) から、 $g(0)$  を求めることができる。上記したように、確率密度関数  $f(x)$  と確率密度関数  $h(x)$  は、エンドホスト 1 1 1 からPingを送信することで推定することができるため、確率密度関数  $f(0)$  と確率密度関数  $h(0)$  は求めることができ、その求められる確率密度関数  $f(0)$  と確率密度関数  $h(0)$  を式 ( 8 - 1 ) に代入すれば、確率密度関数  $g(0)$  を求めることができる。

20

【 0 0 3 9 】

また、同様に、式 ( 8 - 2 ) から、確率密度関数  $g(1)$  を求めることができる。式 ( 8 - 2 ) において、確率密度関数  $h(1)$ 、確率密度関数  $f(1)$ 、確率密度関数  $f(0)$  は、測定により求めることが可能であり、確率密度関数  $g(0)$  は、式 ( 8 - 1 ) から求めることが可能であるため、確率密度関数  $g(1)$  も、式 ( 8 - 2 ) から求めることが可能である。

30

【 0 0 4 0 】

このように、逆畳み込み演算を行うことで、所定のノード 1 1 2 間の確率密度関数  $g(x)$  を、エンドホスト 1 1 1 が求めることができる。すなわち、エンドホスト 1 1 1 にて、ネットワーク内の各リンクの確率密度関数  $g(x)$  を、個別に求めることができる。式 ( 7 - n ) を変形することで、次式 ( 8 - n ) を求めることができ、この式 ( 8 - n ) から、ネットワーク内の所定のリンク n の確率密度関数  $g(n)$  を求めることができる。

【 数 3 】

数 3

40

$$g(n) = \frac{h(n) - \sum_{i=0}^{n-1} f(n-i)g(i)}{f(0)} \dots (8-n)$$

【 0 0 4 1 】

このようにして、逆畳み込み演算を行うことで推定された遅延時間の確率密度関数  $g(x)$

50

の最小値の統計をとることで、そのリンクで使用できる帯域率  $A$  を推定することができる。すなわち、エンドホスト 1 1 1 が直接求められない遅延時間の確率密度関数  $g(x)$  は、直接求めることができる確率密度関数  $h(x)$  と確率密度関数  $f(x)$  を観測し、逆畳みこみ演算を行うことで求めることができる。このような逆畳みこみ演算を、通信経路中の全リンクに対して行うことで、全リンクの帯域使用率  $U$  (帯域率  $A$ ) を推定することができ、通信経路の性能を推定することが可能となる。

【0042】

ここで再度図3を参照する。エンドホスト 1 1 1 が、ノード 1 1 2 - 2 に対してPingを送信し、RTTを測定した場合、エンドホスト 1 1 1 から、ノード 1 1 2 - 1 を介してノード 1 1 2 - 2 にRequestパケットが送信される。そして、ノード 1 1 2 - 2 からノード 1 1 2 - 1 を介して、エンドホスト 1 1 1 にReplyパケットが送信される。RTTは、エンドホスト 1 1 1 において、Requestパケットを送信してからReplyパケットを受信するまでの時間である。よって、RTTには、ノード 1 1 2 - 1 において、Requestパケットが処理される時間(往路のキューイング遅延と称する)と、Replyパケットが処理される時間(復路のキューイング遅延と称する)が含まれる。

10

【0043】

所定のノード間の確率密度関数  $g(x)$  を推定する際、往路のキューイング遅延と、復路のキューイング遅延を考慮する必要がある。例えば、往路のキューイング遅延と復路のキューイング遅延が異なるときがあるからである。ここでは特に復路のキューイング遅延を考慮した場合を説明する。

20

【0044】

第1の実施の形態として、復路のキューイング遅延がないネットワークを考える。復路のキューイング遅延がないネットワークは、上記したようにして、確率密度関数  $g(x)$  を推定し、使用率などを推定するのに、最良の形態のネットワークである。復路のキューイング遅延がない場合、往路のキューイング遅延だけを考慮すればよいため、逆畳みこみ演算によって、復元したリンクの遅延時間の確率密度関数は往路のみのものであると仮定して処理することが可能である。よって、復元した確率密度関数の最小値の出現頻度から、そのリンクの帯域使用率を一意に推定できる。

【0045】

第2の実施の形態として、復路のキューイング遅延が発生することを想定し、その復路のキューイング遅延ができるだけ発生しないように制御されることにより、復路のキューイング遅延がないネットワークと仮定して処理できるようにすることを考える。例えば、ICMP(Internet Control Message Protocol)のReplyパケットが、各ノードにおいて優先的に処理されるようにすることで実現する。通常、ノード 1 1 2 に受信されたパケットは、宛先アドレスに基づき次のノードの転送先が決定される。そして、そのパケットは出力インタフェースのキュー 1 2 1 (一時的な記憶領域)に格納される。受信されたパケットは、順次キュー 1 2 1 に格納されていく。

30

【0046】

そして、電気信号もしくは光信号に変換され送信されるまで、その領域に格納された状態とされる。これを通常処理もしくは通常制御と称する。仮に、パケットが大量に受信されると、このキューバッファ(キュー 1 2 1)に大量のパケットが格納され、送信されるまでの遅延が生ずる。これが、上記してきたキューイング遅延となる。

40

【0047】

一方、通常処理に対し優先処理と称する処理は、パケットヘッダに優先制御フラグが存在するパケットに対し、キュー 1 2 1 に格納せず、ただちに送信処理を行う処理のことである。キュー 1 2 1 にバッファリングされているパケットが存在しようとも、優先制御フラグがあるパケットは、ただちに送信処理されるため、キューイング遅延は発生しない状況を作り出すことができる。例えば、Replyパケットに、このような優先制御フラグを立てることで、Replyパケットは、受信されたノード 1 1 2 で優先的に処理されるため、復路でのキューイング遅延は発生しないことになる。

50

## 【 0 0 4 8 】

このReply パケットが優先処理に対応している状態を、ここではReply 優先モードと記述する。そして例えば、この優先制御を実現する手段としては、IP パケットヘッダにTOS (Type Of Service、優先制御するか否かのフラグを記録する領域) フィールドにフラグを立てるといったことで実現できる。または、他の手段を用いても良い。

## 【 0 0 4 9 】

このようなReply優先モードを設け、処理されるようにすることで、仮に、Replyパケットを受信したノード112のキュー121に、パケットが既に格納されていても、それらの格納されているパケットよりも先に処理される制御することができ、結果として、復路のキューイング遅延が無い状態にすることができる。復路のキューイング遅延が無い状態と仮定して処理することができれば、第1の実施の形態と同じく、逆置みこみ演算によって、復元したリンクの遅延時間の確率密度関数は往路のみのものであると仮定して処理することが可能となる。よって、この場合も、復元した確率密度関数の最小値の出現頻度から、そのリンクの帯域使用率を一意に推定できる。

10

## 【 0 0 5 0 】

第3の実施の形態として、往路と復路の経路が一致し、復路にもキューイング遅延がある場合を考える。往路と復路の経路が一致するとは、例えば、図2を参照するに、エンドホスト111から、ノード112-1、ノード112-2、・・・、ノード112-Nを介して、エンドホスト113に到達する往路の経路と、エンドホスト113から、ノード112-N、・・・、ノード112-2、ノード112-1を介して、エンドホスト111に到達する復路の経路が一致している場合である。

20

## 【 0 0 5 1 】

往路と復路の経路が一致し、復路にもキューイング遅延がある場合、逆置みこみ演算することで復元されたリンクの遅延時間の確率密度関数は、そのリンクの往路と復路の遅延時間の確率密度関数が畳み込まれたものとなる。よって、復元された遅延時間の確率密度関数の最小値の出現頻度から、リンクの帯域使用率は特定することはできない。そこで、復元された遅延時間の確率密度関数  $h(x)$  ( $h(x) = (f * g)(x)$ ) とし、その最小値の出現頻度  $h(0) = A$  とする。

## 【 0 0 5 2 】

帯域使用率  $U$  は、 $0 \leq U \leq 1$  を満たすので、往路と復路における、確率密度関数の最小値の出現頻度は、

30

$$\begin{aligned} A &= f(0) \cdot 1 \\ A &= g(0) \cdot 1 \\ f(0) \cdot g(0) &= A \end{aligned}$$

の条件が満たされる。これらの条件から、往路、復路のいずれかの使用できる帯域率の最低値は  $A$  であることになる。すなわち、往路または復路の帯域使用率は、 $1 - A$  を越えないということがわかる。帯域使用率は一意に推定できないが、往復路の最高の帯域使用率がわかるので、そのリンクの負荷の状況を推定することができる。

## 【 0 0 5 3 】

第4の実施の形態として、往路と復路の経路が一致せず、復路にもキューイング遅延がある場合を考える。往路と復路の経路が一致していない場合とは、図2に示したネットワークにおいて、往路は、ノード112-1からノード112-2にパケットが渡されたが、復路は、ノード112-2からノード131-1を介してノード112-1にパケットが渡されたような場合であり、このような経路のときには、往路と復路の経路が一致していないことになる。

40

## 【 0 0 5 4 】

第4の実施の形態においては、第3の実施の形態と同じく、特定のリンクの帯域使用率を一意に推定することは困難である。しかしながら、帯域使用率  $U$  は、 $0 \leq U \leq 1$  を満たし、リンクのいずれも帯域使用率  $U$  は、 $1 - A$  を越えることがない。よって、ネットワークに負荷が掛かっている箇所の特定はできるため、その負荷が掛かっている箇所を

50

避けて通信を行うなどの制御を行うことが可能となる。

【 0 0 5 5 】

第 5 の実施の形態として、途中ノードの IP アドレスが不明なネットワークである場合を考える。途中ノードの IP アドレスが不明なネットワークに対して、ICMP の Trace Route を用いることで、経路上に存在するノードの IP アドレスを特定することができる。Trace Route とは、送信元ホスト（たとえば、エンドホスト 1 1 1）から指定した宛先ノード（たとえば、エンドホスト 1 1 3）までの通信経路を調査することを意味する。まず送信元ホストは指定した宛先ノードまで、TTL が 1 の Request パケットを送信する。TTL とは、パケットがノードを何回通過できるかの回数を表す。中継ノードにパケットが到着すると、その中継ノードにより、TTL の値が 1 だけ減算された値とされる。その結果、TTL の値が 0 になった場合、その中継ノードにより、送信元ホストに向かって宛先到達不能通知が送信されるとともに、当該パケットは破棄される。

10

【 0 0 5 6 】

宛先到達不能通知には、パケットを破棄した中継ノードの IP アドレスが記載されているため、そのパケットを受信した送信元ホストは、パケットを破棄した中継ノードの IP アドレスを取得することができる。このような宛先到達不能通知を受信されると、送信元ホストは、TTL を 1 だけ増やした値のパケットを送信する。このようにパケットを送信することで、パケットを破棄した中継ノードの次の中継ノードの IP アドレスを取得することができる。通信経路上の各中継ノードから、このような宛先到達不能通知を順次受信することで、送信元ホストは、経路上の各中継ノードの IP アドレスを取得する。なお、ここでは、Trace Route を例にあげて説明を続けるが、通信経路上の中継ノードの IP アドレスを取得する方法が、Trace Route に限定されることを示す記載ではなく、他の方法により、通信経路上の中継ノードの IP アドレスが取得されるような場合であっても、本発明を適用することはできる。

20

【 0 0 5 7 】

このように、通信経路のエンドホスト（送信元ホスト）は、もう一方のエンドホスト（宛先ノード）に向かって、TTL（初期値 1）が小さい Request パケットを複数回送信する。そして、TTL の値を 1 ずつ増加させるたびに複数回パケットが送信され、順次中継ノードの IP アドレスが取得される。このような処理が、宛先ノードから Reply パケットが届くまで繰り返される。この処理によれば、IP アドレスの不明なネットワークにおいて、通信経路上の中継ノードの IP アドレスが取得できるだけでなく、途中ノードから、順々に TTL 切れの宛先到達不能通知が送信元ホストに届くので、送信元ホストと各中継ノード間の RTT の遅延時間の確率密度関数も同時に取得できることになる。確率密度関数が取得できるので、第 1 の実施の形態乃至第 4 の実施の形態と同じく、逆畳み込み演算により、リンクの帯域使用率や負荷の状況を推定することができる。

30

【 0 0 5 8 】

次に、エンドホストや各中継ノードにおける処理について説明する。図 4 に示したネットワークモデルを用いて説明する。図 4 において、エンドホスト 1 1 1 を送信側のホスト（帯域使用率を算出するホスト）とし、エンドホスト 1 1 3 を最終的な宛先ノードとし、ノード 1 1 2 を中継ノードとする。ここで、エンドホスト 1 1 3 を最終的な宛先ノードと記載したのは、以下に説明するように、中継ノードが順次宛先ノードに設定されて処理され、最終的な宛先ノードがエンドホスト 1 1 3 であるからである。エンドホスト 1 1 1 は、送信処理部 1 5 1 と受信処理部 1 5 2 を備える。ノード 1 1 2 は、送受信処理部 1 6 1 を備える。エンドホスト 1 1 3 は、送受信処理部 1 7 1 を備える。

40

【 0 0 5 9 】

まず、図 5 のフローチャートを参照し、エンドホスト 1 1 1 が行う処理について説明する。図 5 のフローチャートを参照して説明するエンドホスト 1 1 1 が行う処理は、宛先ノードまでの通信経路上の中継ノードの IP アドレスがわかっている状態のときに行われる処理である。換言すれば、Trace Route を用いないときの、送信側のホストが行う送信時の処理である。さらに換言すれば、第 1 乃至第 4 の実施の形態に該当するときの送信元ホ

50

ストが実行する処理である。

【0060】

ステップS11において、送信処理部111は、中継ノードのうち、所定のノード112を宛先ノードに設定し、そのノードのアドレスをセットする。送信側のホストであるエンドホスト111は、エンドホスト113までの通信経路上に存在するノードのアドレスを認識している。そして、その通信経路上に存在し、帯域使用率を推定したいリンクの両端のノードの全てを、順次宛先ノードに設定し、Requestパケットを送信する。そのような処理を行うため、ステップS11においては、まだRequestパケットを送信していない通信経路上のノードのうち、1つのノードが選択され、宛先ノードに設定されるという処理が行われる。また、同一の宛先ノードに複数回Requestパケットを送信するため、アドレスがセットされる。

10

【0061】

ステップS12において、Replyパケット優先モードにするか否かが判断される。Replyパケット優先モードとは、上記したように、Replyパケットを中継ノードにおいて、他のパケットよりも優先して処理するように指示するためのモードであり、復路のキューイング遅延を考慮しなくても処理できるようにしたときなどに設定されるモードである。ステップS12において、Replyパケット優先モードにすると判断された場合、ステップS13に処理が進められ、Replyパケット優先モードにはしないと判断された場合、ステップS13の処理はスキップされ、ステップS14の処理に進められる。

20

【0062】

ステップS13において、Requestパケットヘッダに優先制御フラグが立てられる。Requestパケットヘッダに優先制御フラグが立てられた場合、上記したように、そのRequestパケットに対するReplyパケットは、各ノードにおいて優先的に処理される。ステップS13において、Requestパケットヘッダに優先制御フラグが立てられた場合、または、ステップS12において、Replyパケット優先モードにしないと判断された場合、ステップS14に処理が進められる。ステップS14において、Requestパケットが、セットされているアドレスを有する宛先ノードに対して送信され、その送信時刻が記録される。エンドホスト111は、記憶部(不図示)を有し、送信時刻を記憶する。記憶部としては、メモリ、ハードディスクなどを用いることが可能である。

30

【0063】

ステップS15において、所定の回数、同一の宛先ノードにRequestパケットを送信したか否かが判断される。所定の回数は、その宛先ノードに関するRTTの遅延時間の確率分布を得るのに十分な回数とされる。この所定の回数は、N回などと予め設定された固定値であっても良いし、確率分布を得るのに十分な結果が得られたと判断された時点までの回数とする可変値であっても良い。ステップS15において、所定の回数、同一の宛先ノードにRequestパケットを送信はしていないと判断された場合、ステップS12に処理が戻され、それ以降の処理が実行される。

40

【0064】

なお、ここでは、ステップS12に処理が戻されるとして説明を続けるが、ステップS13またはステップS14に処理が戻されるようにしても良い。ステップS12に処理が戻されることで、同一の宛先ノードに対して、Requestパケット優先モードでパケットを送信したり、Requestパケット優先モードでない状態でパケットを送信したりすることができるため、例えば、x回、Requestパケット優先モードで送信し、y回、Requestパケット優先モードではないモードで送信するといった使い分けを行うことが可能となる。また、モードを変えることで、復路の影響を計測することも可能となり、計測される復路の影響を、何らかの推測に用いるといったことも可能となる。

40

【0065】

ステップS13またはステップS14に処理が戻されるようにした場合、同一の宛先ノードには、同一のモードでRequestパケットが送信されることになる。よって、Requestパケット優先モードに設定された場合には、ステップS13に処理が戻され、Requestパケ

50

ット優先モードに設定されていない場合には、ステップ S 1 4 に処理が戻される。

【 0 0 6 6 】

一方、ステップ S 1 5 において、所定の回数、同一の宛先ノード（中継ノード）に Request パケットを送信したと判断された場合、ステップ S 1 6 に処理が進められる。ステップ S 1 6 において、通信経路上の宛先として設定したい全ノードに対して、Request パケットを送信したか否かが判断される。宛先として設定したい全ノードとは、帯域使用率を推定したい全リンクの両端に位置するノードのことである。ステップ S 1 6 において、通信経路上の宛先として設定したい全ノードに対して、Request パケットを送信していないと判断された場合、ステップ S 1 7 に処理が進められる。ステップ S 1 7 において、通信経路上の中継ノードのうち、宛先ノードに設定していない中継ノードのうちの 1 つが選択され、その選択されたノードが新たな宛先ノードとして設定される処理が実行される。そして、新たに設定された宛先ノードに対して、ステップ S 1 2 以降の処理が繰り返される。

10

【 0 0 6 7 】

このようにして、通信経路上の宛先として設定したい全ノードに対して、Request パケットが、複数回送信される。エンドホスト 1 1 1 は、Request パケットを送信することにより、宛先ノードから Reply パケットを受信する。この受信に関する処理および受信後に行われる処理に関して、図 6 のフローチャートを参照して説明する。

【 0 0 6 8 】

図 6 のフローチャートは、エンドホスト 1 1 1（送信側のホストであり、リンクの帯域使用率などを測定するホスト）の受信処理部 1 5 2 が行う処理である。ステップ S 3 1 において、宛先ノードに設定されているノードから Reply パケットが受信される。ステップ S 3 2 において、Reply パケットの受信時刻が記録される。そして、ステップ S 3 3 において、R T T が算出され、記録される。R T T が算出され、記憶されると、ステップ S 3 4 において、宛先ノードが変更されたか否かが判断される。この判断は、例えば、ステップ S 1 5（図 5）において、所定の回数、同一の宛先ノードに Request パケットを送信したと判断され、ステップ S 1 6 に処理が進められたとき、ステップ S 3 4 において宛先ノードが変更されたと判断される。

20

【 0 0 6 9 】

エンドホスト 1 1 1 は、図 5 のステップ S 1 4 における処理で、宛先ノードに Request パケットを送信するため、その Request パケットを受信した宛先ノードから、Reply パケットを受信する。またエンドホスト 1 1 1 は、Request パケットを送信した送信時刻と、その Request パケットに対応する Reply パケットの受信時刻を記録しているため、その送信時刻と受信時刻の差分を計算することで、R T T を算出することができる。また、同一の宛先ノードに、所定の回数、Request パケットを送信しているため、所定の回数分だけ、同一の宛先ノードに対する R T T が算出される。複数の R T T が算出されることで、R T T の確率分布を算出することが可能な状態となる。

30

【 0 0 7 0 】

ステップ S 3 4 において、宛先ノードが変更されたと判断された場合、ステップ S 3 5 に処理が進められる。ステップ S 3 5 において、R T T の遅延時間の確率分布が算出され、記録される。この処理は、その時点で処理対象とされている 1 つの宛先ノードに対して行われる。すなわち、ステップ S 3 1 乃至 S 3 4 の処理が繰り返されることで、1 つの宛先ノードに対する複数の R T T が算出される。そして、その複数の R T T の度数分布が取られることで、その宛先ノードに関する R T T の遅延時間の確率分布が算出される。

40

【 0 0 7 1 】

ステップ S 3 6 において、宛先として設定した全ての中継ノードに対する R T T の遅延時間の確率分布が算出されたか否かが判断される。通信経路上に存在する、帯域使用率を推定したいリンクの両端の全中継ノードに対して、Request パケットが複数回送信されているため、通信経路上に存在する、帯域使用率を推定したいリンクの両端の中継ノード毎に、R T T の遅延時間の確率分布が算出される。よって、指定した全ての中継ノードの R

50

TTの遅延時間の確率分布が算出されたか否かが、ステップS36において判断される。なお、ステップS31乃至34の処理と、ステップS35, S36の処理は、並列処理される。すなわち、ステップS31乃至S34の処理が実行されることで、所定の中継ノードに関するRTTが収集され、ステップS35, S36の処理が実行されることで、中継ノード毎のRTTの遅延時間の確率分布が算出される。

#### 【0072】

ステップS36において、指定した全ての中継ノードに対するRTTの遅延時間の確率分布が算出されたと判断された場合、ステップS37に処理が進められる。ステップS37において、逆畳み込み演算が実行され、リンク間の遅延時間の確率分布が復元される。ステップS36までの処理で、上述した確率密度関数 $f(x)$ 、確率密度関数 $h(x)$ が求められているため、ステップS37においては、式(8)に基づき、確率密度関数 $g(x)$ が求められる。そして、ステップS38において、帯域使用率が推定され、記録される。そして、ステップS39において、帯域資料率を推定したい全てのリンクに対して帯域使用率が推定されたか否かが判断され、推定されていないと判断された場合、ステップS37に処理が戻され、それ以降の処理が繰り返され、推定されたと判断された場合、処理は終了される。

10

#### 【0073】

帯域使用率は、上記したように、帯域使用率 $U = 1 - A$ で算出することができる。またAは、使用できるリンクの割合であり、RTTminから算出できる値である。所定のリンクのRTTminは、所定のリンクの確率密度関数 $g(x)$ が求められた時点で算出されており、その算出されている値を用いることができる。よって、所定のリンクにおける確率密度関数 $g(x)$ が算出されることで、所定のリンクにおける帯域使用率を算出することができる。このようにして、本発明によれば、所定のリンクの帯域使用率などを、その所定のリンク外のホストが推定することができる。

20

#### 【0074】

次に、中継ノードの処理について、図7のフローチャートを参照して説明する。中継ノードは、エンドホスト111とエンドホスト113との間の通信経路上に存在するノード112であり、順次、エンドホスト111により宛先ノードに設定されるノードとの間に存在するノードである。例えば、図2において、宛先ノードがノード112-Nに設定されている場合、中継ノードは、ノード112-1乃至112-N-1の各ノードとなる。ここでは、中継ノードは、ノード112(図5)であり、図7のフローチャートの処理は、ノード112の送受信処理部161が行うとして説明を続ける。また、図7のフローチャートの処理は、Trace Routeが用いられていないときの処理である。

30

#### 【0075】

ステップS51において、パケットを受信する。ステップS51において受信されるパケットは、RequestパケットまたはReplyパケットである。ステップS52において、受信されたパケットのパケットヘッダに、優先制御フラグがあるか否かが判断される。受信されたパケットが、Replyパケットである場合、そのReplyパケットのパケットヘッダに、Replyパケット優先モードであるときには、優先制御フラグが立てられている。よって、そのようなフラグが立てられているか否かが、ステップS52において判断される。

40

#### 【0076】

ステップS52において、パケットヘッダに優先制御フラグが立てられていないと判断された場合、ステップS53に処理が進められ、その受信されたパケットは、通常処理される。一方、ステップS52において、パケットヘッダに優先制御フラグが立てられていると判断された場合、ステップS54に処理が進められ、その受信されたパケットは、優先処理される。通常処理、または優先処理されたパケットは、次のノード112に対して転送される。このように、中継ノードの基本的な処理は、受信されたパケットを次のノードに転送する処理である。

#### 【0077】

次に、図8のフローチャートを参照して、宛先ノードにおける処理について説明する。

50



宛先ノードは、エンドホスト 1 1 1 と宛先ノードとの間の通信経路上に存在するノード 1 1 2 と、エンドホスト 1 1 3 のうちのいずれか 1 つが設定される。図 8 のフローチャートの処理は、ノード 1 1 2 が宛先ノードとして設定されているときには、ノード 1 1 2 の送受信処理部 1 6 1 が実行する処理であり、エンドホスト 1 1 3 が宛先ノードとして設定されているときには、エンドホスト 1 1 3 の送受信処理部 1 7 1 が実行する処理である。

【 0 0 7 8 】

ステップ S 7 1 において、Request パケットが受信される。ステップ S 7 2 において、受信された Request パケットのパケットヘッダに、優先制御フラグが立てられているか否かが判断される。ステップ S 7 2 において、Request パケットのパケットヘッダに、優先制御フラグが立てられていると判断された場合、ステップ S 7 3 に処理が進められる。ステップ S 7 3 において、送信される Reply パケットのパケットヘッダに、優先制御フラグが立てられる。

10

【 0 0 7 9 】

ステップ S 7 3 において優先制御フラグが立てられた Reply パケットは、ステップ S 7 4 において、送信元ホスト、この場合、エンドホスト 1 1 1 に対して送信される。一方、ステップ S 7 2 において、Request パケットのパケットヘッダに、優先制御フラグは立てられていないと判断された場合、優先フラグが立てられていない Reply パケットが生成され、ステップ S 7 4 において、送信元ホストに対して送信される。

【 0 0 8 0 】

このように、宛先ノードに設定されているノード 1 1 2 ( エンドホスト 1 1 3 ) は、受信された Request パケットに、優先制御フラグが立てられているときには、優先制御フラグを立てた Reply パケットを生成し、送信し、優先制御フラグが立てられていないときには、優先制御フラグを立てない Reply パケットを生成し、送信する処理を、Request パケットを受信する毎に実行する。

20

【 0 0 8 1 】

上記した処理は、Trace Route を用いていないときの送信元ホストの処理、中継ノードの処理、および宛先ノードの処理であった。次に、Trace Route を用いていないときの送信元ホストの処理、中継ノードの処理、および宛先ノードの処理について説明する。まず、図 9 のフローチャートを参照し、送信元ホスト ( エンドホスト 1 1 1 ) における送信時の処理について説明する。

30

【 0 0 8 2 】

Trace Route が用いられる場合、通信経路中の各ノードのアドレスはわからない状態であり、一方のエンドホスト 1 1 3 ( 宛先ホスト ) のアドレスはわかっている状態である。そして、そのような状態のときに、送信元ホストは、宛先ホスト ( エンドホスト 1 1 3 ) に対して、Trace Route を実行する。Trace Route が用いられない送信処理 ( 図 5 のフローチャートを参照して説明した処理 ) では通信経路上の各ノードのアドレスが順次指定されて Request パケットが送信されたが、Trace Route が用いられた処理では宛先ホストのアドレスは固定され、その宛先ホストはエンドホスト 1 1 3 とされ、アドレスが変更されない状態で Request パケットが送信される点が異なる。図 5 のフローチャートを参照して説明した処理と同一の処理については、適宜説明を省略する。

40

【 0 0 8 3 】

まずステップ S 9 1 において、T T L の値が 1 に設定 ( 初期設定 ) される。ステップ S 9 2 において、宛先ノードのアドレスが設定される。この場合、エンドホスト 1 1 3 のアドレスが設定される。設定されたアドレスは、処理が終了するまで、Request パケットの送信先のアドレスとされる。ステップ S 9 3 において、Reply パケット優先モードにするか否かが判断され、優先モードにすると判断された場合、ステップ S 9 4 に処理が進められ、パケットヘッダに、優先制御フラグが立てられた Request パケットが生成され、ステップ S 9 5 に処理が進められる。

【 0 0 8 4 】

一方ステップ S 9 3 において、Reply パケット優先モードにしないと判断された場合、

50

ステップS 9 4 の処理はスキップされ、ステップS 9 5 に処理が進められる。ステップS 9 5 において、Requestパケットが生成される。その生成されるRequestパケットのパケットヘッダにセットされるT T Lは、その時点で設定されている値とされる。ステップS 9 1 から処理が来た場合、T T L = 1 に設定されているため、T T L が 1 に設定されたRequestパケットが生成される。

**【 0 0 8 5 】**

ステップS 9 6 において、生成されたRequestパケットが宛先ノードに送信され、その送信した時刻が記録される。そして、ステップS 9 7 において、所定の回数、同一のノードに対して、Requestパケットが送信されたか否かが判断される。この場合、宛先ノードは、エンドホスト1 1 3 であり、固定とされているため、エンドホスト1 1 3 に対してRequestパケットが送信されることになるが、T T L として設定されている値が同一の値とされている間、エンドホスト1 1 1 とエンドホスト1 1 3 との2点間の通信経路上にある所定のノード1 1 2 に対してRequestパケットが送信されることになる。よって、所定の回数、同一のノードに対して、Requestパケットを送信することができる。

10

**【 0 0 8 6 】**

ステップS 9 7 において、所定の回数、同一のノードに対して、Requestパケットが送信されてはいないと判断された場合、ステップS 9 3 に処理が戻され、それ以降の処理が繰り返され、送信されたと判断された場合、ステップS 9 8 に処理が進められる。ステップS 9 3 乃至S 9 7 の処理が繰り返されることにより、同一のノードに複数回、Requestパケットが送信されることになる。

20

**【 0 0 8 7 】**

ステップS 9 8 において、宛先ノードからのReplyパケットを受信したか否かが判断される。この場合、宛先ノードは、エンドホスト1 1 3 であるため、エンドホスト1 1 3 からのReplyパケットを受信されたか否かが判断される。なお、エンドホスト1 1 3 以外のノードからのReplyパケットは、宛先到達不能通知パケットであるため、そのような宛先到達不能通知パケットではないか否かを判断することで、ステップS 1 6 の処理が行われるようにしても良い。宛先ノードからのReplyパケットは受信していないと判断されるのは、エンドホスト1 1 3 にRequestパケットが到達していないときである。

**【 0 0 8 8 】**

ステップS 9 8 において、宛先ノードからのReplyパケットは受信していないと判断された場合、ステップS 9 9 に処理が進められ、T T L の値が1 だけ増加される。ステップS 9 9 において、T T L の値が1 だけ増加されると、ステップS 9 3 に処理が戻され、それ以降の処理が繰り返される。ステップS 9 3 乃至9 9 の処理が繰り返されることにより、エンドホスト1 1 1 に近い側のノードから順次、Requestパケットが複数回送信されることになる。

30

**【 0 0 8 9 】**

このようなTrace Routeが用いられて送信元ホスト( エンドホスト1 1 1 ) の送信に係わる処理が実行された場合、受信に係わる処理は、図6 のフローチャートの処理に基づき行われる。すなわち、Trace Routeが用いられた場合と、Trace Routeが用いられていない場合とでは、送信時の処理は異なるが、受信時の処理は同一の処理とされる。すなわち、Trace Routeが用いられた場合であっても、同一のノードから、複数回、Replyパケット( 宛先到達不能通知パケット ) 受信されるため、各ノードにおけるR T T の遅延時間に関する確率密度関数を算出することができる。よって、受信時の処理として、同一の処理を行えば、所定のリンクの帯域使用率を推定することができる。

40

**【 0 0 9 0 】**

Trace Routeが用いられた場合、中継ノードにおける処理が、Trace Routeが用いられていない場合と比較して異なる処理を含むため、次に、図1 0 を参照して、Trace Routeが用いられたときの中継ノードの処理について説明する。中継ノードは、図4 において、ノード1 1 2 であり、図1 0 のフローチャートの処理は、ノード1 1 2 の送受信処理部1 6 1 が行う。

50

## 【 0 0 9 1 】

ステップ S 1 1 1 において、Request パケットが受信される。ステップ S 1 1 2 において、受信された Request パケットに含まれる T T L の値が 1 だけ減算された値とされる。その結果、T T L が 0 になったか否かがステップ S 1 1 3 において判断される。ステップ S 1 1 3 において、T T L が 0 にはなっていないと判断された場合、ステップ S 1 1 4 に処理が進められ、受信されたパケットは、通常処理される。すなわち、ステップ S 1 1 5 において、次ノードに転送される処理が実行される。

## 【 0 0 9 2 】

一方、ステップ S 1 1 3 において、T T L は 0 になったと判断された場合、ステップ S 1 1 6 に処理が進められる。ステップ S 1 1 6 において、宛先到達不能通知パケットが生成される。ステップ S 1 1 7 において、受信された Request パケットのパケットヘッダに優先制御フラグが立っているか否かが判断される。ステップ S 1 1 7 において、優先制御フラグが立っていると判断された場合、ステップ S 1 1 8 に処理が進められる。ステップ S 1 1 8 において、宛先到達不能通知パケットのパケットヘッダに、優先制御フラグが立てられる。

10

## 【 0 0 9 3 】

ステップ S 1 1 8 において、優先フラグが立てられた宛先到達不能通知パケットが生成された場合、または、ステップ S 1 1 7 において、優先制御フラグは立てられていないと判断された場合、ステップ S 1 1 9 において、送信元ホストに対して、宛先到達不能通知パケットが送信される。そして、宛先到達不能通知パケットが送信されたときには、受信された Request パケットはステップ S 1 2 0 において破棄される。

20

## 【 0 0 9 4 】

このように、Trace Route が適用されているときに、Request パケットを受信した中継ノードは、T T L の値が 0 になったときに、宛先到達不能通知パケットを生成し、送信元ホストに送信するといった処理を行う。宛先到達不能通知パケットには、宛先到達不能通知パケットを生成した中継ノードの IP アドレスが含まれるため、そのような宛先到達不能通知パケットを受信した送信元ホストにおいては、中継ノードの IP アドレスを知ることができる。よって、宛先ノードまでの通信経路上に存在する中継ノードの IP アドレスを調べながら、R T T を得ることができ、リンク毎の帯域使用率などを推定することが可能となる。

30

## 【 0 0 9 5 】

なお、上述した説明においては、通信経路上のノードのアドレスが不明なネットワークに対して、T T L の値を順次増やすことで、IP アドレスを順次調べ、同一の値の T T L を複数回送信することで、R T T の確率密度関数を求め、所定のリンクの帯域使用率を求めるとして説明した。

## 【 0 0 9 6 】

他の実施の形態として、通信経路上のノードの IP アドレスが不明なネットワークに対して、上記したように T T L の値を順次増やすことで、中継ノードの IP アドレスを順次調べ、経路上の全てのノードの IP アドレスを調べ終わった後に、その調べ上げた IP アドレスを用いて、IP アドレスがわかっているネットワークに対する処理、すなわち、図 5 乃至図 8 のフローチャートの処理がホストやノードにおいて行われることで、R T T の確率密度関数が求められ、所定のリンクの帯域使用率が求められるようにしても良い。

40

## 【 0 0 9 7 】

本発明を適用することで、Ping をエンドホストや通信経路上に存在する各ノードに送るといった簡便な処理だけで、各ノードに対する R T T の遅延時間の確率密度関数が取得でき、それを逆畳み込み演算することで、通信経路中の各リンクの帯域使用率を推定することができる。すなわち、本発明によれば、Ping という簡便なものを用いて、通信経路中の各リンクの帯域使用率を推定することができる。

## 【 0 0 9 8 】

[ 記録媒体について ]

50

上述した一連の処理は、ハードウェアにより実行することもできるし、ソフトウェアにより実行することもできる。一連の処理をソフトウェアにより実行する場合には、そのソフトウェアを構成するプログラムが、コンピュータにインストールされる。ここで、コンピュータには、専用のハードウェアに組み込まれているコンピュータや、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどが含まれる。

【0099】

図11は、上述した一連の処理をプログラムにより実行するコンピュータのハードウェアの構成例を示すブロック図である。コンピュータにおいて、CPU (Central Processing Unit) 201、ROM (Read Only Memory) 202、RAM (Random Access Memory) 203は、バス204により相互に接続されている。バス204には、さらに、入出力インタフェース205が接続されている。入出力インタフェース205には、入力部206、出力部207、記憶部208、通信部209、およびドライブ210が接続されている。

10

【0100】

入力部206は、キーボード、マウス、マイクロフォンなどよりなる。出力部207は、ディスプレイ、スピーカなどよりなる。記憶部208は、ハードディスクや不揮発性のメモリなどよりなる。通信部209は、ネットワークインタフェースなどよりなる。ドライブ210は、磁気ディスク、光ディスク、光磁気ディスク、または半導体メモリなどのリムーバブルメディア211を駆動する。

20

【0101】

以上のように構成されるコンピュータでは、CPU201が、例えば、記憶部208に記憶されているプログラムを、入出力インタフェース205およびバス204を介して、RAM203にロードして実行することにより、上述した一連の処理が行われる。

【0102】

コンピュータ (CPU201) が実行するプログラムは、例えば、パッケージメディア等としてのリムーバブルメディア211に記録して提供することができる。また、プログラムは、ローカルエリアネットワーク、インターネット、デジタル衛星放送といった、有線または無線の伝送媒体を介して提供することができる。

30

【0103】

コンピュータでは、プログラムは、リムーバブルメディア211をドライブ210に装着することにより、入出力インタフェース205を介して、記憶部208にインストールすることができる。また、プログラムは、有線または無線の伝送媒体を介して、通信部209で受信し、記憶部208にインストールすることができる。その他、プログラムは、ROM202や記憶部208に、予めインストールしておくことができる。

【0104】

なお、コンピュータが実行するプログラムは、本明細書で説明する順序に沿って時系列に処理が行われるプログラムであっても良いし、並列に、あるいは呼び出しが行われたとき等の必要なタイミングで処理が行われるプログラムであっても良い。

【0105】

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

40

【0106】

なお、本発明の実施の形態は、上述した実施の形態に限定されるものではなく、本発明の要旨を逸脱しない範囲において種々の変更が可能である。

【符号の説明】

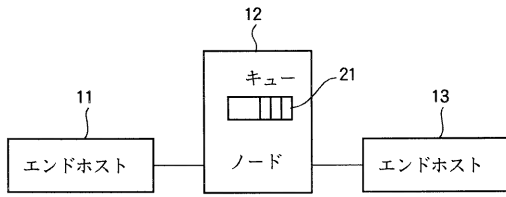
【0107】

111 エンドホスト、 112 ノード、 113 エンドホスト、 121 キュー、 131 ノード、 141 キュー、 151 送信処理部、 152 受信処理部、 161 送受信処理部、 171 送受信処理部

50

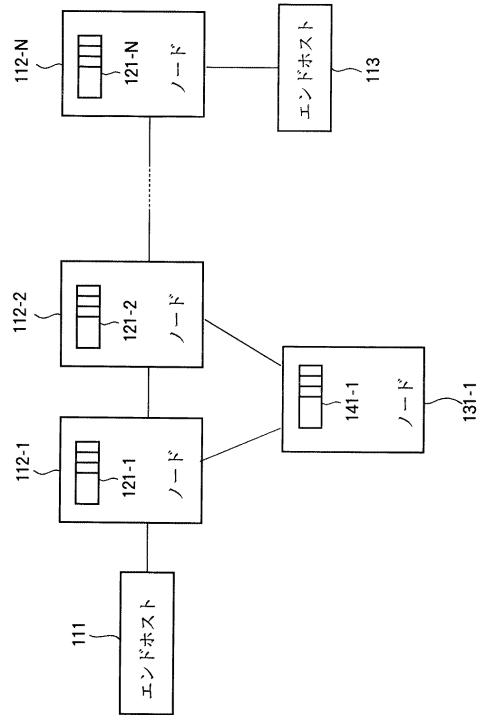
【 図 1 】

図 1



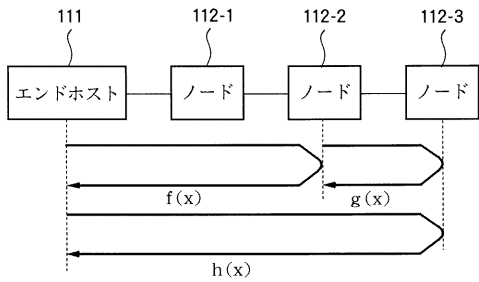
【 図 2 】

図 2



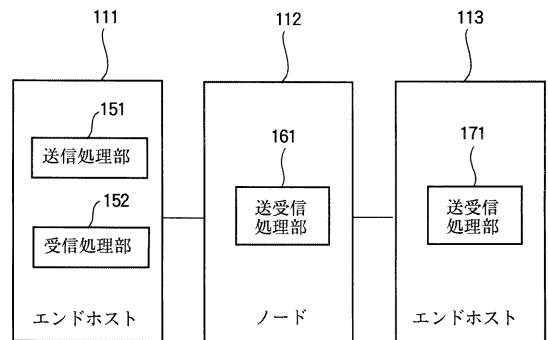
【 図 3 】

図 3



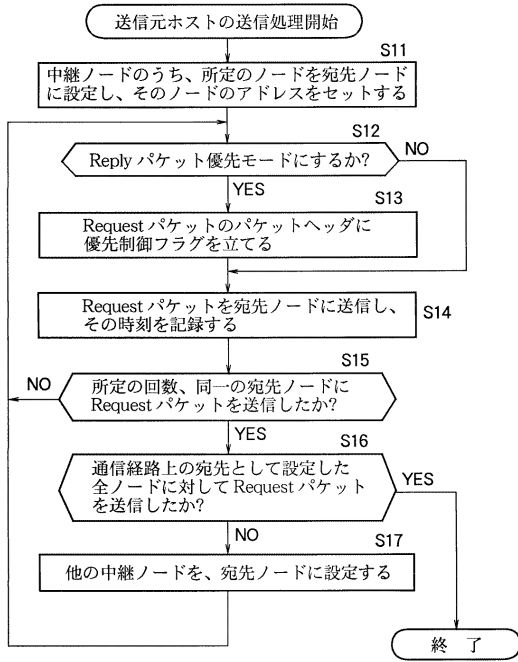
【 図 4 】

図 4



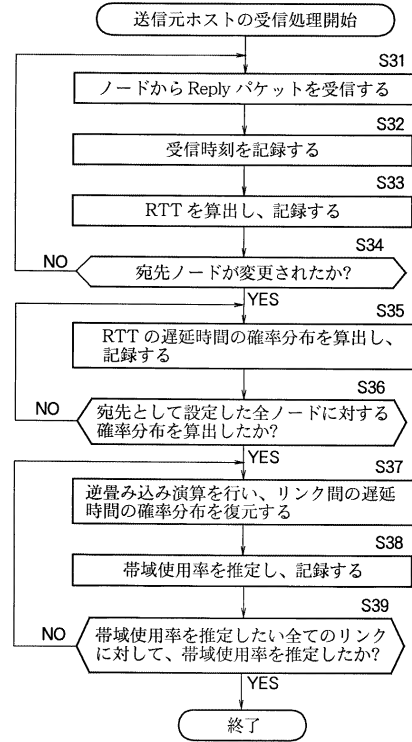
【 図 5 】

図 5



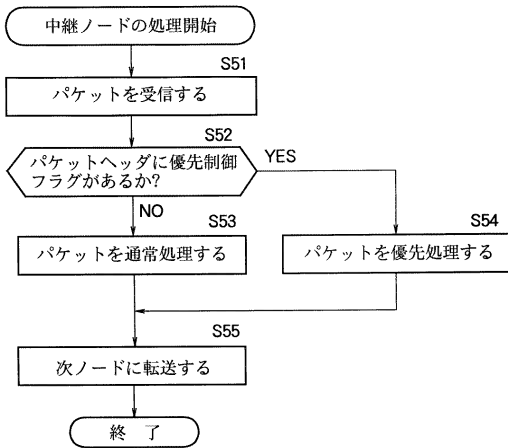
【 図 6 】

図 6



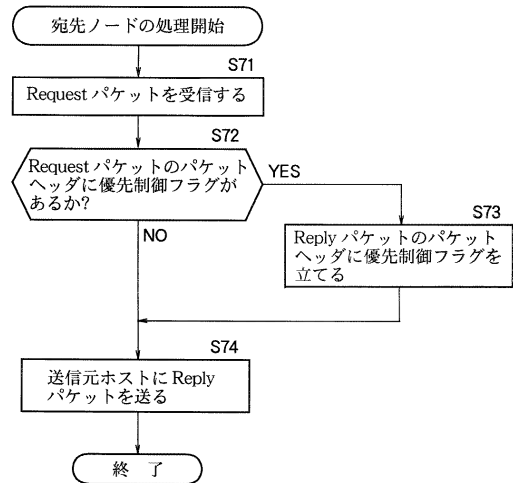
【 図 7 】

図 7



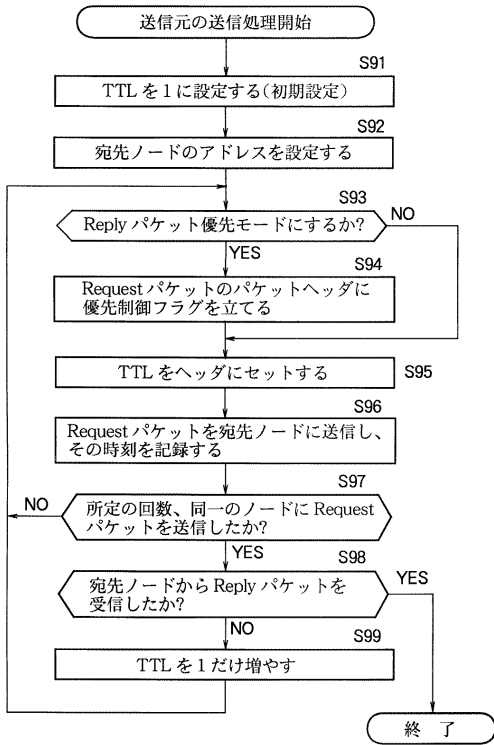
【 図 8 】

図 8



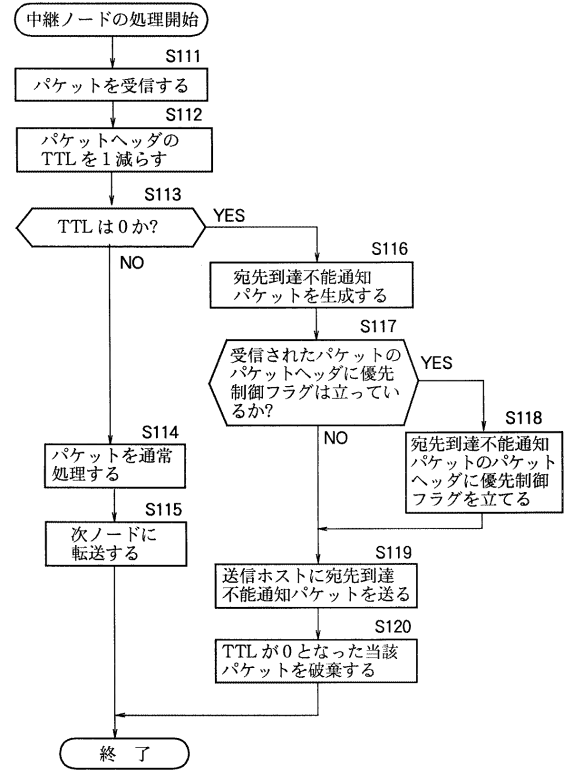
【 図 9 】

図 9



【 図 1 0 】

図 10



【 図 1 1 】

図 11

