

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6135078号
(P6135078)

(45) 発行日 平成29年5月31日(2017.5.31)

(24) 登録日 平成29年5月12日(2017.5.12)

(51) Int.Cl.		F I	
H O 4 L	1/16	(2006.01)	H O 4 L 1/16
H O 4 L	29/08	(2006.01)	H O 4 L 13/00 3 O 7 Z

請求項の数 11 (全 23 頁)

(21) 出願番号	特願2012-203427 (P2012-203427)	(73) 特許権者	000005223
(22) 出願日	平成24年9月14日 (2012.9.14)		富士通株式会社
(65) 公開番号	特開2014-60533 (P2014-60533A)		神奈川県川崎市中原区上小田中4丁目1番1号
(43) 公開日	平成26年4月3日 (2014.4.3)	(74) 代理人	100103528
審査請求日	平成27年5月12日 (2015.5.12)		弁理士 原田 一男
		(72) 発明者	佐野 健
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		(72) 発明者	池本 健太郎
			神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内
		審査官	谷岡 佳彦

最終頁に続く

(54) 【発明の名称】 到達確認メッセージについての情報処理方法及び情報処理装置

(57) 【特許請求の範囲】

【請求項 1】

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを確認するためのビット列に写像し、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する

処理を、コンピュータに実行させるためのプログラムであって、

前記ビット列が、確率的データ構造であり、

前記写像する処理が、

前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっているか否かを検証する処理と、

前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっている場合には、前記集合に含まれる識別子を削減してから前記ビット列を生成し直す処理と、

を含むプログラム。

【請求項 2】

前記写像する処理が、

前記検証する処理と前記生成し直す処理とを、前記検証する処理において前記識別子の範囲に含まれるが前記集合には含まれない識別子が前記集合に含まれていると判断されることがなくなるまで繰り返す処理

10

20

をさらに含む請求項 1 記載のプログラム。

【請求項 3】

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを確認するためのビット列に写像し、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する

処理を、コンピュータに実行させるためのプログラムであって、

前記ビット列が、確率的データ構造であり、

前記写像する処理は、

前記集合に含まれる識別子の範囲に含まれる識別子の数に対する前記集合に含まれる識別子の数の比率が、一定値以上となるように前記集合に含まれる識別子を特定する処理と

10

、
前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっているか否かを検証する処理と

を含むプログラム。

【請求項 4】

前記特定する処理が、

前記集合に含まれる識別子のうち最後の識別子を削除する処理、又は前記集合に含まれる識別子の間隔のうち最大の間隔が生じている部分で前記集合を分割する処理

を含む請求項 3 記載のプログラム。

20

【請求項 5】

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを確認するためのビット列に写像し、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する

処理を、コンピュータに実行させるためのプログラムであって、

前記ビット列が、確率的データ構造であり、

前記写像する処理が、

前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっているか否かを検証する処理と、

30

前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっている場合には、前記集合に含まれる識別子を削減してから前記ビット列を生成し直す処理と、

を含み、

前記生成し直す処理が、

前記集合に含まれる識別子のうち最後の識別子を削除する処理、前記集合に含まれる識別子のうち先頭の識別子を削除した場合における新たな集合に含まれる識別子の第 1 の範囲と前記集合に含まれる識別子のうち最後の識別子を削除した場合における新たな集合に含まれる識別子の第 2 の範囲とのうち狭い範囲を生じさせる識別子を削除する処理、又は、前記集合に含まれる識別子の間隔のうち最も広い間隔以降の識別子を削除する処理

40

を含むプログラム。

【請求項 6】

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを確認するためのビット列に写像し、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する

処理を、コンピュータが実行する到着確認メッセージ送信方法であって、

前記ビット列が、確率的データ構造であり、

前記写像する処理が、

前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子

50

が、前記集合に含まれていることになっているか否かを検証する処理と、

前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっている場合には、前記集合に含まれる識別子を削減してから前記ビット列を生成し直す処理と、

を含む到着確認メッセージ送信方法。

【請求項 7】

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを検証するためのビット列に写像し、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する

処理を、コンピュータが実行する到着確認メッセージ送信方法であって、

前記ビット列が、確率的データ構造であり、

前記写像する処理は、

前記集合に含まれる識別子の範囲に含まれる識別子の数に対する前記集合に含まれる識別子の数の比率が、一定値以上となるように前記集合に含まれる識別子を特定する処理と、

前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっているか否かを検証する処理と

を含む到着確認メッセージ送信方法。

【請求項 8】

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを検証するためのビット列に写像し、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する

処理を、コンピュータが実行する到着確認メッセージ送信方法であって、

前記ビット列が、確率的データ構造であり、

前記写像する処理が、

前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっているか否かを検証する処理と、

前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっている場合には、前記集合に含まれる識別子を削減してから前記ビット列を生成し直す処理と、

を含み、

前記生成し直す処理が、

前記集合に含まれる識別子のうち最後の識別子を削除する処理、前記集合に含まれる識別子のうち先頭の識別子を削除した場合における新たな集合に含まれる識別子の第 1 の範囲と前記集合に含まれる識別子のうち最後の識別子を削除した場合における新たな集合に含まれる識別子の第 2 の範囲とのうち狭い範囲を生じさせる識別子を削除する処理、又は、前記集合に含まれる識別子の間隔のうち最も広い間隔以降の識別子を削除する処理

を含む到着確認メッセージ送信方法。

【請求項 9】

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを検証するためのビット列に写像するビット列生成部と、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する送信部と、

を有する情報処理装置であって、

前記ビット列が、確率的データ構造であり、

前記ビット列生成部が、

前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっているか否かを検証し、

10

20

30

40

50

前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっている場合には、前記集合に含まれる識別子を削減してから前記ビット列を生成し直す

情報処理装置。

【請求項 10】

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを確認するためのビット列に写像するビット列生成部と、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する送信部と、

を有する情報処理装置であって、

前記ビット列が、確率的データ構造であり、

前記ビット列生成部は、

前記集合に含まれる識別子の範囲に含まれる識別子の数に対する前記集合に含まれる識別子の数の比率が、一定値以上となるように前記集合に含まれる識別子を特定し、

前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっているか否かを検証する

情報処理装置。

【請求項 11】

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを確認するためのビット列に写像するビット列生成部と、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する送信部と、

を有する情報処理装置であって、

前記ビット列が、確率的データ構造であり、

前記ビット列生成部が、

前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっているか否かを検証する処理と、

前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっている場合には、前記集合に含まれる識別子を削減してから前記ビット列を生成し直す処理とを実行し、

前記ビット列生成部は、前記生成し直す処理において、

前記集合に含まれる識別子のうち最後の識別子を削除する処理、前記集合に含まれる識別子のうち先頭の識別子を削除した場合における新たな集合に含まれる識別子の第1の範囲と前記集合に含まれる識別子のうち最後の識別子を削除した場合における新たな集合に含まれる識別子の第2の範囲とのうち狭い範囲を生じさせる識別子を削除する処理、又は、前記集合に含まれる識別子の間隔のうち最も広い間隔以降の識別子を削除する処理

を実行する情報処理装置。

【発明の詳細な説明】

【技術分野】

【0001】

本技術は、通信ノード間の通信パスが不確定なネットワークの通信技術に関する。

【背景技術】

【0002】

モバイル端末や固定端末（これらについては、以下通信ノード、又は単にノードと呼ぶ）間でデータの送受信を行い、各ノードが受信したデータを蓄積、運搬（モバイル端末のみ）、転送を行うようなネットワークが考案されている。このように、送信元ノードと宛先ノードとの間の通信パスが常に存在するとは限らないような、すなわち通信パスが不確定な通信環境においてもデータ転送が可能となるネットワークは、遅延耐性ネットワーク（Delay Tolerant Network）又は途絶耐性ネットワーク（DTN：Disruption-Tolerant Network）と呼ばれる。

10

20

30

40

50

【 0 0 0 3 】

D T Nにおいては、送信元ノードから宛先ノードまでのEnd-to-Endのデータ転送にかかる時間が長く、またパケットロスや通信回線の切断が頻繁に発生する等の劣悪な通信環境を含むことがあり、このような通信環境においても効率的にデータ転送を行うことが求められている。

【 0 0 0 4 】

D T Nにおいては、移動可能な移動ノードや移動不可能な固定ノードがデータの転送を繰り返すことで送信元ノードから宛先ノードへデータを届けることになる。具体的には、データを受信したノードは、転送先となるノードが近くにいない場合、その受信データを一旦自身の記憶領域に蓄積する。その後、自ノード又は他ノードの移動により、他のノードが近接することでデータの転送を行う。このようにEnd-to-Endの通信パスが常に存在するとは限らないような通信環境においても、データ転送が可能となる。

10

【 0 0 0 5 】

D T N上のノードはデータ転送を行うために、他のノードから受信したデータを自身の記憶領域（ハードディスクドライブ、メモリなど）に一時的にキャッシュとして蓄積する。しかしながら、ノードの記憶領域の容量には制限があり、新たに送信元ノードから発生するデータを蓄積及び転送するためには、宛先ノードに到達したデータのキャッシュを消去することが好ましい。そのため、データの宛先ノードは、データの到達を送信元ノード及び転送ノードに対して伝えるために、到達確認メッセージ、すなわちA C KメッセージをD T N上で伝送することが好ましい。一方で、データが宛先ノードに到達する度にA C Kメッセージを送信すると、D T N上に大量のA C Kメッセージが伝播することになる。これは、通信効率上好ましくない。

20

【 0 0 0 6 】

図1に示すように、インターネットなどの通常の通信環境においては、左端のサーバから右端のサーバへデータを送信する場合、左端のサーバは、当該データを含むパケットにシーケンス番号を付与して送信する。このようなネットワークでは、途中でパケットが無くなったり大幅に遅延したりすることはあまりないので、右端のサーバは、連続して受信したパケットに付与されたシーケンス番号のうち最大のシーケンス番号を含むA C Kパケットを、左端のサーバに送信する。例えば、最大のシーケンス番号が「4」であれば、「4」を含むA C Kパケットを送信する。左端のサーバは、A C Kパケットを受信すると、当該A C Kパケットに含まれるシーケンス番号以下のシーケンス番号についてのパケットのデータを破棄する。シーケンス番号「4」を含むA C Kパケットの場合には、0乃至4のシーケンス番号に係るデータを破棄することになる。

30

【 0 0 0 7 】

D T Nでは、途中でパケットが無くなったり大幅に遅延したりする場合があるので、このような技術は、連続してパケットを受信できない場合がある。例えば、シーケンス番号「0」「2」「4」といったように、とびとびのシーケンス番号のパケットを受信する場合もある。受信したパケットのシーケンス番号が連続していないので、最大のシーケンス番号は「0」となり、A C Kパケットを集約できない。

【 先行技術文献 】

40

【 特許文献 】

【 0 0 0 8 】

【 特許文献 1 】 特開 2 0 1 1 - 4 3 3 3 号 公 報

【 特許文献 2 】 特開 2 0 1 1 - 1 7 2 1 9 6 号 公 報

【 非特許文献 】

【 0 0 0 9 】

【 非特許文献 1 】 「DTN：遅延と仲良くするネットワーク」、通信ソサイエティマガジンNo.16 [春号]2011

【 非特許文献 2 】 「DTN技術の現状と展望」、通信ソサイエティマガジンNo.16 [春号]201

【非特許文献3】Wing Ho Yuen, et. al, "IMPROVING SEARCH EFFICIENCY USING BLOOM FILTER IN PARTIALLY CONNECTED AD HOC NETWORKS, A NODE-CENTRIC ANALYSIS", Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE

【発明の概要】

【発明が解決しようとする課題】

【0010】

従って、本技術の目的は、一側面として、通信ノード間の通信パスが不確定なネットワークにおいて効率よく到達確認メッセージを送信できるようにするための技術を提供することである。

【課題を解決するための手段】

【0011】

本技術の第1の態様に係る到着確認メッセージ送信方法は、(A)ある送信元ノードから受信されたデータの識別子の集合を、識別子が上記集合に含まれているか否かを確認するためのビット列に写像する処理と、(B)上記集合に含まれる識別子の範囲を規定するデータと上記ビット列とを含む到着確認メッセージを、ある送信元ノード宛に送信する処理とを含む。

【0012】

本技術の第2の態様に係るデータ削除方法は、(C)ある送信元ノードから受信されたデータの識別子の集合の範囲を規定するデータと、識別子が上記集合に含まれているか否かを確認するためのビット列とを含む到着確認メッセージを受信する処理と、(D)ある送信元ノードについてのデータ及び当該データの識別子とを格納するデータ格納部から、到着確認メッセージに含まれる上記データから特定される範囲に含まれる識別子を抽出する処理と、(E)上記ビット列に基づき、抽出された識別子が、上記集合に含まれているか否かを判断する処理と、(F)上記集合に含まれていると判断された識別子及び当該識別子についてのデータを、データ格納部から削除する処理とを含む。

【発明の効果】

【0013】

通信ノード間の通信パスが不確定なネットワークにおいて効率よく到達確認メッセージを送信できるようになる。

【図面の簡単な説明】

【0014】

【図1】図1は、通常のネットワークにおけるACKメッセージの送信について説明するための図である。

【図2】図2は、DTNにおけるデータパケットの伝送の様子を模式的に示す図である。

【図3】図3は、DTNにおけるACKパケットの伝送の様子を模式的に示す図である。

【図4】図4は、本技術の実施の形態の概要を説明する図である。

【図5】図5は、送信元ノードとして機能するノードの機能ブロック図である。

【図6】図6は、転送ノードとして機能するノードの機能ブロック図である。

【図7】図7は、宛先ノードとして機能するノードの機能ブロック図である。

【図8】図8は、宛先ノードの受信部で実行される処理の処理フローを示す図である。

【図9】図9は、宛先ノードのデータ格納部に格納されるデータの一例を示す図である。

【図10】図10は、宛先ノードで実行される処理の処理フローを示す図である。

【図11】図11は、第1の前処理の処理フローを示す図である。

【図12】図12は、第2の前処理の処理フローを示す図である。

【図13】図13は、ブルームフィルタを説明するための図である。

【図14】図14は、ブルームフィルタを説明するための図である。

【図15】図15は、誤検出検証処理の処理フローを示す図である。

【図16】図16は、誤検出検証処理について説明するための図である。

【図17】図17は、ACKパケットの構成例を示す図である。

【図18】図18は、転送ノード及び送信元ノードに格納されるデータの一例を示す図で

10

20

30

40

50

ある。

【図 19】図 19 は、転送ノード及び送信元ノードにおいて A C K パケットを受信した場合の処理の処理フローを示す図である。

【図 20】図 20 は、転送ノード及び送信元ノードにおいて A C K パケットを受信した場合の処理の処理フローを示す図である。

【図 21】図 21 は、ノードである情報処理装置の機能ブロック図である。

【発明を実施するための形態】

【0015】

図 2 は、D T N におけるデータパケット（又はデータメッセージ）の伝送の様子を例示する図である。図 2 の例では、左端に示された固定の送信元ノード 1 0 0 1 から右端に示された固定の宛先ノード 1 0 0 7 へデータを送信する場合を示している。この例では、移動可能な転送ノード 1 0 0 2 が、送信元ノード 1 0 0 1 に接近してきたところで、送信元ノード 1 0 0 1 は、データパケットを転送ノード 1 0 0 2 に送信し、転送ノード 1 0 0 2 は、データパケットを受信してキャッシュに蓄積する。転送ノード 1 0 0 2 の付近に他の転送ノードがなければ、データパケットの転送が行われないが、例えば転送ノード 1 0 0 2 が移動して付近に固定の転送ノード 1 0 0 3 や移動可能な転送ノード 1 0 0 5 が近接すると、転送ノード 1 0 0 2 は、キャッシュに格納されているデータパケットを転送する。転送ノード 1 0 0 3 は、受信したデータパケットをキャッシュし、移動可能な転送ノード 1 0 0 4 が近づいてくると、当該転送ノード 1 0 0 4 にキャッシュ内のデータパケットを転送する。また、転送ノード 1 0 0 5 は、転送ノード 1 0 0 2 からデータパケットを受信すると、当該データパケットをキャッシュして移動する。そして、転送ノード 1 0 0 5 は、宛先ノード 1 0 0 7 と通信可能な程度に近づくと、キャッシュしているデータパケットを宛先ノード 1 0 0 7 に送信する。このようにして、送信元ノード 1 0 0 1 から宛先ノード 1 0 0 7 までデータパケットが転送されることになる。

【0016】

また、図 3 は、D T N における A C K パケット（又は A C K メッセージ）の伝送の様子を例示する図である。基本的には図 2 の逆方向に送信元ノード 1 0 0 1 宛に A C K パケットを送信することになるが、必ずしも同じノードが用いられるわけではない。宛先ノード 1 0 0 7 は、以下で詳細に述べる A C K パケットを、その際に近隣に存在する移動可能な転送ノード 1 0 0 5 に送信する。転送ノード 1 0 0 5 は、以下で述べるように A C K パケットに含まれるデータに基づきキャッシュからデータを削除する処理を実行する。さらに転送ノード 1 0 0 5 は移動して、固定の転送ノード 1 0 0 3 や移動可能な転送ノード 1 1 0 3 の付近にやってくる。そうすると、転送ノード 1 0 0 5 は、A C K パケットを、転送ノード 1 0 0 3 及び 1 1 0 3 に送信する。

【0017】

転送ノード 1 0 0 3 は、A C K パケットを受信すると、以下で述べるように A C K パケットに含まれるデータに基づきキャッシュからデータを削除する処理を実行する。また、移動可能な転送ノード 1 1 0 2 が近づいてくると、転送ノード 1 0 0 3 は、A C K パケットを転送ノード 1 1 0 2 に転送する。また、転送ノード 1 1 0 3 は、A C K パケットを受信した後、移動を開始する。そして、転送ノード 1 1 0 3 は、送信元ノード 1 0 0 1 に近づいて通信可能な状態になると、A C K パケットを転送する。そうすると、送信元ノード 1 0 0 1 も、以下で述べるように A C K パケットに含まれるデータに基づきキャッシュからデータを削除する処理を実行する。

【0018】

このような処理を様々なノード間で実施することで、D T N 上でデータ通信が行われるようになる。なお、実施の形態では説明を簡単にするため、送信元ノード、転送ノード、宛先ノードの区別をつけて説明するが、いずれのノードも送信元ノード、転送ノード又は宛先ノードとして機能する。

【0019】

なお、本実施の形態は、図 4 に模式的に示すように、左端の送信元ノードから、シーケ

10

20

30

40

50

ンシャルな識別番号「0」乃至「9」のデータパケットを、右端の宛先ノードまで送信したが、宛先ノードでは識別番号「0」「2」及び「4」のデータパケットしか受信できていないような場合に有効である。すなわち、この場合識別番号「0」「2」及び「4」のデータパケットが受信済みであることを表し且ついずれの識別番号についてのデータパケットを受信したのかを確認するためのビット列を付加したACKパケットを、宛先ノードが送信元ノードへ送信する。これによって、ACKパケットの転送経路上の転送ノード及び送信元ノードにおいて、いずれのデータパケットを破棄してもよいのかを確認できるようになる。

【0020】

以下、本実施の形態においてDTNに含まれるノードの構成及び当該ノードが実行する処理について説明する。

【0021】

図5は、送信元ノードとして機能するノード100の機能ブロック図を示す。送信元ノード100は、通信インタフェース101と、送信部102と、受信部103と、データ生成部104と、ID生成部105と、データ削除処理部106と、データ格納部107とを有する。

【0022】

通信インタフェース101は、例えば無線通信処理を行うモジュールである。データ生成部104は、他のノードに送信すべきデータを生成する。生成されるデータは、例えば、位置センサなどのセンサの出力データや処理のログデータ等である。

【0023】

送信部102は、データ生成部104によって生成された、他のノードに送信すべきデータからパケットを生成して通信インタフェース101を介して送信する。ID生成部105は、例えば送信部102と連携して、宛先ノードの識別子毎に、送信するパケットの識別子(ID)であるシーケンシャルな識別番号を順次生成して、送信部102に出力する。送信部102は、宛先ノードの識別子及び識別番号に対応付けて、送信したデータをデータ格納部107に格納する。なお、データ生成部104が、ID生成部105と連携して、宛先ノードの識別子毎に識別番号を生成させ、データと共にデータ格納部107に格納するようにしても良い。

【0024】

受信部103は、通信インタフェース101を介して他のノードからパケットを受信し、データ格納部107に格納し、以下で述べるようにACKパケットを受信した場合にはデータ削除処理部106に処理を実行させる。データ削除処理部106は、受信部103が受信したACKパケットに含まれるデータを用いて、データ格納部107に格納されているデータのうち削除可能なデータを特定して削除する。

【0025】

また、図6は、転送ノードとして機能するノード200の機能ブロック図を示す。転送ノード200は、通信インタフェース201と、送信部202と、受信部203と、転送処理部204と、データ削除処理部205と、データ格納部206とを有する。

【0026】

通信インタフェース201は、例えば無線通信処理を行うモジュールである。送信部202は、転送処理部204からの指示に応じてデータ格納部206に格納されているデータパケットを、他のノードに対して送信する。受信部203は、他のノードからデータパケット及びACKパケットを受信すると、データ格納部206に格納する。データパケットは、宛先ノードの識別子及び識別番号に関連付けてデータ格納部206において管理される。なお、ACKパケットについては転送すれば削除する。

【0027】

転送処理部204は、通信インタフェース201を介して他のノードが近接していることを検出すると、データ格納部206に格納されているデータパケット及びACKパケットを送信部202に送信させる。データ削除処理部205は、受信部203が受信したA

10

20

30

40

50

C K パケットに含まれるデータを用いて、データ格納部 206 に格納されているデータのうち削除可能なデータを特定して削除する。送信元ノード 100 におけるデータ削除処理部 106 と、転送ノード 200 のデータ削除処理部 205 とは、同様な処理を実行する。

【0028】

図 7 は、宛先ノードとして機能するノード 300 の機能ブロック図である。宛先ノード 300 は、通信インタフェース 301 と、送信部 302 と、受信部 303 と、ビット列生成部 304 と、検証処理部 305 と、データ格納部 306 とを有する。

【0029】

通信インタフェース 301 は、例えば無線通信処理を行うモジュールである。送信部 302 は、ACK パケット等を通信インタフェース 301 を介して送信する。受信部 303 は、データパケットを通信インタフェース 301 を介して受信し、データ格納部 306 に格納する。

【0030】

ビット列生成部 304 は、送信元ノード毎に、受信したデータパケットの識別番号がビット列生成時に用いられたか否かを確認するためのビット列を生成する。また、検証処理部 305 は、ビット列生成部 304 により生成されたビット列の検証処理を実行する。

【0031】

次に、図 8 乃至図 17 を用いて、宛先ノード 300 における処理について説明する。宛先ノード 300 の受信部 303 は、データパケットの受信を待機しており（図 8：ステップ S1）、データパケットを受信すると、データパケットのデータ部分をデータ格納部 306 に格納する（ステップ S2）。また、受信部 303 は、当該データパケットから当該データパケットの送信元ノードの識別子を特定する（ステップ S3）。

【0032】

そして、受信部 303 は、送信元ノードの未処理パケット数のカウンタを 1 インクリメントすると共に（ステップ S5）、データパケットに含まれるデータの識別子を抽出して、データ格納部 306 に格納する（ステップ S6）。データ格納部 306 には、例えば図 9 に示すようなデータが送信元ノード毎に格納される。図 9 の例では、未処理パケット数のカウンタ値と、未処理の識別番号群とが格納されるようになっている。

【0033】

このような処理を、例えばユーザによる終了指示などの理由で処理終了となるまで繰り返す（ステップ S7）。

【0034】

これに対してビット列生成部 304、検証処理部 305 及び送信部 302 が実行する処理について図 10 乃至図 17 を用いて説明する。

【0035】

ビット列生成部 304 は、データ格納部 306 にデータが格納されている送信元ノードのうち、例えばラウンドロビン式に 1 つの送信元ノードを選択する（図 10：ステップ S11）。そうすると、ビット列生成部 304 は、選択された送信元ノードについてデータ格納部 306 に格納されている未処理パケット数のカウンタ値が閾値以上となっているか判断する（ステップ S13）。閾値以上の個数のデータパケットを受信してまとめて集約して ACK パケットを送信するためである。

【0036】

選択された送信元ノードについて未処理パケット数のカウンタ値が閾値以上とない場合には、処理はステップ S11 に戻る。一方、選択された送信元ノードについて未処理パケット数のカウンタ値が閾値以上であれば、ビット列生成部 304 は、前処理を実行する（ステップ S15）。前処理については、図 11 及び図 12 を用いて説明する。前処理は、本実施の形態では、後に実行される誤検出検証処理の処理効率を向上させるために実行される。

【0037】

まず、ビット列生成部 304 は、データ格納部 306 に格納されている未処理パケット

10

20

30

40

50

の識別番号を集合 S に設定する (図 11 : ステップ S 31)。そして、ビット列生成部 304 は、集合 S に含まれる識別番号の範囲に対する集合 S に含まれる識別番号の数の割合を算出する (ステップ S 33)。集合 S に、識別番号 { 1, 3, 5, 9, 11 } が含まれる場合、識別番号の範囲は 10 (= 11 - 1) であり、集合 S に含まれる識別番号の数は 5 であるから、割合は 50 % (= 5 / 10) となる。

【 0038 】

そして、ビット列生成部 304 は、算出された割合が閾値を超えているか判断する (ステップ S 35)。閾値は例えば 50 % 以上の所定の数値である。後に実行される誤検出検証処理では、(100 % - 算出された割合) の数の識別番号について誤検出するか否かを検証するので、集合 S に含まれる識別番号の範囲に含まれる識別番号の過半数が誤検出検証処理の処理対象となるのは非効率だからである。

10

【 0039 】

算出された割合が閾値を超えていない場合には、ビット列生成部 304 は、集合 S から、識別番号最大値を除外する (ステップ S 37)。そして処理はステップ S 33 に戻る。

【 0040 】

一方、算出された割合が閾値を超えている場合には、ビット列生成部 304 は、集合 S に含まれる識別番号を、ビット列生成対象の識別番号に設定する (ステップ S 39)。そして、呼出元の処理に戻る。

【 0041 】

この前処理は、例えば図 12 に示すような処理に変更してもよい。ビット列生成部 304 は、データ格納部 306 に格納されている未処理パケットの識別番号を集合に設定する (ステップ S 41)。そして、ビット列生成部 304 は、未処理の集合を 1 つ選択する (ステップ S 43)。

20

【 0042 】

その後、ビット列生成部 304 は、選択された集合に含まれる識別番号の範囲に対する、選択された集合に含まれる識別番号の数を算出する (ステップ S 45)。演算内容はステップ S 33 と同様である。

【 0043 】

そして、ビット列生成部 304 は、算出された割合が閾値を超えたか判断する (ステップ S 47)。算出された割合が閾値を超えていない場合には、ビット列生成部 304 は、選択された集合に含まれる識別番号間の間隔が最大となる部分で、選択された集合を分割する (ステップ S 49)。そして処理はステップ S 43 に戻る。なお、分割によって生成された集合のうち、あまりに包含される識別番号の数が少ない集合については、処理対象から外すようにしてもよい。

30

【 0044 】

一方、算出された割合が閾値を超えている場合には、ビット列生成部 304 は、選択された集合に含まれる識別番号を、ビット列生成対象に設定する (ステップ S 51)。そして、ビット列生成部 304 は、未処理の集合が存在するか判断する (ステップ S 53)。未処理の集合が存在する場合には、処理はステップ S 43 に戻る。一方、未処理の集合が存在しない場合には、処理は呼出元の処理に戻る。

40

【 0045 】

本前処理によれば、ビット列生成対象の識別番号の集合が複数存在するので、以下の処理は、集合毎に実行することになる。

【 0046 】

図 10 の処理の説明に戻って、ビット列生成部 304 は、任意の識別番号が、ビット列生成対象の集合に含まれているか否かを確認するためのビット列を生成する (ステップ S 17)。本実施の形態では、ビット列として確率的データ構造、具体的にはブルームフィルタ (Bloom Filter) を採用する。

【 0047 】

ブルームフィルタは、よく知られているが、図 13 及び図 14 を用いて簡単に説明して

50

おく。ブルームフィルタを生成するためには、図 13 に示すように、k 個のハッシュ関数を用意し、識別番号をそれぞれに入力してハッシュ値を取得する。図 13 の例では、ハッシュ関数 1 からはハッシュ値「8」が生成され、ハッシュ関数 2 からはハッシュ値「1」が生成され、ハッシュ関数 k からはハッシュ値「11」が生成されたものとする。そうすると、m ビット配列において、ハッシュ値の位置のビットを「1」にセットする。図 13 の例では、「8」ビット目、「1」ビット目、「11」ビット目を「1」にセットする。このようにして生成されたビット列がブルームフィルタである。k 個のハッシュ関数は、DTN において共通である。

【0048】

さらに、図 14 に示すように、ビット列生成対象の集合に含まれる他の識別番号についても、k 個のハッシュ関数にそれぞれ入力してハッシュ値を取得する。図 14 の例では、ハッシュ関数 1 からハッシュ値「0」が生成され、ハッシュ関数 2 からハッシュ値「1」が生成され、ハッシュ関数 k からハッシュ値「8」が生成されたものとする。そうすると、図 13 で生成されたブルームフィルタの m ビット配列において、ハッシュ値の位置のビットを「1」にセットする。図 14 の例では、「0」ビット目、「1」ビット目、「8」ビット目に「1」をセットする。但し、「1」及び「8」ビット目は既に「1」がセットされているので変更はない。このように、複数の識別番号についてのビット操作を、重ねて同じ m ビット配列に対して実施することになる。

【0049】

図 10 の処理の説明に戻って、このようにビット列生成部 304 により生成されたビット列は検証処理部 305 に出力されて、検証処理部 305 は、誤検出検証処理を実行する（ステップ S19）。誤検出検証処理については、図 15 及び図 16 を用いて説明する。

【0050】

検証処理部 305 は、ビット列生成に用いられた識別番号の範囲に含まれ且つビット列生成に用いられなかった特定識別番号のうち未処理の識別番号を 1 つ特定する（図 15：ステップ S61）。識別番号 {1, 3, 5, 9, 11} がビット列生成に用いられた場合には、識別番号 1 乃至 11 の範囲において {2, 4, 6, 7, 8, 10} が特定識別番号となる。

【0051】

そして、検証処理部 305 は、ブルームフィルタの誤検出の有無を判定する（ステップ S63）。ブルームフィルタは、その性質から誤検出が発生しうることが知られており、この誤検出を許容してしまうと、再送すべきデータパケットが、通信経路上の転送ノード及び送信元ノードにおいて削除されてしまう。そこで、誤検出が発生するか否かをここで検証しておく。

【0052】

例えば図 16 に示すように、特定識別番号を、ブルームフィルタ生成時と同じ k 個のハッシュ関数に入力してハッシュ値を取得する。ハッシュ値が示すビット位置の値が全て「1」であれば、特定識別番号はブルームフィルタ生成時の集合に含まれていたことになるので、誤検出と判断される。図 16 の例では、ハッシュ関数 1 のハッシュ値が「3」であり、ハッシュ関数 2 のハッシュ値が「10」であり、ハッシュ関数 k のハッシュ値が「4」であり、これらのビット位置のビット値は、全て「1」となっているので、誤検出と判断される。一方、ハッシュ値が示すビット位置の値が 1 つでも「0」となっている場合には、誤検出無しと判断される。

【0053】

ステップ S63 の結果として誤検出ありとされた場合には（ステップ S65：Yes ルート）、ビット列生成部 304 に誤検出ありを出力する（ステップ S67）。そして、処理は呼出元の処理に戻る。一方、ステップ S63 の結果として誤検出無しとされた場合には（ステップ S65：No ルート）、検証処理部 305 は、特定識別番号に未処理の識別番号が存在するか判断する（ステップ S69）。未処理の識別番号が存在する場合には、処理はステップ S61 に戻る。一方、未処理の識別番号が存在しない場合には、生成され

10

20

30

40

50

たブルームフィルタについて誤検出無しということになるので、検証処理部 305 は、ビット列生成部 304 に、誤検出無しを出力する（ステップ S71）。そして処理を終了する。

【0054】

このような処理を実行することで、生成されたビット列、すなわちブルームフィルタをそのまま使用できるか否かが判定される。

【0055】

図10の処理の説明に戻って、ビット列生成部 304 は、検証処理部 305 からの出力に基づき、誤検出があったか否かを判断する（ステップ S21）。誤検出が存在していた場合には、ビット列生成部 304 は、ビット列生成対象の識別番号の調整を実行する（ステップ S29）。 10

【0056】

この識別番号の調整については、様々な方法が可能である。但し、どの識別番号をビット列生成対象から外せば誤検出が無くなるのかは直接的には不明なので、（A）ビット列生成対象の識別番号のうち最大の識別番号を削除する方法、（B）ビット列生成対象の識別番号のうち最小の識別番号と最大の識別番号のうち、除外した場合にビット列生成対象の識別番号の範囲が狭くなる識別番号を削除する方法、（C）ビット列生成対象の識別番号の間隔を算出し、最も広い部分から後ろの識別番号を削除する方法などを採用しうる。

【0057】

その後処理はステップ S17 に戻って再度ビット列を生成し直した上で、誤検出検証処理を実施する。 20

【0058】

一方、誤検出がなかった場合には、ビット列生成部 304 は、生成したビット列と、ビット列生成対象の識別番号の範囲を規定するデータ（例えば先頭識別番号及び末尾識別番号の組み合わせ。但し、先頭識別番号と末尾識別番号までのオフセット値との組み合わせ）を含む ACK パケットを生成して、送信部 302 に送信させる（ステップ S23）。

【0059】

例えば図17に示すような ACK パケットのデータ構成例を示す。図17の例では、パケットタイプとして到達確認（ACK）が指定されており、送信元ノードの識別子と、先頭識別番号と、末尾識別番号と、ビット列であるブルームフィルタとが含まれている。 30

【0060】

そして、ビット列生成部 304 は、処理終了が指示された等の理由で処理終了するか判断する（ステップ S25）。処理終了ではない場合には、ビット列生成部 304 は、処理に係る送信元ノードについてデータ格納部 306 において、未処理パケット数のカウンタ値を、ビット列生成に用いた識別子の個数分だけ減分し、ビット列生成対象の識別番号を削除する（ステップ S27）。そして処理はステップ S11 に戻る。

【0061】

以上のような処理を実行することで、識別番号の誤検出無しを保証した上で多数のデータパケットの ACK パケットを集約した ACK パケットを1つ生成して送信できるようになる。すなわち、通信効率が向上する。 40

【0062】

次に、転送ノード及び送信元ノードにおいて ACK パケットを受信した際の処理について図18乃至図20を用いて説明する。なお、送信元ノード及び転送ノードにおいては、図18に示すようなデータを、データ格納部に格納しているものとする。図18の例では、データパケットのデータだけではなく、未処理の識別番号群を、送信元ノード毎（送信元ノードについては自ノード分のみ）に保持しているものとする。なお、ここでは、転送ノード 200 を一例に説明するが、送信元ノード 100 についてもデータ削除処理部 106 が同様の処理を実行することになる。

【0063】

まず、転送ノード 200 のデータ削除処理部 205 は、受信部 203 から ACK パケッ 50

トを受信するまで待機する（図19：ステップS81）。ACKパケットを受信すると、データ削除処理部205は、ACKパケットからデータの送信元ノードの識別子、先頭及び末尾識別番号、そしてビット列を抽出する（ステップS83）。

【0064】

また、データ削除処理部205は、抽出された送信元ノードの識別子についてデータ格納部206に格納されている未処理パケットの識別番号を抽出する（ステップS85）。図18に示すようなデータを読み出す。

【0065】

その後、データ削除処理部205は、抽出された識別番号のうち未処理の識別番号を1つ選択する（ステップS87）。そして、データ削除処理部205は、ACKパケットに含まれるビット列に基づき、選択された識別番号の包含判定を実行する（ステップS89）。ブルームフィルタの場合、k個のハッシュ関数のハッシュ値が示すビット位置の値が全て「1」であれば、ブルームフィルタ生成時の識別番号集合に包含されていたことが確認できる。従って、本ステップでも、k個のハッシュ関数に、選択された識別番号を入力してハッシュ値を取得し、当該ハッシュ値が示すビット位置の値を確認し、全てのビット値が「1」であれば、包含と判定する。そして処理は端子Aを介して図20の処理に移行する。

10

【0066】

図20の処理の説明に移行して、データ削除処理部205は、選択された識別番号は包含されていると判定されたか判断する（ステップS91）。包含されていないと判定された場合には、処理はステップS95に移行する。一方、包含されていると判断された場合には、データ削除処理部205は、データ格納部206において、選択された識別番号のパケットのデータを削除する（ステップS93）。到達確認ができた分だけキャッシュの容量を有効活用できるようになる。

20

【0067】

そして、データ削除処理部205は、未処理の識別番号が存在するか判断する（ステップS95）。未処理の識別番号が存在する場合には、端子Bを介してステップS87に戻る。一方、未処理の識別番号が存在しない場合には、処理終了が明示的に指示されるまで、上で述べた処理を繰り返す（ステップS87）。すなわち、処理終了でない場合には端子Cを介してステップS81に戻る。

30

【0068】

以上のような処理を実施することで、宛先ノードにデータパケットが到着したことを確認した上で、保持しているデータパケットを破棄できるようになる。これによってキャッシュが有効活用できるようになる。

【0069】

上で述べた実施の形態では、ビット列としてブルームフィルタを用いる例を示した。ブルームフィルタを用いることで、多数のデータパケットを集約できるが、ブルームフィルタの性質から誤検出防止の処理を行うことになる。例えば、ACKパケットに含めることができるビット列のビット長に相当するデータパケットを集約すればよい場合には、ブルームフィルタを用いずとも良い。

40

【0070】

例えば、ビット長がmビットであれば、最大mデータパケットを集約できる。例えば、先頭識別番号「11」で末尾識別番号「23」であって、ビット列生成対象の識別番号が{11, 13, 15, 17, 19, 21, 23}であれば、各識別番号について先頭識別番号からのオフセット位置に相当するビット位置を「1」にセットした「101010101010101」というビット列を採用するようにしても良い。すなわち、ビット列生成処理は、ビット列生成対象の識別番号から、先頭識別番号からのオフセット位置に相当するビット位置のビット値によってビット列生成対象の集合に包含されているか否かを確認するための所定長のビット列への写像となる。なお、ブルームフィルタを用いる場合においても、ビット列生成対象の識別番号からブルームフィルタに相当するビット列への写像と

50

も言える。

【 0 0 7 1 】

この場合には、誤検出検証処理及びビット列生成対象の識別番号を調整する処理については行わずに済む。また、転送ノード及び送信元ノードについても、ビット列におけるビット位置が「 1 」か否かだけで包含判定ができるので、処理が簡易になる。

【 0 0 7 2 】

以上本技術の実施の形態を説明したが、本技術はこれに限定されるものではない。例えば、機能ブロック図は、プログラムモジュール構成とは一致しない場合もある。データ構造例も一例にすぎず、同様の内容を保持する他のデータ構造を採用しても良い。

【 0 0 7 3 】

さらに、処理フローについても処理結果が変わらない限りにおいて処理ステップの順番を入れ替えたり、処理を並列実行するようにしても良い。

【 0 0 7 4 】

なお、上で述べた送信元ノード 1 0 0、転送ノード 2 0 0、宛先ノード 3 0 0 は、コンピュータ装置であって、図 2 1 に示すように、メモリ 2 5 0 1 と CPU (Central Processing Unit) 2 5 0 3 とハードディスク・ドライブ (HDD: Hard Disk Drive) 2 5 0 5 と表示装置 2 5 0 9 に接続される表示制御部 2 5 0 7 とリムーバブル・ディスク 2 5 1 1 用のドライブ装置 2 5 1 3 と入力装置 2 5 1 5 とネットワークに接続するための通信制御部 2 5 1 7 とがバス 2 5 1 9 で接続されている。オペレーティング・システム (OS: Operating System) 及び本実施例における処理を実施するためのアプリケーション・プログラムは、HDD 2 5 0 5 に格納されており、CPU 2 5 0 3 により実行される際には HDD 2 5 0 5 からメモリ 2 5 0 1 に読み出される。CPU 2 5 0 3 は、アプリケーション・プログラムの処理内容に応じて表示制御部 2 5 0 7、通信制御部 2 5 1 7、ドライブ装置 2 5 1 3 を制御して、所定の動作を行わせる。また、処理途中のデータについては、主としてメモリ 2 5 0 1 に格納されるが、HDD 2 5 0 5 に格納されるようにしてもよい。本技術の実施例では、上で述べた処理を実施するためのアプリケーション・プログラムはコンピュータ読み取り可能なリムーバブル・ディスク 2 5 1 1 に格納されて頒布され、ドライブ装置 2 5 1 3 から HDD 2 5 0 5 にインストールされる。インターネットなどのネットワーク及び通信制御部 2 5 1 7 を経由して、HDD 2 5 0 5 にインストールされる場合もある。このようなコンピュータ装置は、上で述べた CPU 2 5 0 3、メモリ 2 5 0 1 などのハードウェアと OS 及びアプリケーション・プログラムなどのプログラムとが有機的に協働することにより、上で述べたような各種機能を実現する。

【 0 0 7 5 】

以上述べた本実施の形態をまとめると、以下ようになる。

【 0 0 7 6 】

本実施の形態の第 1 の態様に係る到着確認メッセージ送信方法は、(A) ある送信元ノードから受信されたデータの識別子の集合を、識別子が上記集合に含まれているか否かを確認するためのビット列に写像する処理と、(B) 上記集合に含まれる識別子の範囲を規定するデータと上記ビット列とを含む到着確認メッセージを、ある送信元ノード宛に送信する処理とを含む。

【 0 0 7 7 】

このようにすれば到着確認メッセージを集約することができ、通信効率を上げることができる。なお、メッセージは、どの階層レベルのメッセージ又はパケットであってもよい。

【 0 0 7 8 】

さらに、上で述べたビット列は、確率的データ構造であってもよい。そして、上で述べた写像する処理が、(a 1) ビット列に基づき、上記識別子の範囲に含まれるが上記集合には含まれない識別子が、上記集合に含まれていることになっているか否かを検証する処理と、(a 2) 上記識別子の範囲に含まれるが上記集合には含まれない識別子が、上記集合に含まれていることになっている場合には、上記集合に含まれる識別子を削減してから

10

20

30

40

50

上記ビット列を生成し直す処理とを含むようにしても良い。より多くのデータメッセージ又はデータパケットの集約を行うことができるようになる。

【 0 0 7 9 】

また、上で述べた写像する処理が、(a 3) 上記検証する処理と上記生成し直す処理とを、上記検証する処理において上記識別子の範囲に含まれるが上記集合には含まれない識別子が上記集合に含まれていると判断されることが無くなるまで繰り返す処理をさらに含むようにしても良い。

【 0 0 8 0 】

また、上で述べた写像する処理は、上記集合に含まれる識別子の範囲に含まれる識別子の数に対する上記集合に含まれる識別子の数の比率が、一定値以上となるように上記集合に含まれる識別子を特定する処理を含むようにしても良い。例えば、集約度合いを一定以上にすることで通信効率を上げ、また上記検証する処理を効率化するためである。

【 0 0 8 1 】

さらに、上で述べた特定する処理が、上記識別子の範囲に含まれる識別子のうち最後の識別子を削除する処理、又は上記識別子の範囲に含まれる識別子の間隔のうち最大の間隔が生じている部分で上記集合を分割する処理を含むようにしてもよい。

【 0 0 8 2 】

さらに、上で述べた生成し直す処理が、上記識別子の範囲に含まれる識別子のうち最後の識別子を削除する処理、上記識別子の範囲に含まれる識別子のうち先頭の識別子を削除した場合における新たな集合に含まれる識別子の第 1 の範囲と上記識別子の範囲に含まれる識別子のうち最後の識別子を削除した場合における新たな集合に含まれる識別子の第 2 の範囲とのうち狭い範囲を生じさせる識別子を削除する処理、又は、上記識別子の範囲に含まれる識別子の間隔のうち最も広い間隔以降の識別子を削除する処理を含むようにしても良い。

【 0 0 8 3 】

本技術の第 2 の態様に係るデータ削除方法は、(C) ある送信元ノードから受信されたデータの識別子の集合の範囲を規定するデータと、識別子が上記集合に含まれているか否かを確認するためのビット列とを含む到着確認メッセージを受信する処理と、(D) ある送信元ノードについてのデータ及び当該データの識別子とを格納するデータ格納部から、到着確認メッセージに含まれる上記データから特定される範囲に含まれる識別子を抽出する処理と、(E) 上記ビット列に基づき、抽出された識別子が、上記集合に含まれているか否かを判断する処理と、(F) 上記集合に含まれていると判断された識別子及び当該識別子についてのデータを、データ格納部から削除する処理とを含む。

【 0 0 8 4 】

このようにすればデータ格納部の容量を有効活用することができるようになる。

【 0 0 8 5 】

なお、上記処理をコンピュータに行わせるためのプログラムを作成することができ、当該プログラムは、例えばフレキシブルディスク、CD - ROM、光磁気ディスク、半導体メモリ、ハードディスク等のコンピュータ読み取り可能な記憶媒体又は記憶装置に格納される。尚、中間的な処理結果はメインメモリ等の記憶装置に一時保管される。

【 0 0 8 6 】

以上の実施例を含む実施形態に関し、さらに以下の付記を開示する。

【 0 0 8 7 】

(付記 1)

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを確認するためのビット列に写像し、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する

処理を、コンピュータに実行させるためのプログラム。

【 0 0 8 8 】

10

20

30

40

50

(付記2)

前記ビット列が、確率的データ構造であり、
前記写像する処理が、

前記ビット列に基づき、前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっているか否かを検証する処理と、

前記識別子の範囲に含まれるが前記集合には含まれない識別子が、前記集合に含まれていることになっている場合には、前記集合に含まれる識別子を削減してから前記ビット列を生成し直す処理と、

を含む付記1記載のプログラム。

【0089】

10

(付記3)

前記写像する処理が、

前記検証する処理と前記生成し直す処理とを、前記検証する処理において前記識別子の範囲に含まれるが前記集合には含まれない識別子が前記集合に含まれていると判断されることが無くなるまで繰り返す処理

をさらに含む付記2記載のプログラム。

【0090】

(付記4)

前記写像する処理は、

前記集合に含まれる識別子の範囲に含まれる識別子の数に対する前記集合に含まれる識別子の数の比率が、一定値以上となるように前記集合に含まれる識別子を特定する処理
を含む付記1乃至3のいずれか1つ記載のプログラム。

20

【0091】

(付記5)

前記特定する処理が、

前記識別子の範囲に含まれる識別子のうち最後の識別子を削除する処理、又は前記識別子の範囲に含まれる識別子の間隔のうち最大の間隔が生じている部分で前記集合を分割する処理

を含む付記4記載のプログラム。

【0092】

30

(付記6)

前記生成し直す処理が、

前記識別子の範囲に含まれる識別子のうち最後の識別子を削除する処理、前記識別子の範囲に含まれる識別子のうち先頭の識別子を削除した場合における新たな集合に含まれる識別子の第1の範囲と前記識別子の範囲に含まれる識別子のうち最後の識別子を削除した場合における新たな集合に含まれる識別子の第2の範囲とのうち狭い範囲を生じさせる識別子を削除する処理、又は、前記識別子の範囲に含まれる識別子の間隔のうち最も広い間隔以降の識別子を削除する処理

を含む付記2記載のプログラム。

【0093】

40

(付記7)

ある送信元ノードから受信されたデータの識別子の集合の範囲を規定するデータと、識別子が前記集合に含まれているか否かを確認するためのビット列とを含む到着確認メッセージを受信し、

前記ある送信元ノードについてのデータ及び当該データの識別子とを格納するデータ格納部から、前記到着確認メッセージに含まれる前記データから特定される範囲に含まれる識別子を抽出し、

前記ビット列に基づき、抽出された前記識別子が、前記集合に含まれているか否かを判断し、

前記集合に含まれていると判断された識別子及び当該識別子についてのデータを、前

50

記データ格納部から削除する

処理を、コンピュータに実行させるためのプログラム。

【0094】

(付記8)

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを確認するためのビット列に写像し、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する

処理を、コンピュータが実行する到着確認メッセージ送信方法。

【0095】

10

(付記9)

ある送信元ノードから受信されたデータの識別子の集合の範囲を規定するデータと、識別子が前記集合に含まれているか否かを確認するためのビット列とを含む到着確認メッセージを受信し、

前記ある送信元ノードについてのデータ及び当該データの識別子とを格納するデータ格納部から、前記到着確認メッセージに含まれる前記データから特定される範囲に含まれる識別子を抽出し、

前記ビット列に基づき、抽出された前記識別子が、前記集合に含まれているか否かを判断し、

前記集合に含まれていると判断された識別子及び当該識別子についてのデータを、前記データ格納部から削除する

20

処理を、コンピュータが実行するデータ削除方法。

【0096】

(付記10)

ある送信元ノードから受信されたデータの識別子の集合を、識別子が前記集合に含まれているか否かを確認するためのビット列に写像するビット列生成部と、

前記集合に含まれる識別子の範囲を規定するデータと前記ビット列とを含む到着確認メッセージを、前記ある送信元ノード宛に送信する送信部と、

を有する情報処理装置。

【0097】

30

(付記11)

ある送信元ノードから受信されたデータの識別子の集合の範囲を規定するデータと、識別子が前記集合に含まれているか否かを確認するためのビット列とを含む到着確認メッセージを受信する受信部と、

前記ある送信元ノードについてのデータ及び当該データの識別子とを格納するデータ格納部から、前記到着確認メッセージに含まれる前記データから特定される範囲に含まれる識別子を抽出し、前記ビット列に基づき、抽出された前記識別子が、前記集合に含まれているか否かを判断し、前記集合に含まれていると判断された識別子及び当該識別子についてのデータを、前記データ格納部から削除する削除部と、

を有する情報処理装置。

40

【符号の説明】

【0098】

100 送信元ノード

101 通信インタフェース

102 送信部

103 受信部

104 データ生成部

105 ID生成部

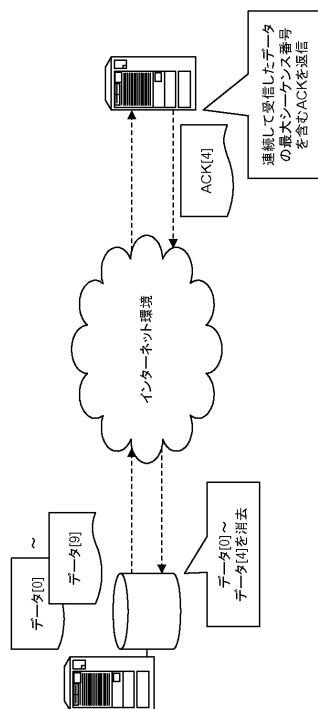
106 データ削除処理部

107 データ格納部

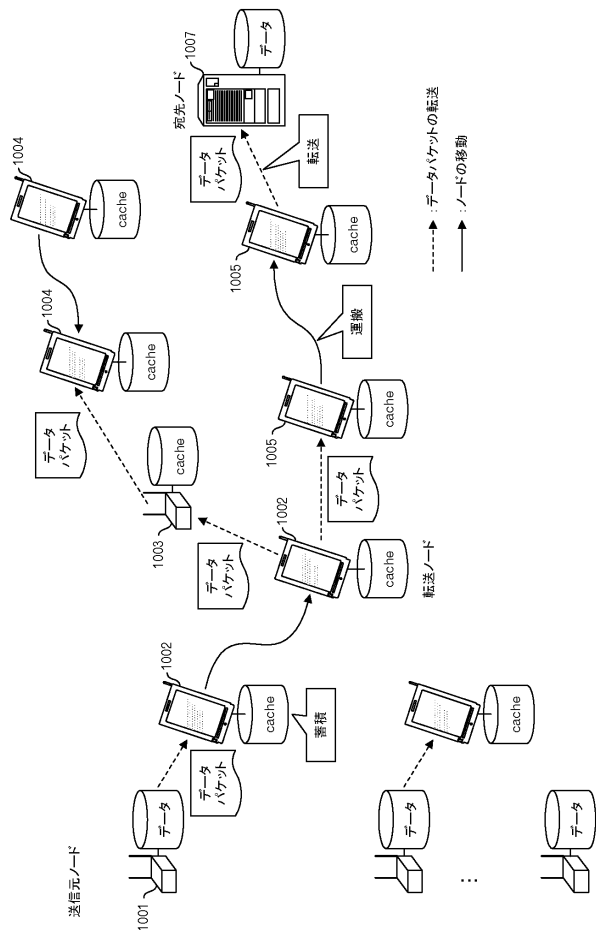
50

- | | |
|-------|-----------|
| 2 0 0 | 転送ノード |
| 2 0 1 | 通信インタフェース |
| 2 0 2 | 送信部 |
| 2 0 3 | 受信部 |
| 2 0 4 | 転送処理部 |
| 2 0 5 | データ削除処理部 |
| 2 0 6 | データ格納部 |
| 3 0 0 | 宛先ノード |
| 3 0 1 | 通信インタフェース |
| 3 0 2 | 送信部 |
| 3 0 3 | 受信部 |
| 3 0 4 | ビット列生成部 |
| 3 0 5 | 検証処理部 |
| 3 0 6 | データ格納部 |

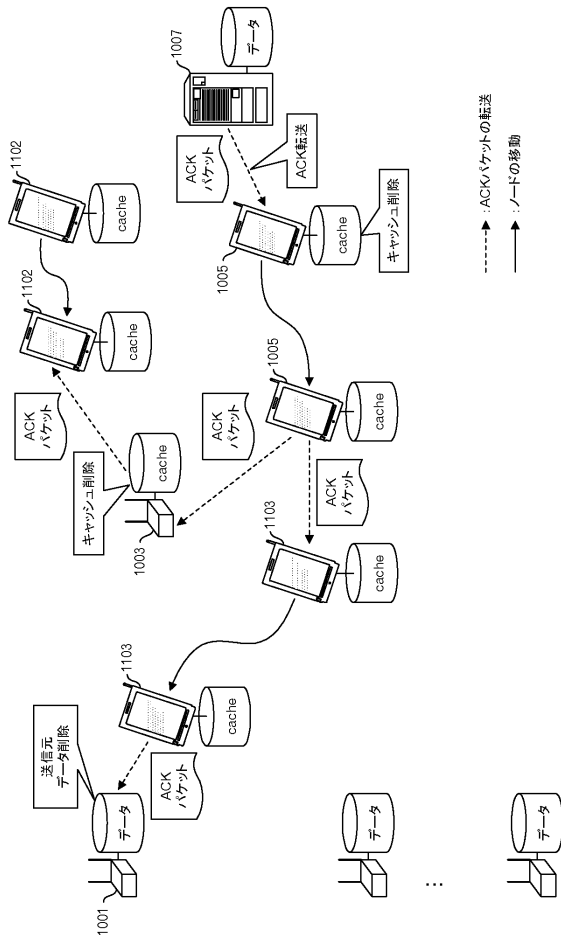
【 図 1 】



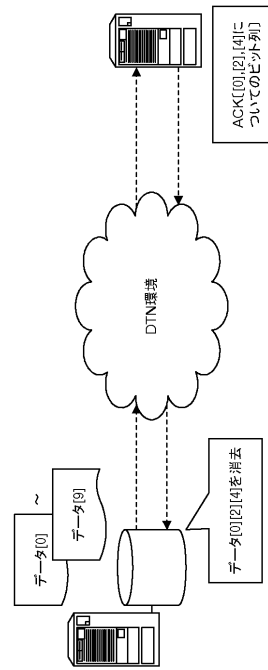
【 図 2 】



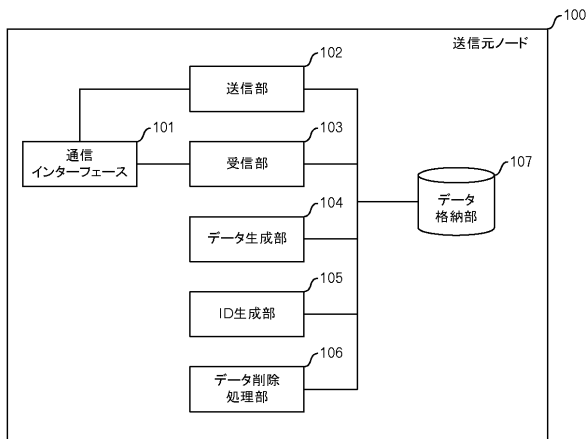
【図 3】



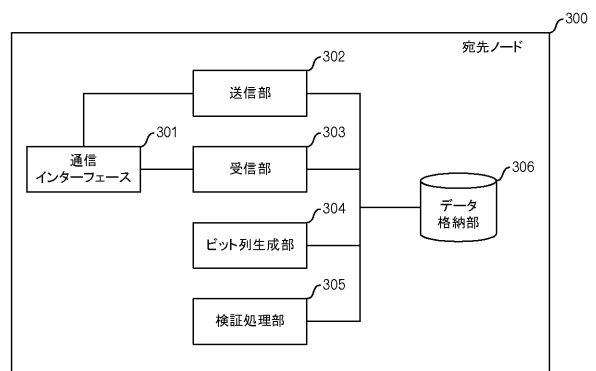
【図 4】



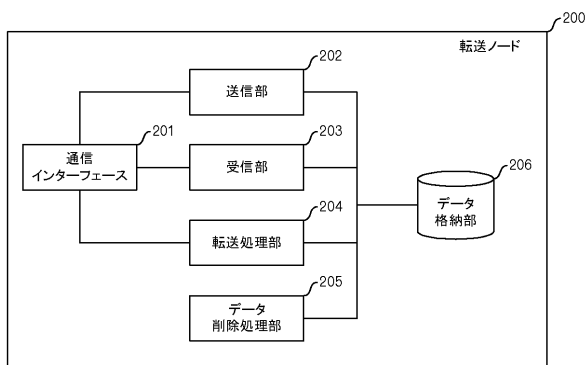
【図 5】



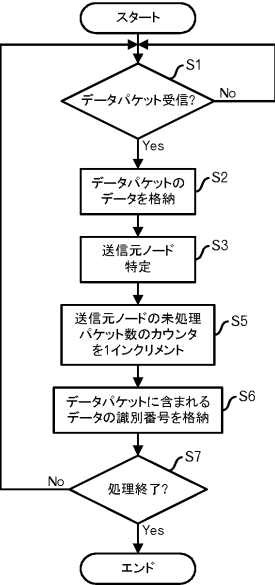
【図 7】



【図 6】



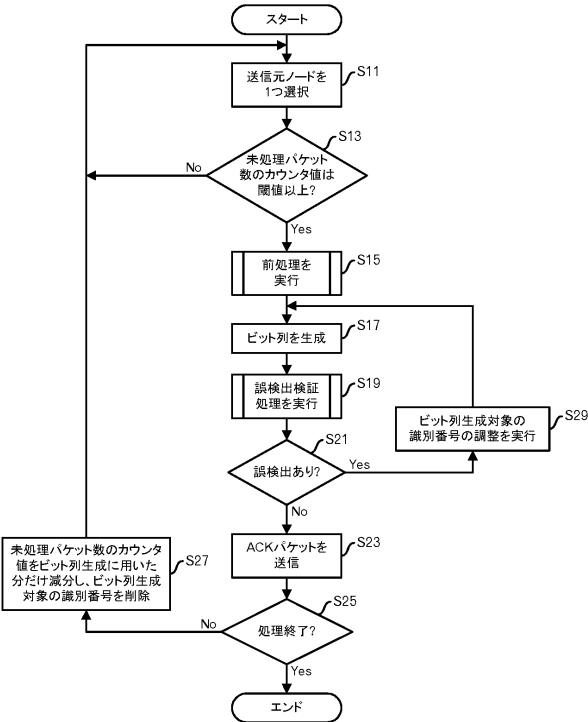
【図 8】



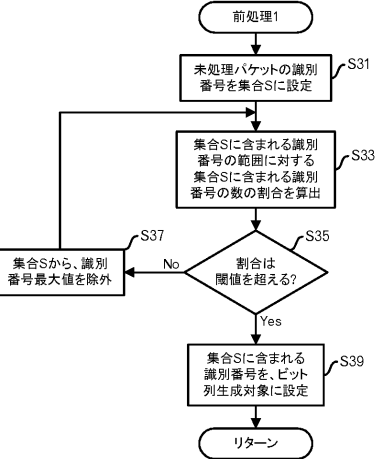
【図 9】

カウンタ	X
未処理識別番号	$z_1, z_2, z_3, \dots, z_m$

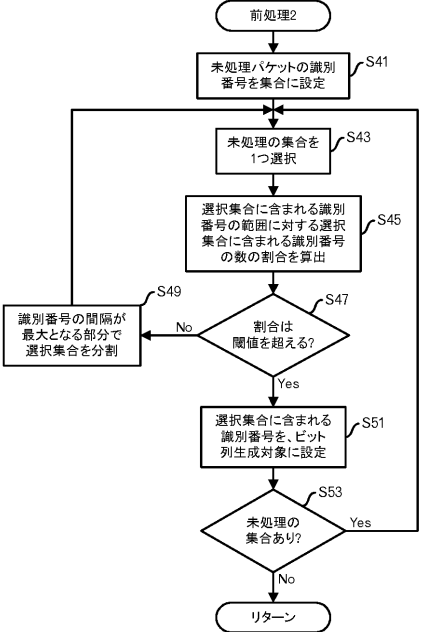
【図 10】



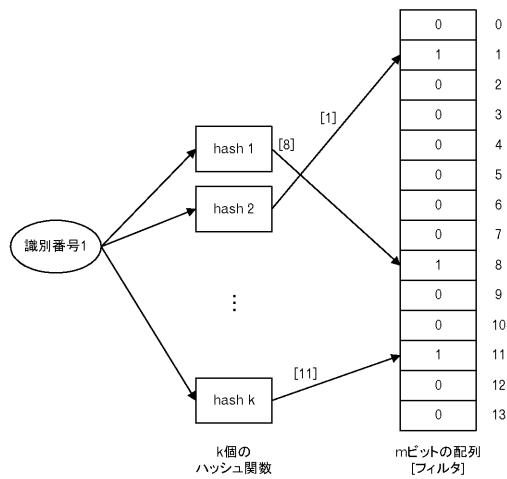
【図 11】



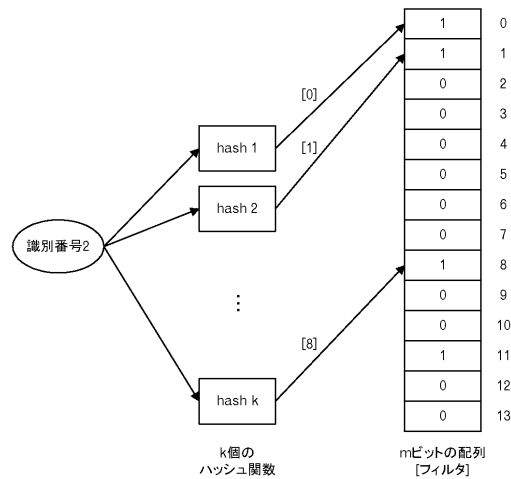
【図 12】



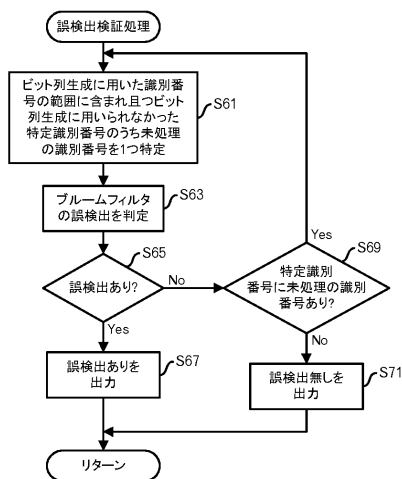
【図 13】



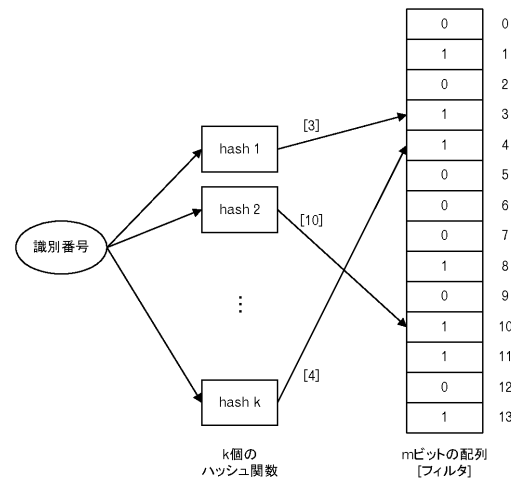
【図 14】



【図 15】



【図 16】



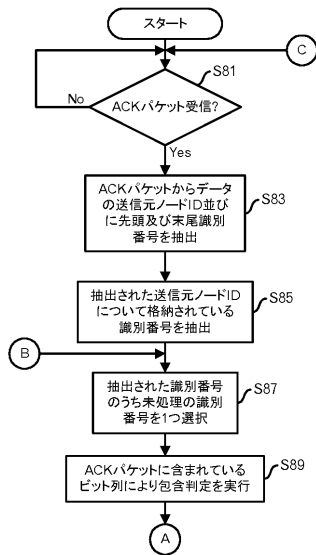
【図 17】

パケットタイプ: 到達確認
 送信元ノードID: S1
 先頭識別番号: 0
 末尾識別番号: 51
 ブルームフィルタ: {0,1,1,1,1,1,0,0,0,0,1,0,1,1,0,0,0,1,1,1,0,1}

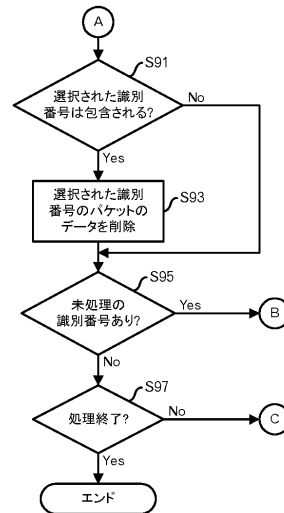
【図 18】

未処理識別番号	$y_1, y_2, y_3, \dots, y_n$
---------	-----------------------------

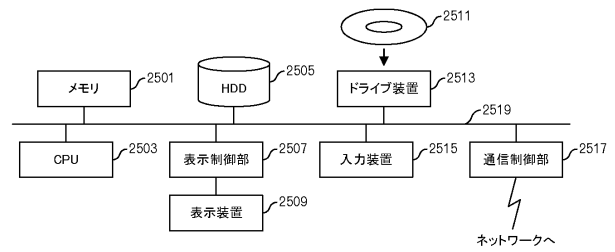
【図 19】



【図 20】



【図 21】



フロントページの続き

(56)参考文献 特表2010-539816(JP,A)
特開平07-202856(JP,A)
特開平11-177536(JP,A)
特開2010-213239(JP,A)
特表2012-501127(JP,A)

(58)調査した分野(Int.Cl., DB名)
H04L 1/16
H04L 29/08