(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2017-108451
(P2017-108451A)
(43)公開日　平成29年6月15日(2017.6.15)

(51)Int.Cl.			F I			テーマコード（参考）
H04L	***9/08***	***(2006.01)***	H04L	9/00	601B	5J104
H04L	***9/14***	***(2006.01)***	H04L	9/00	641	5K030
H04L	***12/70***	***(2013.01)***	H04L	9/00	601E	
			H04L	12/70	F	

審査請求 有　請求項の数 9　OL　（全 22 頁）

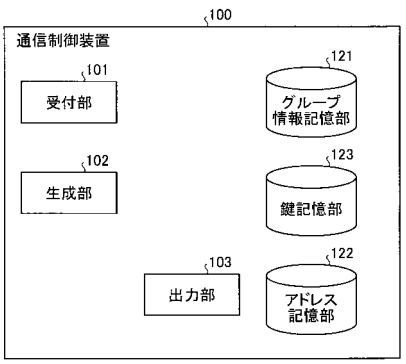
(21)出願番号　特願2017-30227 (P2017-30227)
(22)出願日　平成29年2月21日 (2017.2.21)
(62)分割の表示　特願2015-554420 (P2015-554420)の分割
原出願日　平成25年12月26日 (2013.12.26)

(71)出願人　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(74)代理人　110002147
特許業務法人酒井国際特許事務所
(72)発明者　大場　義洋
東京都港区芝浦一丁目１番１号　株式会社東芝内
(72)発明者　上林　達
東京都港区芝浦一丁目１番１号　株式会社東芝内
(72)発明者　花谷　嘉一
東京都港区芝浦一丁目１番１号　株式会社東芝内

最終頁に続く

(54)【発明の名称】通信制御装置、通信制御方法、プログラムおよび通信システム

(57)【要約】
【課題】スケーラビリティを確保しながら動的なグループ管理を実現する。
【解決手段】通信制御装置は、受付部と、生成部と、出力部と、を備える。受付部は、葉ノードそれぞれにインデックスが割り当てられた二分木と、葉ノードのうちグループに属する葉ノードを識別するノードＩＤと、の入力を受け付ける。生成部は、ノードＩＤで識別される葉ノードのみをそれぞれ含む二分木の部分木を所定数含む集合を示す集合情報と、集合に含まれる部分木の葉ノードに割り当てられたインデックスの範囲情報と、を生成する。出力部は、集合情報と、範囲情報とを、少なくともグループに属する葉ノードに対応づけられる通信装置に対して出力する。
【選択図】図３



【特許請求の範囲】
【請求項１】
葉ノードそれぞれにインデックスが割り当てられた二分木と、葉ノードのうちグループに属する葉ノードを識別するノードＩＤと、を用いて生成された集合情報と範囲情報とを少なくとも前記グループに属する葉ノードに対応づけられる通信装置に対して出力する出力部を備え、
前記集合情報は、前記ノードＩＤで識別される前記葉ノードのみをそれぞれ含む前記二分木の部分木を所定数含む集合を示す情報であって前記部分木のルートノードのノードＩＤを含み、
前記範囲情報は、前記集合に含まれる前記部分木の葉ノードに割り当てられた前記インデックスの下限値と上限値を示す、
通信制御装置。
【請求項２】
前記集合情報および前記範囲情報は、左端の葉ノードおよび右端の葉ノードの一方から他方に向かって、前記ノードＩＤで識別される前記葉ノードのみをそれぞれ含む前記部分木を前記所定数含む集合を示す前記集合情報を求める処理と、求めた前記集合に含まれる葉ノードの前記インデックスの前記範囲情報を求める処理と、を含むトレース処理を繰り返すことにより生成され、
請求項１に記載の通信制御装置。
【請求項３】
前記集合情報および前記範囲情報は、左端の葉ノードおよび右端の葉ノードの一方から他方に向かって、前記ノードＩＤで識別される前記葉ノードのみをそれぞれ含む前記部分木を求める処理を繰り返して１以上の前記部分木を求め、求めた前記部分木を、それぞれ前記所定数含む前記集合に分割し、前記集合それぞれに含まれる前記部分木の葉ノードに割り当てられた前記インデックスの前記範囲情報を求めることにより生成される、
請求項１に記載の通信制御装置。
【請求項４】
前記集合情報および前記範囲情報は、前記グループに葉ノードが追加された場合に、葉ノードを追加後のグループに属する葉ノードの前記ノードＩＤを用いて再度生成される、
請求項１に記載の通信制御装置。
【請求項５】
前記集合情報および前記範囲情報は、前記グループに葉ノードが削除された場合に、葉ノードを削除後のグループに属する葉ノードの前記ノードＩＤを用いて再度生成される、
請求項１に記載の通信制御装置。
【請求項６】
前記集合情報は、前記集合に含まれる前記部分木の葉ノードに対応づけられる通信装置がグループ鍵を導出可能な鍵情報を含む、
請求項１に記載の通信制御装置。
【請求項７】
葉ノードそれぞれにインデックスが割り当てられた二分木と、葉ノードのうちグループに属する葉ノードを識別するノードＩＤと、を用いて生成された集合情報と範囲情報とを少なくとも前記グループに属する葉ノードに対応づけられる通信装置に対して出力する出力ステップを備え、
前記集合情報は、前記ノードＩＤで識別される前記葉ノードのみをそれぞれ含む前記二分木の部分木を所定数含む集合を示す情報であって前記部分木のルートノードのノードＩＤを含み、
前記集合情報は、前記集合に含まれる前記部分木の葉ノードに割り当てられた前記インデックスの下限値と上限値を示す、
通信制御方法。
【請求項８】

コンピュータを、
葉ノードそれぞれにインデックスが割り当てられた二分木と、葉ノードのうちグループに属する葉ノードを識別するノードＩＤと、を用いて生成された集合情報と範囲情報とを少なくとも前記グループに属する葉ノードに対応づけられる通信装置に対して出力する出力部として機能させ、
前記集合情報は、前記ノードＩＤで識別される前記葉ノードのみをそれぞれ含む前記二分木の部分木を所定数含む集合を示す情報であって前記部分木のルートノードのノードＩＤを含み、
前記範囲情報は、前記集合に含まれる前記部分木の葉ノードに割り当てられた前記インデックスの下限値と上限値を示す、
プログラム。
【請求項９】
葉ノードそれぞれにインデックスが割り当てられた二分木と、葉ノードのうちグループに属する葉ノードを識別するノードＩＤと、を用いて生成された集合情報と範囲情報とを少なくとも前記グループに属する葉ノードに対応づけられる通信装置に対して出力する出力部と、
前記範囲情報が示す範囲に、事前に割り当てられたインデックスが含まれるか否かを判定する判定部と、を備え、
前記集合情報は、前記ノードＩＤで識別される前記葉ノードのみをそれぞれ含む前記二分木の部分木を所定数含む集合を示す情報であって前記部分木のルートノードのノードＩＤを含み、
前記集合情報は、前記集合に含まれる前記部分木の葉ノードに割り当てられた前記インデックスの下限値と上限値を示す、
通信システム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明の実施形態は、通信制御装置、通信制御方法およびプログラムに関する。
【背景技術】
【０００２】
ネットワークで接続された多数の機器の管理を効率的に行うために、機器をグループで管理する方法が存在する。グループによる管理法には、予め決められたグループ構造を利用する静的な管理法と、状況に応じてグループを生成および削除する動的な管理法が存在する。
【先行技術文献】
【非特許文献】
【０００３】
【非特許文献１】M. Baugher et al.、 "RFC 3547、 The Group Domain of Interpretation"、 [online]、 July 2003、 retrieved from the Internet: <URL: http://www.ietf.org/rfc/rfc3547.txt>
【発明の概要】
【発明が解決しようとする課題】
【０００４】
しかしながら、動的なグループ管理方法は、状況に応じて柔軟な管理が可能であるが、スケーラビリティの確保が課題となる。
【０００５】
本発明は、上記に鑑みてなされたものであって、スケーラビリティを確保しながら動的なグループ管理を実現できる通信制御装置、通信制御方法およびプログラムを提供することを目的とする。
【課題を解決するための手段】

【０００６】
実施形態の通信制御装置は、受付部と、生成部と、出力部と、を備える。受付部は、葉ノードそれぞれにインデックスが割り当てられた二分木と、葉ノードのうちグループに属する葉ノードを識別するノードＩＤと、の入力を受け付ける。生成部は、ノードＩＤで識別される葉ノードのみをそれぞれ含む二分木の部分木を所定数含む集合と、集合に含まれる部分木の葉ノードに割り当てられたインデックスの範囲情報と、を生成する。出力部は、集合情報と、範囲情報とを、少なくともグループに属する葉ノードに対応づけられる通信装置に対して出力する。
【図面の簡単な説明】
【０００７】
【図１】本実施形態のグループ管理木の構造の一例を示す図。
【図２】実施形態にかかる通信システムのブロック図。
【図３】実施形態にかかる通信制御装置のブロック図。
【図４】実施形態にかかる通信装置のブロック図。
【図５】実施形態における通信制御処理のフローチャート。
【図６】実施形態における生成処理のフローチャート。
【図７】左優先の場合のＣｈｅｃｋ（）関数のフローチャート。
【図８】右優先の場合のＣｈｅｃｋ（）関数のフローチャート。
【図９】左優先の場合の生成処理を表す擬似コードを示す図。
【図１０】生成処理の処理結果の一例を示す図。
【図１１】本実施形態におけるグループ制御処理のフローチャート。
【図１２】変形例１の生成処理のフローチャート。
【図１３】リスト生成処理のフローチャート。
【図１４】左優先の場合の変形例１のＣｈｅｃｋ（）関数のフローチャート。
【図１５】左優先の場合の変形例１の生成処理を表す擬似コードを示す図。
【図１６】本実施形態にかかる通信制御装置のハードウェア構成図。
【発明を実施するための形態】
【０００８】
以下に添付図面を参照して、この発明にかかる通信制御装置の好適な実施形態を詳細に説明する。
【０００９】
ＧＤＯＩ（The Group Domain of Interpretation）は、マルチキャストを用いて、グループメンバの参加と離脱、および、グループ鍵の安全な配布を行うための技術である。ＧＤＯＩでは、グループの作成、グループの更新、および、グループ鍵の配布を行うことができる。しかし、ＧＤＯＩでは、グループメンバの更新ごとに、ほとんどすべてのメンバにおいて、階層構造を持つ鍵情報（LKH__DOWNLOAD__ARRAY）が更新される。そのため、１つの通信装置が複数のグループに所属するような場合には、単一の通信装置が複数のLKH__DOWNLOAD__ARRAYを持つことが必要となり、効率的な管理を行うことが困難である。
【００１０】
そこで、本実施形態では、ＭＫＢ（メディアキーブロック）という技術を用いてグループ操作を行う。ＭＫＢを用いることにより、単一のデバイス鍵（LKH__DOWNLOAD__ARRAYに相当する鍵束）によって、複数グループへの所属を効率的に管理することが可能である。
【００１１】
例えば、グループ管理木（詳細は後述）を用いることにより、グループに属する通信装置の範囲を示す範囲情報、および、この範囲情報で示される範囲の通信装置が処理できるＭＫＢ（ＭＫＢフラグメント）が得られる。なお、ＭＫＢを用いなくてもグループに属するかを管理（判定）することは可能である。例えば、グループ管理木から生成される集合情報（詳細は後述）と範囲情報とが得られれば、グループに属するか否かを判定できる。この場合は、ＭＫＢ（ＭＫＢフラグメント）は生成しなくてもよい。
【００１２】

ＭＫＢとは、対応するデバイス鍵を用いて処理を行うことで、メディアに記録されたコンテンツを復号するためのメディア鍵を導出可能なデータである。ＭＫＢは１以上の要素を含む。典型的なＭＫＢは、１つのメディア鍵を１以上のデバイス鍵でそれぞれ暗号化することで生成される１以上の暗号文（要素）を含む。さらに、ＭＫＢは、各暗号文を処理するためのデバイス鍵を特定する情報を含んでも良い。ＭＫＢが含む暗号文の個数は、対応するデバイス鍵に依存して決まる。そのため、対応するデバイス鍵によっては、ＭＫＢは、非常に多くの暗号文を要素として含む場合がある。

【００１３】

本実施形態では、ＭＫＢを処理することで得られるメディア鍵を、グループに属する１以上の通信装置で共有されるグループ鍵として用いる。すなわち、グループに属する通信装置が保持するデバイス鍵で処理することでグループ鍵を導出できるＭＫＢを配布する。このように、グループに属する通信装置にのみグループ鍵を配布できることを用いて、通信装置のグループ管理を実現している。メディア鍵をグループ鍵として用いるため、ＭＫＢの代わりにＧＫＢ（グループキーブロック）と表現することもできる。

【００１４】

ＭＫＢによってグループ管理（操作）を行う場合、あるＭＫＢを処理してグループ鍵を取り出すことができたメンバはグループに所属する（現在所属していなければ、グループに新規参加する）という制御が行われる。また、グループ鍵の取得に失敗したメンバは当該グループに所属しない（現在所属していれば、グループから離脱する）という制御が行われる。

【００１５】

しかし、対象となるメンバの数が非常に大きくなった場合、グループ操作用のＭＫＢのサイズが極めて大きくなる可能性がある。従って、そのままＭＫＢを通信ネットワークに配布した場合、通信負荷が極めて大きくなる可能性がある。

【００１６】

そこで、本実施形態では、ネットワーク負荷を軽減するために、複数の暗号文を要素として含むＭＫＢを分割して送付する。しかし、上記のようなグループ制御方法を前提とする場合、単純に暗号文ごとにＭＫＢを分割送付しても、意図した通りのグループ制御を行うことができない場合がある。例えば、通信装置が分割されたＭＫＢを受信して、かつ、当該ＭＫＢからグループ鍵を取得できなかった場合、当該通信装置はグループからの離脱処理を行う。しかし、実際には、当該通信装置が処理してグループ鍵を取得することができるようなＭＫＢが後から到着する可能性があるからである。

【００１７】

この問題を回避するため、ＭＫＢによるグループ操作の対象となる通信装置の集合を定める情報を、当該ＭＫＢに付す。例えば、対象となる通信装置を識別する装置ＩＤ（以下、デバイスＩＤという）の範囲を、通信装置の集合を定める情報として利用できる。例えば、数値的に連続したデバイスＩＤが割り振られている場合、第１のデバイスＩＤと第２のデバイスＩＤで、第１のデバイスＩＤと第２のデバイスＩＤで特定される範囲に属するデバイスＩＤの集合を表すことができる。この集合には、第１のデバイスＩＤ以上、かつ、第２のデバイスＩＤ以下であるデバイスＩＤが属する。規則に従ってデバイスＩＤが割り振られていれば、上述のように、２つのデバイスＩＤで範囲を表し、その規則より範囲に含まれるデバイスＩＤを特定することや、デバイスＩＤがその範囲に含まれるか否かを判定することができる。通信装置の範囲を定める情報が付されたＭＫＢを受け取った通信装置は、以下の擬似コードに示すような動作を行う。

```
当該通信装置自身が指定範囲に含まれているか否かチェックする；
ｉｆ（集合に含まれている）｛
  ＭＫＢを処理；
  ｉｆ（グループ鍵取得成功）｛
    ｉｆ（現在グループに所属）｛
      グループ更新；
```

```
      }
      ｅｌｓｅ｛　　ｉｆ（現在グループに所属していない）｛
        グループに参加；
      }
    }
    ｅｌｓｅ｛
      ｉｆ（現在グループに所属）｛
        グループ離脱；
      }
    }
  }
```

【００１８】

当該通信装置自身が指定された集合（範囲）に含まれているか否かチェックする。もし、指定された集合に含まれているならば、当該通信装置が保持しているデバイス鍵によりＭＫＢを処理し、グループ鍵の取得に成功し、かつ、グループに参加しているならば、導出したグループ鍵を用いて、参加しているグループの情報を更新する。グループ鍵の取得に成功し、かつ、グループに参加していないならば、導出したグループ鍵を用いて、グループに参加する。グループ鍵の取得に失敗し、かつ、グループに参加しているならば、グループより離脱する。

【００１９】

このように、本実施形態では、最初に通信装置がグループ操作の対象装置であるか否かをチェックする。そして、自身がグループ操作の対象ではない場合、グループ操作を実行しない。これにより、分割されたＭＫＢによっても、意図しないグループ離脱動作が起きないようにすることができる。

【００２０】

次に、本実施形態で用いるＭＫＢの構造について説明する。図１は、本実施形態で用いるＭＫＢで使用されるグループ管理木の構造の一例を示す図である。図１に示すように、本実施形態では、ＣＳ（Complete Subtree）法に従う完全二分木構造のグループ管理木が用いられる。この完全二分木は、例えば対象システムのすべての通信装置をカバーする木（完全二分木Ｔとする）でもよいし、対象システムのすべての通信装置のうち、一部の通信装置（例えばグループ操作の対象となる通信装置）の集合のみをカバーする木（完全二分木Ｔの部分木Ｔ’とする）であってもよい。以下、図１では完全二分木Ｔが用いられるものとして説明する。

【００２１】

上述のように、各通信装置は、対象システムで一意なデバイスＩＤを持つ。完全二分木Ｔの葉ノードは、それぞれ１つの通信装置に対応する。従って、通信装置をグループで管理することは、葉ノードをグループで管理することに置き換えられる。

【００２２】

葉ノードは、対応する通信装置のデバイスＩＤと等しいインデックスを持つ。破線の丸で示す葉ノードは、リボーク（グループから離脱）されたノード（リボークノード）を表す。太線は、ルート（根）からリボークノードまでのパス上のエッジを表す。三角形は、リボークされていない葉ノードのみを含む部分木（部分木ｓ）を表す。塗りつぶされたノードは、この部分木ｓの根ノードを表す。

【００２３】

完全二分木Ｔの各ノードには、異なる暗号鍵（ノード鍵）が割り当てられるかもしれない。各通信装置は、完全二分木Ｔの根ノードから対応する葉ノードに至るパス上の各ノードに割り当てられたノード鍵を含むデバイス鍵が予め設定されるかもしれない。

【００２４】

なお、ノード鍵の割り当て、および、デバイス鍵の設定は行わなくてもよい。例えば、上述のようにＭＫＢを用いずにグループを管理する場合は、ＭＫＢを生成する必要はない

ため、ノード鍵の割り当ておよびデバイス鍵の設定は不要となる。

【００２５】

また、各ノードには、少なくともノードを識別する情報（ノードＩＤ）が属性として割り当てられる。ノードＩＤは、例えば、（ｄ（ノードの深さ），ｂ（ビットマップ））で表される。ノードｎの深さｄは（$H_T - H_n$）で表される。ノードｎは、完全二分木Ｔに含まれる各ノードである。H_Tは完全部分木Ｔの高さ、H_nはノードｎの高さである。ノードＩＤ（ｄ，ｂ）で識別されるノードのインデックスｉｎｄｅｘ（ｂ，ｄ）は、ビットマップｂの先頭ｄビットの値で表される。

【００２６】

ビットマップｂは、例えば図１に示すように「０」または「１」を１以上含む値である。図１で示すインデックスは、完全二分木Ｔの根ノードから葉ノードまでの経路について、左に進んだ場合に０を割り当て、右に進んだ場合に１を割り当てることで得られる。

【００２７】

グループ管理木から生成されるＭＫＢは、例えば以下の要素を含む。ｉは部分木ｓのルートノードである。

（ノードｉのインデックス，Ｅｎｃ（ｉのノード鍵，グループ鍵））

【００２８】

図１のグループ管理木の例では、次の４つの要素を含むＭＫＢが生成される。これらは、ノード０００、０１０、１０、１１０にそれぞれ対応する。ただし、Ｋｇはグループ鍵を表し、例えばＥｎｃ（ｋ（０００），Ｋｇ）はＫｇをｋ（０００）で暗号化したデータを表す。

（０００，Ｅｎｃ（ｋ（０００）、Ｋｇ），（０１０，Ｅｎｃ（ｋ（０１０）、Ｋｇ），（１０，Ｅｎｃ（ｋ（１０）、Ｋｇ），（１１０，Ｅｎｃ（ｋ（１１０）、Ｋｇ）

【００２９】

本実施形態では、このような構造のＭＫＢが、それぞれ一部の要素を含む複数のＭＫＢ（ＭＫＢフラグメント）に分割され、分割されたＭＫＢフラグメントが通信装置に対してマルチキャスト通信またはブロードキャスト通信などにより送信される。

【００３０】

以下、本実施形態の詳細について説明する。図２は、本実施形態にかかる通信システムの構成の一例を示すブロック図である。図２に示すように、本実施形態の通信システムは、通信装置２００ａ～２００ｆと、通信制御装置１００とが、ネットワーク６０で接続された構成となっている。ネットワーク６０は、インターネットなどのあらゆるネットワーク形態を適用できる。各通信装置２００ａ～２００ｆは、通信制御装置１００と直接接続されている必要はない。

【００３１】

通信制御装置１００は、１つに限られるものではなく、２以上の通信制御装置を備えるように構成してもよい。通信装置２００ａ～２００ｆは同様の構成を備えるため、以下では単に通信装置２００という場合がある。通信装置２００の個数は６に限られるものではない。

【００３２】

図２に示すように、本実施形態では、通信制御装置１００が、各通信装置２００に対してグループ操作コマンドを送信する。グループ操作コマンドは、例えば、集合情報、および、範囲情報（例えばデバイスＩＤの範囲）を含む。グループ操作コマンドが、更新後のグループを識別するグループＩＤを含んでもよい。

【００３３】

集合情報は、グループに属する（通信装置に対応する）葉ノードのみを含む部分木（例えば図１の部分木ｓ）を所定数含む集合を示す情報である。集合情報は、例えば、集合に含まれる部分木の葉ノードに対応づけられる通信装置がグループ鍵を導出できるように分割されたＭＫＢ（ＭＫＢフラグメント）を含んでもよい。このように、本実施形態では、ＭＫＢ全体ではなく、所定数ごとに部分木ｓを含むように生成される集合情報に対応して

分割されたＭＫＢフラグメントを通信装置２００に送信する。

【００３４】

範囲情報は、集合情報が示す集合に含まれる各部分木の葉ノードに割り当てられたインデックスの範囲を示す。上述のようにインデックスは対応する通信装置のデバイスＩＤと等しいため、範囲情報によってグループ操作の対象となる通信装置のデバイスＩＤの範囲が特定される。集合情報および範囲情報の生成方法の詳細は後述する。

【００３５】

図３は、通信制御装置１００の構成例を表すブロック図である。図３に示すように、通信制御装置１００は、グループ情報記憶部１２１と、アドレス記憶部１２２と、鍵記憶部１２３と、受付部１０１と、生成部１０２と、出力部１０３と、を備える。

【００３６】

グループ情報記憶部１２１は、１以上の通信装置２００が所属するグループのグループＩＤと、グループＩＤで識別されるグループに属する通信装置２００を識別するデバイスＩＤとを含むグループ情報を記憶する。つまり、グループ情報記憶部１２１は、グループＩＤと、グループＩＤで識別されるグループに属する通信装置２００のデバイスＩＤとを対応付けて記憶する。

【００３７】

本実施形態では、グループ情報記憶部１２１は、予め、１以上のグループＩＤを記憶しているとするが、グループ情報記憶部１２１を備えず、外部装置より受け取ったグループ情報に基づき、グループ操作を行うようにしてもよい。

【００３８】

アドレス記憶部１２２は、１以上の通信装置２００が所属するマルチキャストグループを特定する情報（マルチキャストグループＩＤ、マルチキャストアドレスなど）と、マルチキャストグループに属する通信装置２００のデバイスＩＤとを対応づけて記憶する。マルチキャストグループは、ＭＫＢによるグループ操作の対象となるグループと独立に管理されるグループの一例である。マルチキャストアドレスは、例えば、対応する各デバイスＩＤの通信装置２００に対してマルチキャスト通信で情報を送信するためのアドレスである。マルチキャスト通信を用いない場合（例えばブロードキャスト通信を用いる場合）、アドレス記憶部１２２を備えないように構成してもよい。

【００３９】

本実施形態では、アドレス記憶部１２２は、予めマルチキャストグループを特定する情報を記憶しているとするが、外部装置より受け取った情報に基づき、アドレス記憶部１２２に新たな情報を追加したり、既に記憶されている情報の更新を行ったりするように構成しても良い。

【００４０】

鍵記憶部１２３は、通信装置２００それぞれに割り当てられるデバイス鍵を記憶する。ＭＫＢがＣＳ法などにより生成される場合、鍵記憶部１２３が、木構造のノードに対応づけてデバイス鍵を記憶してもよい。上述のようにＭＫＢを用いずにグループの管理のみを行う場合は、鍵記憶部１２３は備えなくてもよい。

【００４１】

受付部１０１は、通信制御装置１００で用いる各種情報の入力を受け付ける。例えば、受付部１０１は、通信装置２００などの外部装置から各種情報を受信する。受付部１０１は、例えば、グループ制御の要求、および、グループ制御の対象を指定する情報などを受信する。グループ制御の要求とは、グループの新規作成、グループの変更（グループに属する通信装置２００の変更など）などの要求である。例えば、キーボードなどの操作部（図示せず）を用いてオペレータが入力した、操作対象となるグループのグループＩＤと、当該グループに入れるべき通信装置２００のデバイスＩＤとを、受付部１０１が受信するように構成してもよい。なお、外部装置からグループ制御の要求を受信した場合のみでなく、通信制御装置１００内でグループ制御の要否を判断し、必要と判断した場合にグループ制御を実行してもよい。

【００４２】
　また、受付部１０１は、処理対象とする完全二分木構造のグループ管理木（全体木である完全二分木Ｔでもよいし、完全二分木Ｔの部分木Ｔ’であってもよい）、および、グループに属する通信装置２００に対応する葉ノードのノードＩＤの入力を受け付ける。葉ノードは通信装置２００のいずれかと対応するため、グループに属する通信装置２００のデバイスＩＤの入力を受け付けてもよい。受付部１０１は、入力されたデバイスＩＤの通信装置２００に対応する葉ノードのノードＩＤを得ることができる。受付部１０１は、完全二分木構造のグループ管理木、ノードＩＤ、グループ制御の要求、および、グループ制御の対象を指定する情報（入力情報）を生成部１０２に送る。
【００４３】
　生成部１０２は、グループ操作に用いる情報を生成する。例えば生成部１０２は、上述の集合情報と範囲情報とを生成する。生成部１０２は、完全二分木構造のグループ管理木をトレースし、集合情報を算出しながら範囲情報を生成する。例えば生成部１０２は、左端の葉ノードおよび右端の葉ノードの一方から他方に向かって、ノードＩＤで識別される葉ノードのみをそれぞれ含む部分木（グループ管理木の部分木）を所定数（Ｍ（自然数））含む集合を示す集合情報を求める処理と、求めた集合に含まれる葉ノードのインデックスの範囲情報を求める処理と、を含むトレース処理を繰り返す。これにより、計算量をＯ（Ｌ）（Ｌは葉ノードの個数）とすることができる。生成部１０２による生成処理の詳細は後述する。
【００４４】
　出力部１０３は、集合情報と、範囲情報とを、少なくともグループに属する葉ノードに対応づけられる通信装置２００に対して出力する。上述のように、集合情報は、集合に含まれる部分木の葉ノードに対応づけられる通信装置２００がグループ鍵を導出可能な鍵情報（ＭＫＢフラグメント）を含んでもよい。ＭＫＢフラグメントは、例えば上述のように（ノードｉのインデックス，Ｅｎｃ（ｉのノード鍵，グループ鍵））の形式で表される。
【００４５】
　例えば、出力部１０３は、集合情報と範囲情報とを含むグループ操作メッセージを、グループに属する通信装置２００が属するマルチキャストグループ宛にマルチキャストで送信する。このようにグループ変更の対象ではない通信装置２００にも出力部１０３の出力が届くことを許容することで、許容しない場合と比較して、出力部１０３による出力先の決定に要する計算コストを軽減できる。
【００４６】
　また、出力部１０３より、更新前のグループには含まれるが、更新後のグループには含まれない通信装置２００が含まれるマルチキャストグループに、上述の情報を送信するようにしてもよい。そのような通信装置２００は、マルチキャストグループに属しているが、ＭＫＢを正しく処理できないため、更新後のグループから離脱を行う。このように、分割したＭＫＢを用いて、グループの離脱を命じるコマンドを発行することができる。このようなコマンドを発行することで、通信装置２００が保持すべき情報を適切に管理することができる。
【００４７】
　また、更新後のグループに含まれない通信装置２００に、上述のような、離脱を命じるコマンドを送付しなくても良い。これは、更新後のグループに含まれない通信装置２００は、更新を命じるコマンドより、更新後のグループ鍵を導出できないため、更新後のグループには参加できないためである。このように構成することで、通信制御装置１００が発行すべきコマンドの量を削減できる場合がある。
【００４８】
　出力部１０３は、ＭＫＢによるグループ操作の対象となるグループとは独立に管理される通信装置２００の集合（グループ）であって、少なくともグループが更新されたすべての通信装置２００を含む通信装置２００の集合に対して出力情報を出力する。ここで、通信装置２００の集合とは、複数の通信装置２００からなる集まりであって、グループＩＤ

が割り振られているグループとは必ずしも一致しない。通信装置２００の集合の例として、あるマルチキャスト通信でデータを受信する通信装置２００の集合や、ブロードキャスト通信でデータを受信する通信装置２００からなる集合、つまりすべての通信装置２００からなる集合、などが挙げられる。例えば、出力部１０３は、デバイスＩＤリストが含まれる通信装置２００の集合、または、グループに対して、１以上のマルチキャスト通信やブロードキャスト通信により出力情報を送信してもよい。マルチキャスト通信で出力情報を送信する場合、出力部１０３は、例えばアドレス記憶部１２２に記憶されたアドレスのうち、配布対象のデバイスＩＤのデバイスＩＤに対応づけられた１以上のアドレス（マルチキャストアドレス）を宛先として、出力情報を送信する。

【００４９】

図４は、通信装置２００の構成例を表すブロック図である。図４に示すように、通信装置２００は、ＧＩＤ記憶部２２１と、グループ鍵記憶部２２２と、デバイス鍵記憶部２２３と、デバイスＩＤ記憶部２２４と、受信部２０１と、判定部２０２と、ＭＫＢ処理部２０３と、グループ制御部２０４と、を備える。ＭＫＢを用いない場合は、グループ鍵記憶部２２２およびデバイス鍵記憶部２２３は備えなくてもよい。

【００５０】

ＧＩＤ記憶部２２１は、通信装置２００が属するグループのグループＩＤ（ＧＩＤ）を記憶する。グループ鍵記憶部２２２は、ＧＩＤ記憶部２２１が記憶するグループＩＤで識別されるグループのグループ鍵を記憶する。デバイス鍵記憶部２２３は、通信装置２００のデバイス鍵を記憶する。デバイスＩＤ記憶部２２４は、通信装置２００のデバイスＩＤを記憶する。

【００５１】

受信部２０１は、通信制御装置１００および他の通信装置２００などの外部装置から各種情報を受信する。例えば、受信部２０１は、グループ操作メッセージを通信制御装置１００から受信する。受信部２０１は、マルチキャスト通信およびブロードキャスト通信などにより出力情報を受信する。受信部２０１は、受信されたメッセージがグループ操作メッセージであるか判定する。グループ操作メッセージでない場合、受信されたメッセージは、当該メッセージを処理すべき他のモジュール（図示せず）に渡されて処理される。メッセージがグループ操作メッセージである場合、メッセージのデータは判定部２０２に送られる。

【００５２】

判定部２０２は、グループ操作メッセージに含まれる範囲情報に、デバイスＩＤ記憶部２２４に記憶されたデバイスＩＤが含まれるか否か判定する。含まれない場合、当該通信装置２００は受信したグループ操作メッセージの対象機器ではないので、当該グループ操作メッセージに対する動作を中止する。含まれる場合は、当該通信装置２００はグループ操作メッセージの対象であるので、グループ操作メッセージはＭＫＢ処理部２０３に渡される。

【００５３】

ＭＫＢ処理部２０３は、範囲情報にデバイスＩＤ記憶部２２４に記憶されたデバイスＩＤが含まれると判定された場合に、グループ操作メッセージに含まれる集合情報（ＭＫＢフラグメント）と、デバイス鍵記憶部２２３に記憶されたデバイス鍵とからグループ鍵を生成するＭＫＢ処理を実行する。

【００５４】

例えば上述のようにＭＫＢフラグメントが（ノードｉのインデックス，Ｅｎｃ（ｉのノード鍵，グループ鍵））で表される場合、ＭＫＢ処理部２０３は、Ｄｅｃ（ノードｉのノード鍵、ＭＫＢフラグメント）によりグループ鍵を生成する。

【００５５】

ＭＫＢ処理の結果、グループ鍵が得られた場合、当該通信装置２００はＧＩＤに示されるグループに所属することを意味する。ＭＫＢ処理部２０３は、ＧＩＤと、取得したグループ鍵と、をグループ制御部２０４に送る。

【００５６】
なお、グループに所属するかの判定方法はこれに限られるものではない。例えばＭＫＢフラグメントを含まない集合情報を用いる場合であれば、通信装置２００のデバイスＩＤのビットマップとプレフィックス上位ｄビットが一致するノードインデックス（ｄ、ｂ）が集合情報に含まれていればグループに所属すると判定してもよい。
【００５７】
グループ制御部２０４は、ＧＩＤをＧＩＤ記憶部２２１に格納し、グループ鍵をグループ鍵記憶部２２２に格納する。既にＧＩＤが記憶されている場合は、グループ制御部２０４は、グループ操作メッセージ内のＧＩＤで、ＧＩＤ記憶部２２１に記憶されているＧＩＤを更新する。
【００５８】
一方、ＭＫＢ処理の結果、グループ鍵が得られなかった場合は、当該通信装置２００はＧＩＤに示されるグループに所属しないことを意味する。従って、所属している場合は離脱する必要がある。このため、ＭＫＢ処理部２０３は、ＧＩＤと、グループ鍵が取得できなかった旨の通知と、をグループ制御部２０４に送る。
【００５９】
グループ制御部２０４は、ＧＩＤ記憶部２２１とグループ鍵記憶部２２２とを空にする。ＧＩＤまたはグループ鍵が既に格納されている場合は、グループ制御部２０４はそれらを消去する。
【００６０】
なお、上述の各記憶部は、ＨＤＤ（Hard Disk Drive）、光ディスク、メモリカード、ＲＡＭ（Random Access Memory）などの一般的に利用されているあらゆる記憶媒体により構成することができる。
【００６１】
また、通信制御装置１００の受付部１０１、生成部１０２、および、出力部１０３、並びに、通信装置２００の受信部２０１、判定部２０２、ＭＫＢ処理部２０３、および、グループ制御部２０４は、例えば、ＣＰＵ（Central Processing Unit）などの処理装置にプログラムを実行させること、すなわち、ソフトウェアにより実現してもよいし、ＩＣ（Integrated Circuit）などのハードウェアにより実現してもよいし、ソフトウェアおよびハードウェアを併用して実現してもよい。
【００６２】
次に、本実施形態にかかる通信制御装置１００による通信制御処理について図５を用いて説明する。図５は、本実施形態における通信制御処理の一例を示すフローチャートである。
【００６３】
受付部１０１は、完全二分木Ｔ（または完全二分木Ｔの部分木Ｔ’）と、グループに属する通信装置２００に対応する葉ノードのノードＩＤとを受け付ける（ステップＳ１０１）。グループに属する通信装置２００のデバイスＩＤを受け付けて、受け付けたデバイスＩＤの通信装置２００に対応する葉ノードのノードＩＤを求めてもよい。生成部１０２は、受け付けた完全二分木Ｔ（または部分木Ｔ’）とノードＩＤとをもとに、集合情報と範囲情報とを生成する生成処理を実行する（ステップＳ１０２）。生成部１０２は、集合情報と範囲情報とを含むグループ操作メッセージを生成する。出力部１０３は、グループ操作メッセージを出力する（ステップＳ１０３）。
【００６４】
次に、ステップＳ１０２の生成処理の詳細について説明する。生成処理では、完全二分木Ｔの部分木Ｔ’の根ノードＲから左優先または右優先のいずれか一方で再帰的にトレース処理を実行する。左優先は、部分木Ｔ’の左端の葉ノードから右端の葉ノードに向けて順に集合情報および範囲情報を生成することを意味する。右優先は、部分木Ｔ’の右端の葉ノードから左端の葉ノードに向けて順に集合情報および範囲情報を生成することを意味する。生成部１０３は、左優先および右優先のいずれによって集合情報等を生成してもよ

い。
【００６５】
生成処理は、部分木Ｔ’の高々Ｍ個のノードのノードＩＤのリストＳと、リストＳに対応する葉ノードのインデックスの下限値ｍｉｎｒおよびインデックスの上限値ｍａｘｒの組（Ｓ，ｍｉｎｒ，ｍａｘｒ）のリストＯと、を出力する。リストＳが集合情報に相当し、ｍｉｎｒおよびｍａｘｒが範囲情報に相当する。なお、以下ではリストＳにノードｎのノードＩＤを追加することを、単にリストＳにノードｎを追加すると表現する場合がある。
【００６６】
以下では、ノードｎを根ノードとする部分木の左端および右端の葉ノードのノードインデックスを返す関数をそれぞれＬＭＬ（ｎ）およびＲＭＬ（ｎ）とする。部分木Ｔ’の葉ノードの集合を集合Ｌとする。集合Ｌのうちグループに属する葉ノードの集合を集合Ｇとする。
【００６７】
生成処理は、例えば以下のステップを含む：
（Ｓ１）Ｏ＝Ｓ＝ＮＵＬＬ、ｍｉｎｒ＝ＬＭＬ（Ｒ）、ｍａｘｒ＝ＲＭＬ（Ｒ）に初期化する。
（Ｓ２）現在トレース中のノードＣに対し、ノードＣが集合Ｌに含まれる場合、ノードＣが集合Ｇに含まれる場合にはノードＣをリストＳに追加するとともにノードＣを「ＣＳ該当」とマークする。また、ノードＣが集合Ｇに含まれない場合にはノードＣを「ＣＳ非該当」とマークする。
（Ｓ３）ノードＣが集合Ｌに含まれず、かつ、ノードＣの左右両方の子ノードが「ＣＳ非該当」とマークされた場合には、ノードＣを「非ＣＳ該当」とマークする。
（Ｓ４）ノードＣが集合Ｌに含まれず、かつ、ノードＣの左右両方の子ノードが「ＣＳ該当」とマークされた場合には、ノードＣを「ＣＳ該当」とマークするとともに、左右両方の子ノードをリストＳから削除し、ノードＣをリストＳに追加する。
（Ｓ５）ノードＣが集合Ｌに含まれず、かつ、ノードＣの一方の子ノードが「ＣＳ該当」で他方の子ノードが「ＣＳ非該当」とマークされた場合には、|Ｓ|＞Ｍ（リストＳの要素数がＭより大）の場合には、リストＳの先頭からＭ個のノードのリストをＳ［０：Ｍ］、左優先の場合ｍａｘｒ＝ＲＭＬ（Ｓ［Ｍ－１］）、右優先の場合ｍｉｎｒ＝ＬＭＬ（Ｓ［Ｍ－１］）とする。そして、（Ｓ［０：Ｍ］，ｍｉｎｒ，ｍａｘｒ）を集合Ｏに追加し、Ｓ［０：Ｍ］をリストＳから削除し、左優先の場合ｍｉｎｒ＝ｍａｘｒ＋１、右優先の場合ｍａｘｒ＝ｍｉｎｒ－１に設定する。
（Ｓ６）ノードＣが部分木Ｔ’の根ノードＲの場合、左優先の場合ｍａｘｒ＝ＲＭＬ（Ｒ）、右優先の場合ｍｉｎｒ＝ＬＭＬ（Ｒ）として、（Ｓ，ｍｉｎｒ，ｍａｘｒ）を集合Ｏに追加する。
【００６８】
「ＣＳ該当」および「ＣＳ非該当」のマークは、現在トレース中のノードに対する処理を行う関数の返り値として実現してもよいし、各ノードに付随する属性の値として保持することにより実現してもよい。属性の値として保持する場合、「ＣＳ該当」および「ＣＳ非該当」に対応する値に加え、ＣＳ該当かどうかが未確定であることを示す「ＣＳ未定」に対応する値のいずれかを保持してもよい。
【００６９】
葉ノードのインデックスが上述のようにｉｎｄｅｘ（ｂ，ｄ）で表される場合、ノードＩＤ（ｄ，ｂ）で識別されるノードｎに対し、ＬＭＬ（ｎ）＝ｉｎｄｅｘ（ｂ，ｄ）×２＾（H_T－ｄ）、ＲＭＬ（ｎ）＝（ｉｎｄｅｘ（ｂ，ｄ）＋１）×２＾（H_T－ｄ）－１で表される。例えば、Ｔ＝Ｔ’のとき、Ｔ’の左端葉ノードのノードインデックスはＬＭＬ（Ｒ）＝０×２＾（H_T－０）＝０であり、Ｔ’の右端葉ノードのノードインデックスはＲＭＬ（Ｒ）＝（０＋１）×２＾H_T－１＝２＾H_T－１である。
【００７０】

生成処理は、以下の情報を入力として使用する。
Ｉ:グループメンバに含まれる通信装置２００のデバイスＩＤのリスト
Ｔ’:全体木である完全二分木Ｔの部分木（Ｔ’の根ノードＲ）
Ｍ:ＭＫＢフラグメントサイズ（ＭＫＢフラグメントに含まれるノード数）
【００７１】
また、生成処理は、（Ｓ，ｍｉｎｒ、ｍａｘｒ）のリストＯを出力する。ここで、
Ｓ:ＭＫＢフラグメントに含まれるノードのリスト
ｍｉｎｒ:リストＳに対する葉ノードのノードインデックスの下限値
ｍａｘｒ:リストＳに対する葉ノードのノードインデックスの上限値
である。
【００７２】
図６は、生成処理の一例を示すフローチャートである。生成部１０２は、生成処理で使用する各パラメータを初期化する（ステップＳ２０１）。例えば、生成部１０２は、リストＳおよびリストＯを空（ＮＵＬＬ）にし、ｍｉｎｒ＝ＬＭＬ（Ｒ），ｍａｘｒ＝ＲＭＬ（Ｒ）に設定する。
【００７３】
生成部１０２は、根ノードＲに対しＣｈｅｃｋ（）関数を実行する（ステップＳ２０２）。Ｃｈｅｃｋ関数は、指定されたノードの子ノードに対して再帰的に実行され、結果としてリストＯが出力される。
【００７４】
図７は、左優先の場合のＣｈｅｃｋ（）関数の一例を示すフローチャートである。図７は、あるノードｎに対する左優先のＣｈｅｃｋ（）関数が呼ばれた場合の動作を示す。
【００７５】
まずノードｎが葉ノードであるか否かが判定される（ステップＳ３０１）。葉ノードであれば（ステップＳ３０１：Ｙｅｓ）、ノードｎのノードＩＤがリストＩに含まれるか否かが判定される（ステップＳ３０２）。含まれる場合（ステップＳ３０２：Ｙｅｓ）、リストＳにノードｎが追加される（ステップＳ３０３）。またｒｖに１が設定される（ステップＳ３０４）。ｒｖは、「ＣＳ該当」であるか「ＣＳ非該当」であるかを設定するパラメータである。図７の例では、ｒｖ＝１が「ＣＳ該当」、ｒｖ＝０が「ＣＳ非該当」を意味する。ノードｎのノードＩＤがリストＩに含まれない場合（ステップＳ３０２：Ｎｏ）、ｒｖに０が設定される（ステップＳ３０５）。
【００７６】
ステップＳ３０１で、ノードｎが葉ノードでないと判定された場合（ステップＳ３０１：Ｎｏ）、ｌｖａｌ＝Ｃｈｅｃｋ（ノードｎの左の子ノード）、ｒｖａｌ＝Ｃｈｅｃｋ（ノードｎの右の子ノード）、ｒｖ＝０が設定される（ステップＳ３０６）。
【００７７】
次に、ｌｖａｌ×ｒｖａｌが０より大きいか否かが判定される（ステップＳ３０７）。これは、左右の子ノードの両方が「ＣＳ該当」であるかを判定することに相当する。大きい場合（ステップＳ３０７：Ｙｅｓ）、リストＳからノードｎの左の子ノードおよび右の子ノードが削除され、リストＳにノードｎが追加され、ｒｖに１が設定される（ステップＳ３０８）。
【００７８】
ｌｖａｌ×ｒｖａｌが０より大きくない場合、すなわち、０に一致する場合（ステップＳ３０７：Ｎｏ）、（ｌｖａｌ＋ｒｖａｌ）が０より大きいか否かが判定される（ステップＳ３０９）。これは、左右の子ノードのいずれか一方が「ＣＳ該当」であることを判定することに相当する。
【００７９】
（ｌｖａｌ＋ｒｖａｌ）が０より大きい場合（ステップＳ３０９：Ｙｅｓ）、リストＳの要素数がＭを超えたか否かが判定される（ステップＳ３１０）。リストＳの要素数がＭを超えた場合（ステップＳ３１０：Ｙｅｓ）、ｍａｘｒ＝ＲＭＬ（Ｓ［Ｍ－１］）とされ

、リストＯに（Ｓ［０：Ｍ］，ｍｉｎｒ，ｍａｘｒ）が追加され、ｍｉｎｒ＝ｍａｘｒ＋１とされる（ステップＳ３１１）。

【００８０】

ステップＳ３０８の後、ステップＳ３１１の後、（ｌｖａｌ＋ｒｖａｌ）が０より大きくないと判定された場合（ステップＳ３０９：Ｎｏ）、または、リストＳの要素数がＭを超えていない場合（ステップＳ３１０：Ｎｏ）、ノードｎが根ノードであるか否かが判定される（ステップＳ３１２）。

【００８１】

ノードｎが根ノードであれば（ステップＳ３１２：Ｙｅｓ）、ｍａｘｒ＝ＲＭＬ（Ｒ）とされ、リストＯに（Ｓ，ｍｉｎｒ，ｍａｘｒ）が追加される（ステップＳ３１３）。ステップＳ３０４の後、ステップＳ３０５の後、ステップＳ３１３の後、または、ノードｎが根ノードでない場合（ステップＳ３１２：Ｎｏ）、ｒｖの値を返して（ステップＳ３１４）、Ｃｈｅｃｋ関数が終了する。

【００８２】

図８は、右優先の場合のＣｈｅｃｋ（）関数の一例を示すフローチャートである。右優先の場合は、ステップＳ４１１およびステップＳ４１３以外は左優先の場合のＣｈｅｃｋ（）関数（図７）と同様であるため説明を省略する。

【００８３】

ステップＳ４１１では、ｍｉｎｒ＝ＬＭＬ（Ｓ［Ｍ－１］）とされ、リストＯに（Ｓ［０：Ｍ］，ｍｉｎｒ，ｍａｘｒ）が追加され、ｍａｘｒ＝ｍｉｎｒ－１とされる（ステップＳ４１１）。ステップＳ４１３では、ｍｉｎｒ＝ＬＭＬ（Ｒ）とされ、リストＯに（Ｓ，ｍｉｎｒ，ｍａｘｒ）が追加される（ステップＳ４１３）。

【００８４】

図９は、左優先の場合の生成処理を表す擬似コードの例を示す図である。図９のＩｎｐｕｔ　Ｉ，Ｔ，ＲおよびＭは、上記のリストＩ、部分木Ｔ’、部分木Ｔ’の根ノードＲ、ＭＫＢフラグメントサイズＭに対応する。また、Ｏｕｔｐｕｔ　Ｏは、上記のリストＯに対応する。また、ｒｉｇｈｔｍｏｓｔ＿ｌｅａｆ＿ｎｕｍｂｅｒ（ｎ）は、上記のＲＭＬ（ｎ）に対応する。

【００８５】

図１０は、生成処理の処理結果の一例を示す図である。図１０は、$H_T=3$（葉ノード数８）の全体木である図１の完全部分木Ｔに対し、Ｔ’＝Ｔかつ左優先の場合の生成処理の処理結果の例である。

【００８６】

図１０の例では、以下のようなリストＩが入力パラメータとして与えられる。Ｉの各要素はデバイスＩＤを２進表記したものである。

Ｉ＝（０００，０１０，１００，１０１，１１０）

【００８７】

Ｍ＝２のとき、生成処理の出力であるリストＯは以下の２つのリストＳを含む。

Ｓ＝［（３，０００），（３，０１０）］，（ｍｉｎｒ，ｍａｘｒ）＝（０，２）

Ｓ＝［（２，１０），（３，１１０）］，（ｍｉｎｒ，ｍａｘｒ）＝（３，７）

【００８８】

Ｍ＝３のとき、生成処理の出力であるリストＯは以下の２つのリストＳを含む。

Ｓ＝［（３，０００），（３，０１０），（２，１０）］，（ｍｉｎｒ，ｍａｘｒ）＝（０，５）

Ｓ＝［（３，１１０）］，（ｍｉｎｒ，ｍａｘｒ）＝（６，７）

【００８９】

集合情報にＭＫＢフラグメントを含める場合は、例えば、リストＳに含まれるノードごとにＥｎｃ（ｉのノード鍵，グループ鍵）を算出し、当該ノードのインデックスと対応づけて出力すればよい。

【００９０】

以上のように、本実施形態によれば、グループに属する葉ノードのみを含む部分木を求めながら、ＭＫＢの分割（Ｍ個ずつの部分木のリストＳの生成）を行うことが可能となる。このため、例えばすべての部分木が求められるまで待つことなく、生成されたＭＫＢフラグメントを送信することができる。その結果、例えばすべての部分木を求めてからＭＫＢをＭＫＢフラグメントに分割する方法（後述の変形例１など）と比較して、計算量、および、ＭＫＢフラグメントの送信遅延を低減することができる。

【００９１】

次に、本実施形態にかかる通信装置２００によるグループ制御処理について図１１を用いて説明する。図１１は、本実施形態におけるグループ制御処理の一例を示すフローチャートである。

【００９２】

受信部２０１は、通信制御装置１００などの外部装置からメッセージを受信する（ステップＳ５０１）。受信部２０１は、受信したメッセージがグループ操作メッセージであるか否かを判定する（ステップＳ５０２）。グループ操作メッセージでない場合（ステップＳ５０２：Ｎｏ）、グループ制御処理を終了する。なお、上述のようにグループ操作メッセージ以外のメッセージは、当該メッセージを処理すべきモジュールに渡されて適切に処理される。

【００９３】

グループ操作メッセージである場合（ステップＳ５０２：Ｙｅｓ）、判定部２０２は、グループ操作メッセージ内の範囲情報が示す範囲に、デバイスＩＤ記憶部２２４に記憶されたデバイスＩＤが含まれるか否かを判定する（ステップＳ５０３）。

【００９４】

範囲情報が示す範囲にデバイスＩＤが含まれない場合（ステップＳ５０３：Ｎｏ）、グループ操作の対象外であるため、グループ制御処理を終了する。範囲情報が示す範囲にデバイスＩＤが含まれる場合（ステップＳ５０３：Ｙｅｓ）、ＭＫＢ処理部２０３は、グループ操作メッセージ内のＭＫＢフラグメントを処理する（ステップＳ５０４）。

【００９５】

ＭＫＢ処理部２０３は、ＭＫＢ（ＭＫＢフラグメント）が正しく処理されたか否かを判定する（ステップＳ５０５）。正しく処理された場合（ステップＳ５０５：Ｙｅｓ）、グループ制御部２０４は、グループ操作メッセージ内のＧＩＤをＧＩＤ記憶部２２１に記憶するとともに、ＭＫＢ処理により得られたグループ鍵をグループ鍵記憶部２２２に記憶する（ステップＳ５０６）。正しく処理されなかった場合（ステップＳ５０５：Ｎｏ）、グループ制御部２０４は、グループ操作メッセージ内のＧＩＤをＧＩＤ記憶部２２１から削除するとともに、グループ鍵をグループ鍵記憶部２２２から削除する（ステップＳ５０７）。

【００９６】

このように、本実施形態にかかる通信制御装置では、スケーラビリティを確保しながら動的なグループ管理を実現することができる。また、グループ管理のために、ＭＫＢ全体ではなく、ＭＫＢを分割した情報（ＭＫＢフラグメント）を送信するため、通信負荷を軽減することができる。このとき、グループ操作の対象となる通信装置の範囲を定める情報とともにＭＫＢフラグメントを送信するため、意図しないグループ操作が行われることを回避できる。

【００９７】

（変形例１）

上記実施形態では、グループに属する葉ノードのみを含む部分木を求めながら、ＭＫＢを分割し、分割したＭＫＢ（ＭＫＢフラグメント）の範囲情報を生成した。変形例１では、先にグループに属する葉ノードのみを含むソートされた部分木のリストを求め、その後、部分木のリストを分割してＭＫＢフラグメントを生成するとともに、ＭＫＢフラグメントの範囲情報を生成する。

【００９８】

図１２は、変形例１の生成処理の一例を示すフローチャートである。生成部１０２は、左端の葉ノードおよび右端の葉ノードの一方から他方に向かって、ノードＩＤで識別される葉ノードのみをそれぞれ含む部分木を求める処理を繰り返して１以上の部分木を含むソートされた部分木のリストを求める（ステップＳ６０１）。ソートとは、インデックスの昇順または降順で並べることを意味する。例えば左優先の場合は、部分木のルートノードのインデックスRの左端または右端の葉ノードのノードインデックス（ＬＭＬ（Ｒ）またはＲＭＬ（Ｒ））が昇順となるように各部分木がソートされる。右優先の場合は、部分木のルートノードの左端または右端の葉ノードのインデックス（ＬＭＬ（Ｒ）またはＲＭＬ（Ｒ））が降順となるように各部分木がソートされる。

【００９９】

その後、生成部１０２は、求めた部分木のリストを、部分木をそれぞれ所定数（Ｍ（自然数））含む集合に分割し、集合それぞれに含まれる部分木の葉ノードに割り当てられたインデックスの範囲情報を求める（ステップＳ６０２）。

【０１００】

変形例１では、先に部分木のリストを求めた後に、Ｍ個の部分木を含む集合（部分木のリスト）をさらに求める。このため、計算量はＯ（Ｌ＋Ｌ／Ｍ）（Ｌは葉ノードの個数）となる。

【０１０１】

図１３は、ステップＳ６０１のリスト生成処理の一例を示すフローチャートである。生成部１０２は、使用する各パラメータを初期化する（ステップＳ７０１）。例えば、生成部１０２は、リストＳおよびリストＯを空（ＮＵＬＬ）に設定する。生成部１０２は、根ノードＲに対しＣｈｅｃｋ（）関数を実行する（ステップＳ７０２）。変形例１のＣｈｅｃｋ関数は、指定されたノードの子ノードに対して再帰的に実行され、結果としてリストＳが出力される。

【０１０２】

図１４は、左優先の場合の変形例１のＣｈｅｃｋ（）関数の一例を示すフローチャートである。図１４は、あるノードｎに対する左優先のＣｈｅｃｋ（）関数が呼ばれた場合の動作を示す。

【０１０３】

ステップＳ８０１からステップＳ８０７は、図７のステップＳ３０１からステップＳ３０７と同様であるため説明を省略する。

【０１０４】

ステップＳ８０７でｌｖａｌ×ｒｖａｌが０より大きいと判定された場合（ステップＳ８０７：Ｙｅｓ）、リストＳからノードｎの左の子ノードおよび右の子ノードが削除され、リストＳにノードｎが追加され、ｒｖに１が設定される（ステップＳ８０８）。

【０１０５】

ｌｖａｌ×ｒｖａｌが０より大きくない場合（ステップＳ８０７：Ｎｏ）、ステップＳ８０４の後、ステップＳ８０５の後、または、ステップＳ８０８の後、ｒｖの値を返して（ステップＳ８０９）、Ｃｈｅｃｋ関数が終了する。

【０１０６】

次に、ステップＳ６０２の範囲情報算出処理の一例を説明する。範囲情報算出処理では、リスト生成処理（ステップＳ６０１）で求められたリストＳを要素数Ｍごとのリスト Ｆに分割し、リストＦごとにリストＦに含まれる部分木の葉ノードのインデックスの下限値ｍｉｎｒおよびインデックスの上限値ｍａｘｒの組（ｍｉｎｒ，ｍａｘｒ）を算出する。そして、リストＦ、ｍｉｎｒ、ｍａｘｒの組のリストが出力される。この例では、リストＦが集合情報に相当する。

【０１０７】

ｉ番目（１≦ｉ≦Ｎ，Ｎは分割数，Ｎ＝ｃｅｉｌｉｎｇ（|Ｓ|／Ｍ））のリストＦのｍｉｎｒ、ｍａｘｒは以下のように算出される。

１番目のリストＦのｍｉｎｒ＝０

ｉ番目のリストＦのｍｉｎｒ＝（ｉ－１）番目のリストＦのｍａｘｒ＋１（ｉ＞１）
ｉ番目のリストＦのｍａｘｒ＝ｉ番目のリストＦの最後の要素（部分木）の右端葉ノードのインデックス　（ｉ＜Ｎ）
Ｎ番目のリストＦのｍａｘｒ＝Ｔ’の右端葉ノードのインデックス

【０１０８】
図１５は、左優先の場合の変形例１の生成処理を表す擬似コードの例を示す図である。変形例の場合も入力（Ｉｎｐｕｔ　Ｉ，Ｔ，Ｒ，Ｍ）および出力（Ｏｕｔｐｕｔ　Ｏ）は、図９と同様である。図１５のｆｒａｇｍｅｎｔ（Ｓ）が、範囲情報算出処理に相当する。

【０１０９】
（変形例２）
新規の通信装置２００（葉ノード）がグループに追加された場合に、生成部１０２が、追加後のグループに属する葉ノードのノードＩＤを用いて、生成処理を再度実行してもよい。また、グループから通信装置２００（葉ノード）が離脱した場合に、生成部１０２が、削除後のグループに属する葉ノードのノードＩＤを用いて、生成処理を再度実行してもよい。これにより、スケーラビリティを確保しながら動的なグループ管理を実現できる。

【０１１０】
次に、本実施形態にかかる通信制御装置のハードウェア構成について図１６を用いて説明する。図１６は、本実施形態にかかる通信制御装置のハードウェア構成を示す説明図である。

【０１１１】
本実施形態にかかる通信制御装置は、ＣＰＵ（Central Processing Unit）５１などの制御装置と、ＲＯＭ（Read Only Memory）５２やＲＡＭ（Random Access Memory）５３などの記憶装置と、ネットワークに接続して通信を行う通信Ｉ／Ｆ５４と、各部を接続するバス６１を備えている。

【０１１２】
本実施形態にかかる装置（通信制御装置、通信装置）で実行されるプログラムは、ＲＯＭ５２等に予め組み込まれて提供される。

【０１１３】
本実施形態にかかる装置で実行されるプログラムは、インストール可能な形式又は実行可能な形式のファイルでＣＤ－ＲＯＭ（Compact Disk Read Only Memory）、フレキシブルディスク（ＦＤ）、ＣＤ－Ｒ（Compact Disk Recordable）、ＤＶＤ（Digital Versatile Disk）等のコンピュータで読み取り可能な記録媒体に記録してコンピュータプログラムプロダクトとして提供されるように構成してもよい。

【０１１４】
さらに、本実施形態にかかる装置で実行されるプログラムを、インターネット等のネットワークに接続されたコンピュータ上に格納し、ネットワーク経由でダウンロードさせることにより提供するように構成してもよい。また、本実施形態にかかる装置で実行されるプログラムをインターネット等のネットワーク経由で提供または配布するように構成してもよい。

【０１１５】
本実施形態にかかる装置で実行されるプログラムは、コンピュータを上述した各部として機能させうる。このコンピュータは、ＣＰＵ５１がコンピュータ読取可能な記憶媒体からプログラムを主記憶装置上に読み出して実行することができる。

【０１１６】
本発明のいくつかの実施形態を説明したが、これらの実施形態は、例として提示したものであり、発明の範囲を限定することは意図していない。これら新規な実施形態は、その他の様々な形態で実施されることが可能であり、発明の要旨を逸脱しない範囲で、種々の省略、置き換え、変更を行うことができる。これら実施形態やその変形は、発明の範囲や要旨に含まれるとともに、特許請求の範囲に記載された発明とその均等の範囲に含まれる

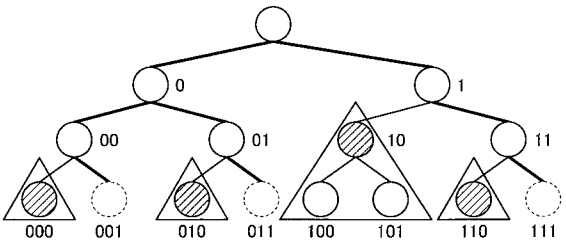
。

【符号の説明】

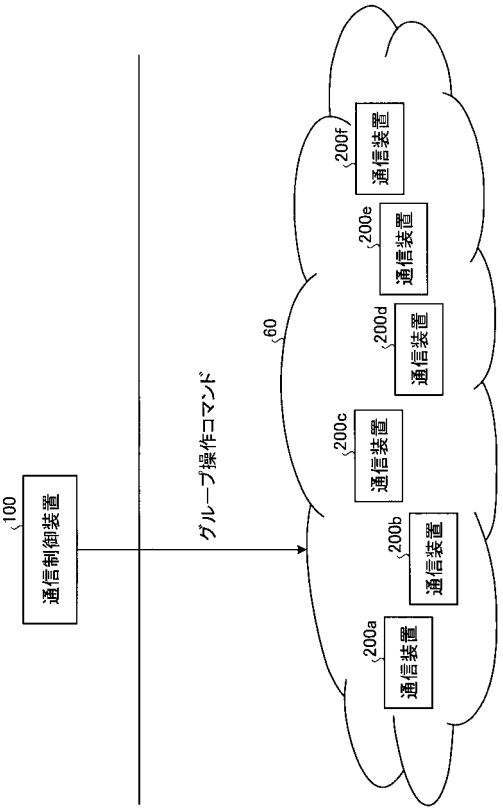
【０１１７】

１００　通信制御装置
１０１　受付部
１０２　生成部
１０３　出力部
１２１　グループ情報記憶部
１２２　アドレス記憶部
１２３　鍵記憶部
２００　通信装置
２０１　受信部
２０２　判定部
２０３　ＭＫＢ処理部
２０４　グループ制御部
２２１　ＧＩＤ記憶部
２２２　グループ鍵記憶部
２２３　デバイス鍵記憶部
２２４　デバイスＩＤ記憶部

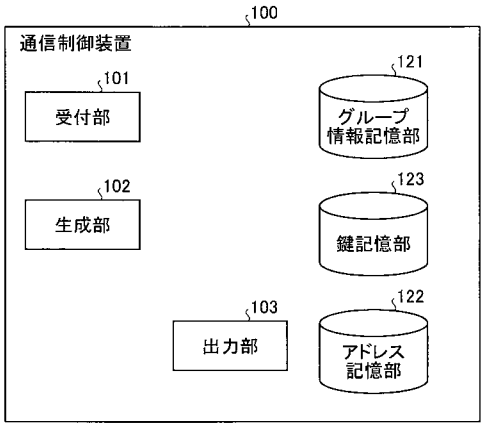
【図１】



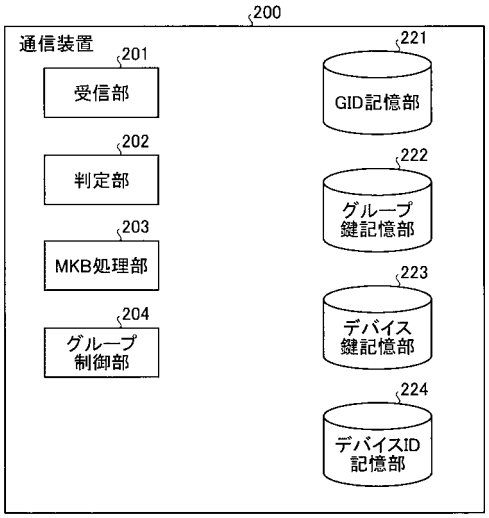
【図２】



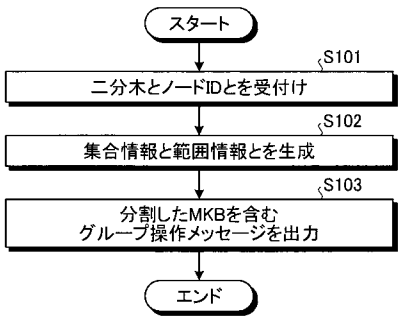
【図３】


100
通信制御装置
101
受付部
102
生成部
103
出力部
121
グループ情報記憶部
123
鍵記憶部
122
アドレス記憶部

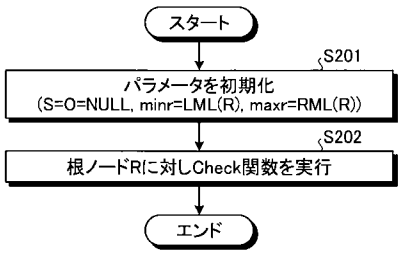
【図４】


200
通信装置
201
受信部
202
判定部
203
MKB処理部
204
グループ制御部
221
GID記憶部
222
グループ鍵記憶部
223
デバイス鍵記憶部
224
デバイスID記憶部

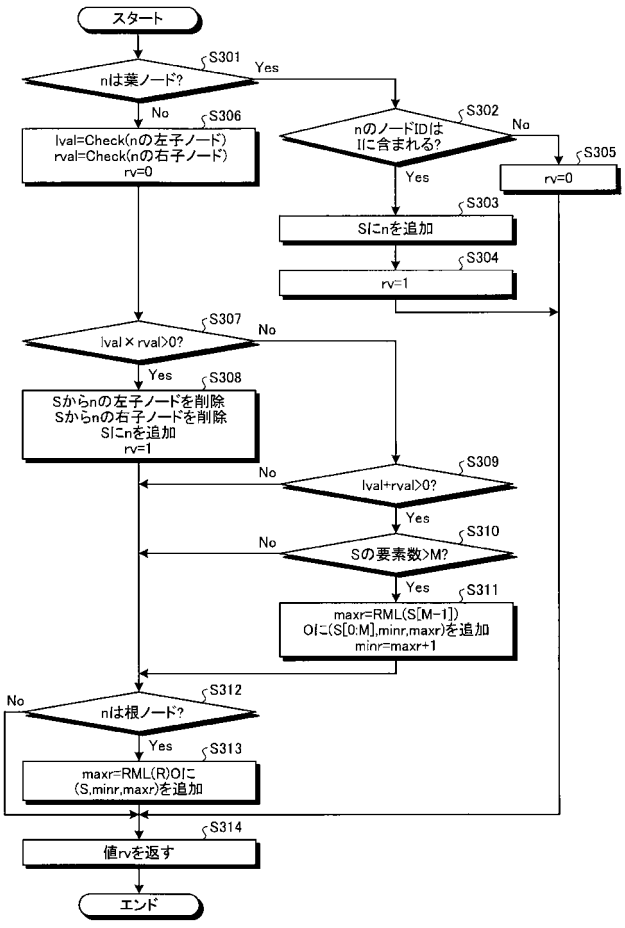
【図５】


スタート
S101
二分木とノードIDとを受付け
S102
集合情報と範囲情報とを生成
S103
分割したMKBを含む
グループ操作メッセージを出力
エンド

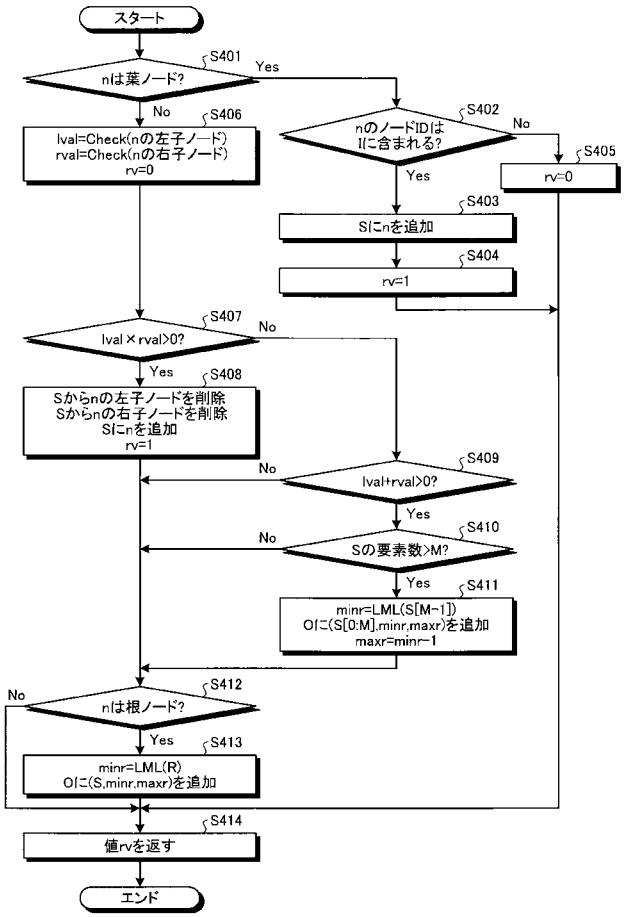
【図６】


スタート
S201
パラメータを初期化
(S=O=NULL, minr=LML(R), maxr=RML(R))
S202
根ノードRに対しCheck関数を実行
エンド

【図７】


スタート
S301
nは葉ノード?
Yes
No
S306
lval=Check(nの左子ノード)
rval=Check(nの右子ノード)
rv=0
S302
nのノードIDは
Iに含まれる?
No
Yes
S305
rv=0
S303
Sにnを追加
S304
rv=1
S307
lval×rval>0?
No
Yes
S308
Sからnの左子ノードを削除
Sからnの右子ノードを削除
Sにnを追加
rv=1
S309
lval+rval>0?
No
Yes
S310
Sの要素数>M?
No
Yes
S311
maxr=RML(S[M-1])
Oに(S[0:M],minr,maxr)を追加
minr=maxr+1
S312
nは根ノード?
No
Yes
S313
maxr=RML(R)Oに
(S,minr,maxr)を追加
S314
値rvを返す
エンド

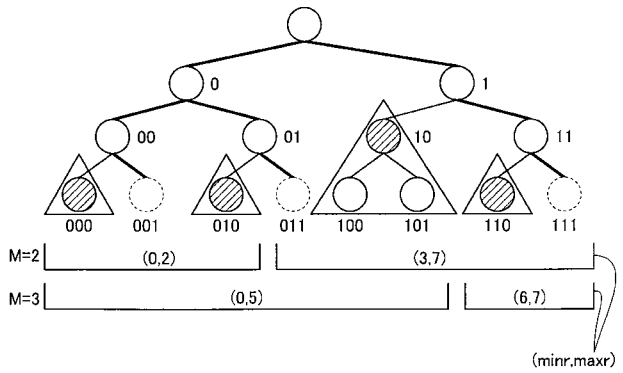
【図８】



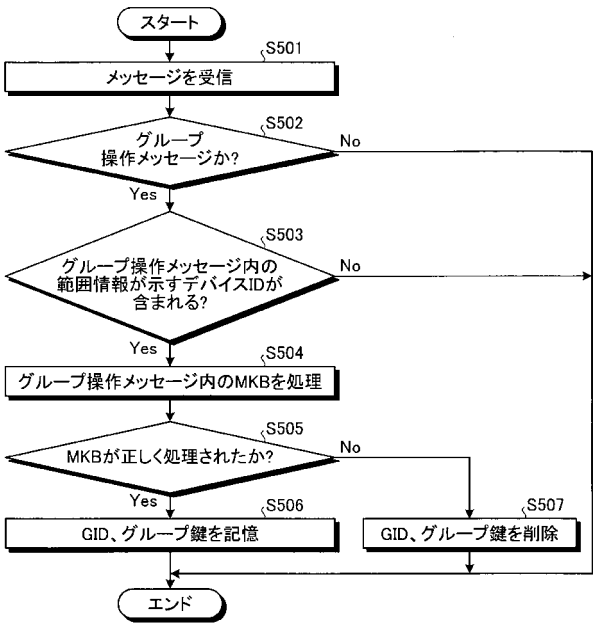
【図９】

```
def CreateCompleteSubtreeFragments(I, T, R, M):
        # Input I: List of indices of leaf nodes to be included in the group
        # Input T: The entire tree that covers all leaf nodes
        # Input R: Root node of the entire tree
        # Input M: Maximum number of subtrees in per fragment
        # Output O : List of (S, minr, maxr):
        #   S: Subtrees covering the group.
        #   minr: Lower bound of Subgroup Range
        #   maxr: Upper bound of Subgroup Range
        O=[]
        S=[]
        depth=int(math.log(len(T)+1,2)-1)

        def rightmost_leaf_number(n):
                # Input n: subtree root node
                # Output y: rightmost leaf number under the subtree
                h=n.index.len  # hierarchy level of node n
                x=int(n.index.val, 2)  # node index in decimal
                y=(x+1)*(2**(depth-h))-1
                return y

        def check(n):
                # Input n: subtree root node
                # Output 0, 1
                #  0 : Some node in the subtree is a non-member of the group.
                #  1 : All nodes in the subtree are members of the group.
                global minr
                rv=0
                if n.left==None and n.right==None:  # n is leaf
                        if n.index.val in I:
                                S.append(n)
                                return 1
                        return 0
                # n is non-leaf
                lval=check(n.left)
                rval=check(n.right)
                if lval*rval>0:
                        S.remove(n.left)
                        S.remove(n.right)
                        S.append(n)
                        rv=1
                elif lval+rval>0:
                        if len(S) > M: # one fragment is ready
                                maxr=rightmost_leaf_number(S[M-1])
                                O.append((S[0:M], minr, maxr))
                                S[0:M]=[]   # Remove the appended subtrees
                                minr=maxr+1
                        rv=0
                if n==R: # Root node. len(S)>0
                        maxr=2**depth-1
                        O.append((S, minr, maxr))
                return rv
        check(R)
        return O
```

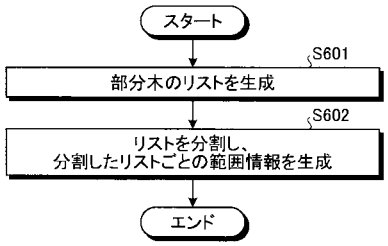
【図１０】



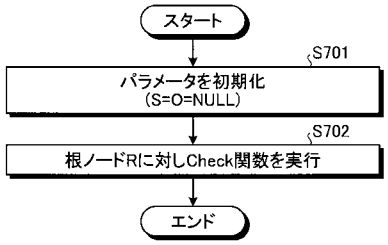
【図１１】



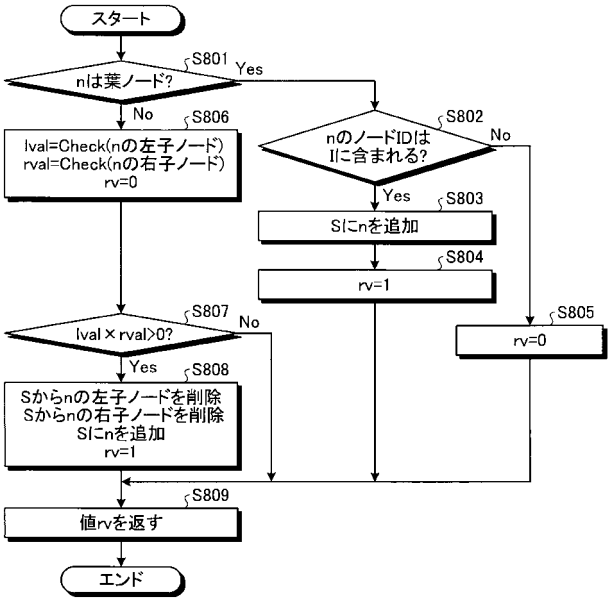
【図１２】



【図１３】



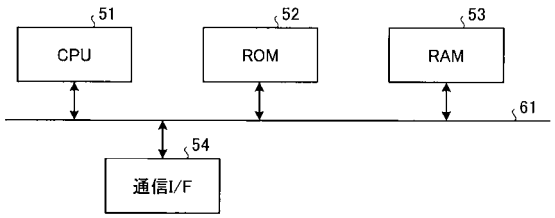
【図１４】



【図１５】

```
def CreateCompleteSubtreeFragments(I, T, R, M):
        # Input I: List of indices of leaf nodes to be included in the group
        # Input T: The entire tree that covers all leaf nodes
        # Input R: Root node of the entire tree
        # Input M: Maximum number of subtrees in per fragment
        # Output O : List of (S, minr, maxr):
        #   S: Subtrees covering the group.
        #   minr: Lower bound of Subgroup Range
        #   maxr: Upper bound of Subgroup Range
        O=[]
        S=[]
        depth=int(math.log(len(T)+1,2)-1)

        def rightmost_leaf_number(n):
                # Input n: subtree root node
                # Output y: rightmost leaf number under the subtree
                h=n.index.len  # hierarchy level of node n
                x=int(n.index.val, 2)  # node index in decimal
                y=(x+1)*(2**(depth-h))-1
                return y

        def fragment(S):
                minr=0
                N=math.ceil(float(len(S))/M)
                i=0
                while i<N:
                        if i+1<N: # Non-last fragment
                                F=S[i*M:(i+1)*M] # Extract M subtrees
                                maxr=rightmost_leaf_number(F[-1])
                        else: # Last fragment
                                F=S[i*M:] # Extract remaining subtrees
                                maxr=2**depth-1
                        O.append((F, minr, maxr))
                        minr=maxr+1
                        i=i+1
                return O

        def check(n):
                # Input n: subtree root node
                # Output 0, 1
                #  0 : Some node in the subtree is a non-member of the group.
                #  1 : All nodes in the subtree are members of the group.
                if n.left==None and n.right==None:  # n is leaf
                        if n.index.val in I:
                                S.append(n)
                                return 1
                        return 0
                # n is non-leaf
                lval=check(n.left)
                rval=check(n.right)
                if lval*rval>0:
                        S.remove(n.left)
                        S.remove(n.right)
                        S.append(n)
                        return 1
                return 0
        check(R)
        O=fragment(S)
        return O
```

【図１６】



フロントページの続き

Ｆターム(参考) 5J104 AA16 EA01 EA07 EA18 NA02 NA37
5K030 GA15 LD06