

本資料について

- ◆ 本資料は下記の論文を基にして作成されたものです。文章の内容の正確さは保証できないため、正確な知識を求める方は原文を参照してください。
- ◆ 著者：田村陽介
- ◆ 論文名：*Ameba*:大規模アドホックネットワークにおける効率的なアドレス割当て方式
- ◆ 発表日：May 2004

Ameba:大規模アドホックネットワークにおける 効率的なアドレス割り当て方式

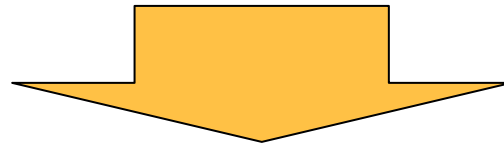
田村洋介

ソニーコンピュータサイエンス研究所

名城大学理工学部
小島崇広

はじめに

- ◆ ユビキタス時代の到来を迎えインターネットの基盤技術の見直しが迫られている
 - 通信端末の小型化・高性能化
 - インターネットの利用形態の大きな変化



- ◆ 場所や時間を気にせずあらゆる物がネットワーク化された環境での利用価値が注目されている技術としてMANETが議論されている

はじめに

- ◆ MANETはIPネットワークなのでIPアドレスを割り当てる必要がある
- ◆ IPアドレスを動的に割り当てる技術
 - BOOTP, DHCP
- ◆ これらはノードが常時サーバと接続しているネットワークを想定している
- ◆ ノードの移動が想定されるMANETでは特定のノードに依存したシステム設計をするべきではない
- ◆ MANET環境において動的に一意のアドレスを割り当てる手法Amebaを提案

Amebaの特徴

- ◆ 分散型アドレス割当て
 - ネットワーク内すべてのノードの通信到達性を保証できない環境でも利用可
 - アドレスを管理する特定のサーバを持たず分散して管理
- ◆ 制御パケットのフラッディング回避
 - 制御パケットは1ホップ通信でフラッディングを利用しない
 - ネットワーク負荷が少ない
 - 消費電力の観点からも効率的
- ◆ アドレスの唯一性の保証
 - アドレス割当てが階層的

関連研究

- ◆ 重複アドレス検知手法DAD (Duplicate Address Detection)
- ◆ MANETconf
- ◆ 重複アドレス検知手法DAD (Duplicate Address Detection) について
 - IETFのZeroconfワーキンググループで議論されている
 - ノードが最初にアドレスを決定しネットワーク内で重複したアドレスがないか確認する方式
 - この手法はMANETを想定していない
 - すべてのノードがリンク層のブロードキャスト、マルチキャストのメッセージを受信できる。または、一台のルータでつながれた複数のネットワークセグメントを想定
 - PerkinsらによってMANETに適用

DADの仕組みをMANETに適用

- ◆ アドレス取得までの流れ
 - 新規ノードはランダムに利用候補のアドレスを決定
 - 利用可否を問うAREQをMANET全体にフラッディング
 - そのアドレスを使用しているノードは新規ノードに対してAREPを送信して使用中だと知らせる
 - AREPが一定期間内に届かなければそのアドレスを利用
- ◆ 欠点
 - フラッディングを利用している
 - 到達可能なMANET内でのアドレスの唯一性しか保証できない
 - 一時的にネットワークを離れていたノードはAREQを受信できない

MANETconf

- ◆ アドレス取得までの流れ
 - 新規ノードが近隣クエリをブロードキャスト
 - MANET内ノードから近隣応答を受け取る
 - 近隣応答を受け取ったノードの中から一つを選択
 - 選択されたノードは利用されていないアドレスの候補をMANETにフラッディング
 - 他のノードから否定確認がなければ新規ノードに割当て
 - アドレスが決定したことをMANET全体にフラッディング
- ◆ 欠点: アドレス割当てにフラッディングを2回使用

Amebaについて

◆Amebaが想定するMANET

◆ 大規模MANETを想定

◆ 初期ノードの存在を想定

- MANETで利用するアドレスの範囲・有効時間を設定

◆ ノードの移動を想定

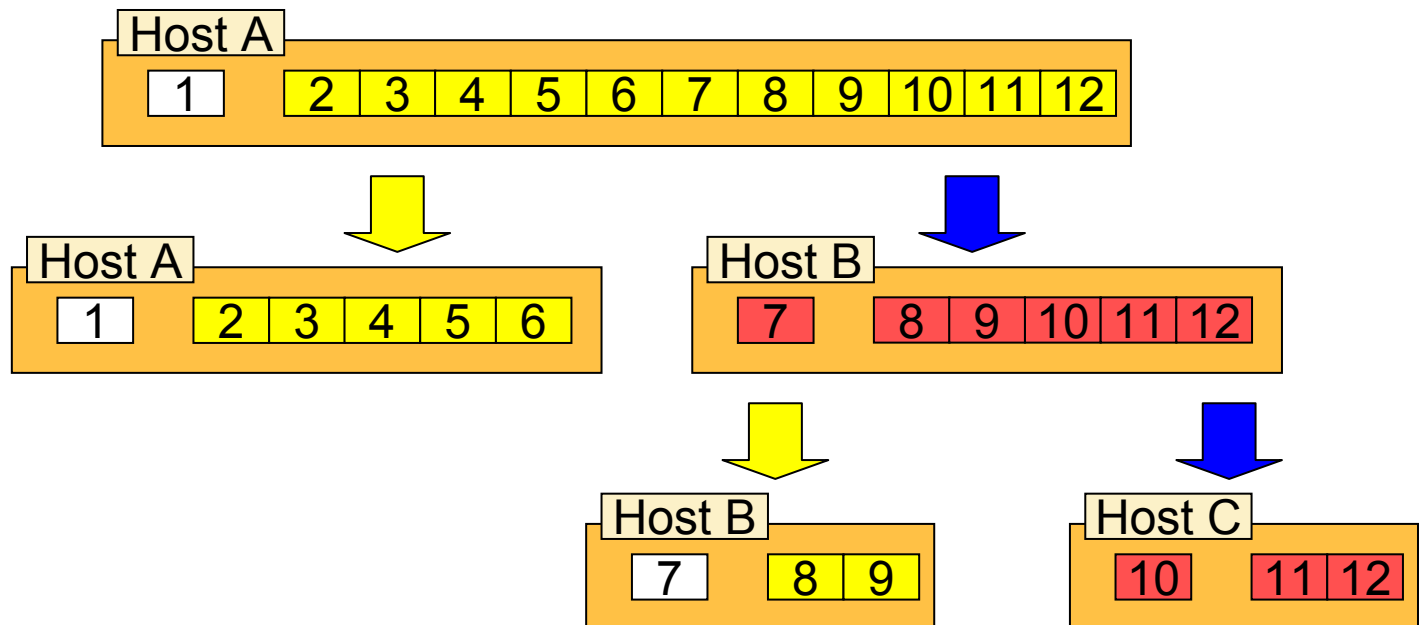
- ノードからのMANET離脱宣言があるまではどこに移動してもMANET構成ノードの一部
- ノードの移動に制限がなく, 他ノードとの通信が可能なら移動速度に制限がない

アドレス割当て方式

- ◆ 3つの方式をネットワーク内で混在して利用
 - ブロック割当て
 - 動的割当て
 - 自動割当て
- ◆ 割当て手法の選択はアドレス要求ノードが決定、割当てノードがその可否を判断
- ◆ Amebaのノードは自らのアドレスのほかに貯蔵アドレスをもちアドレスの割当てができる

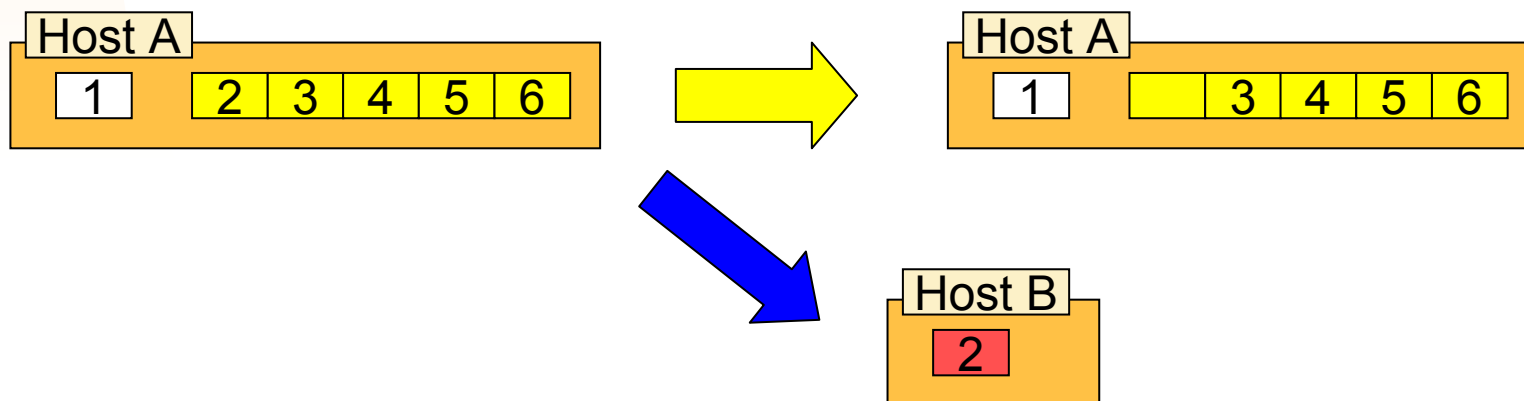
ブロック割当て

- ◆ アドレスを割り当てるのと同時に貯蔵アドレスを割り当てる
- ◆ 貯蔵アドレスの後半部を割り当てる
- ◆ 割り当てられたアドレスをさらに割り当てることが可能



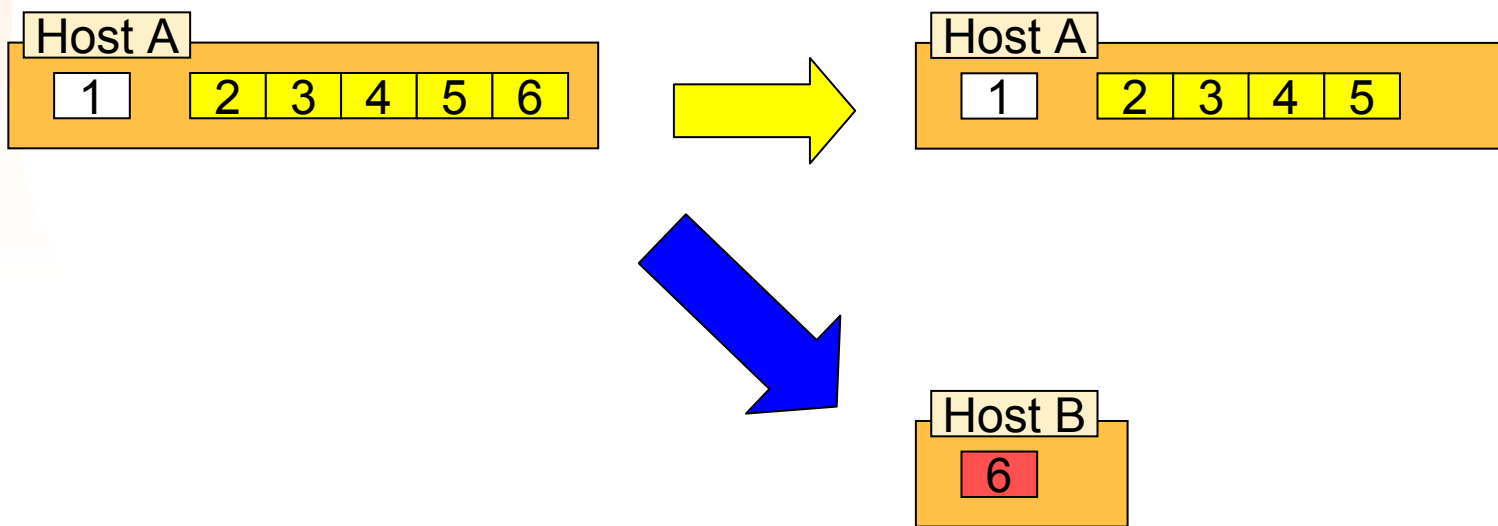
動的割当て

- ◆ 貯蔵アドレスの先頭をリース期限付きで割当ててる
- ◆ アドレス更新でアドレスを利用し続けることが可能
- ◆ MANETでは割当てノードとの接続性が保証されない
 - 割当てノードとのみ一時的に通信する場合
 - お互いが常時接続可能な範囲の場合
- ◆ 利用例
 - グループなどでリーダーとなるノードがいる場合にメンバーにアドレスを割当てるとき



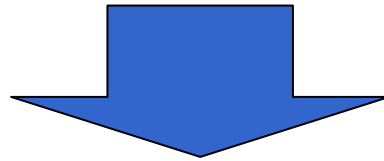
自動割当て

- ◆ 貯蔵アドレスの最後尾を割当てる
- ◆ 割り当てられたアドレスは返還されない
- ◆ 利用例
 - MANETに参加するノードが入場門のように特定の場所を経由する状況に適している
 - ほかのノードにアドレスを割当てる処理を避けたいとき



アドレス解放手法

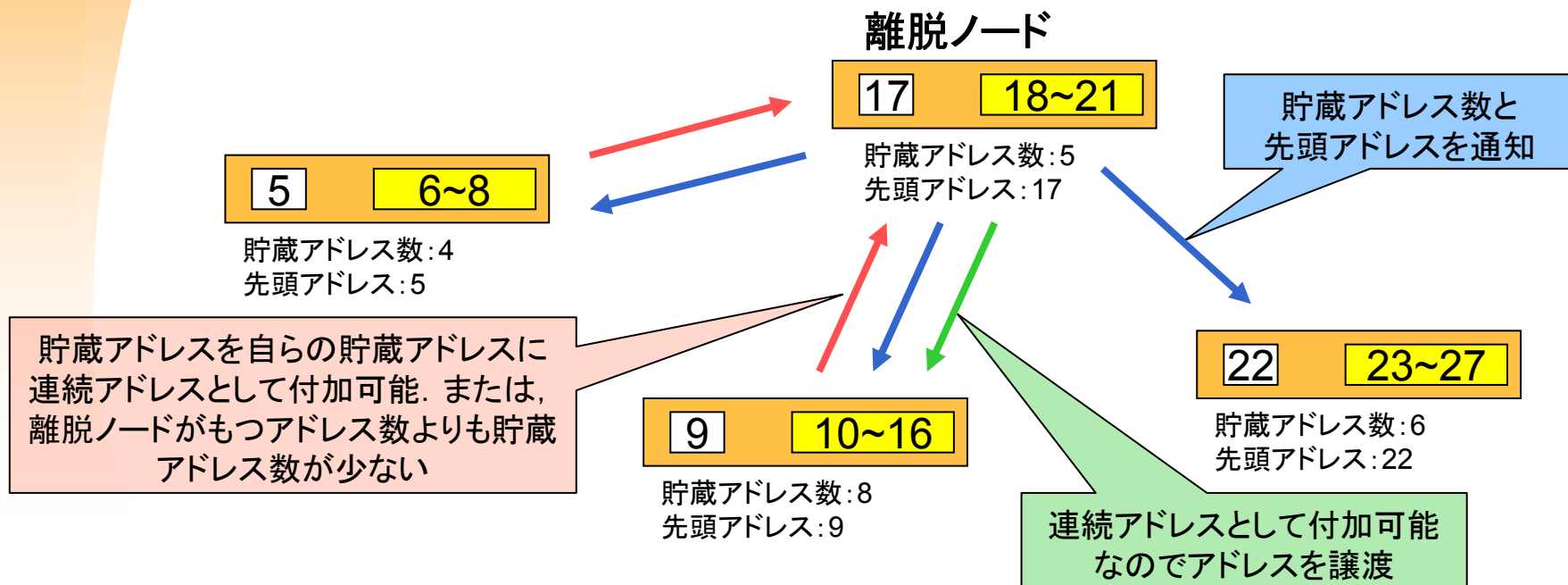
- ◆ ノードは常時MANETとの接続が保証されているわけではない
 - 貯蔵アドレスを持つノードの離脱はアドレス資源の枯渇を招く



- ◆ 初期ノードは貯蔵アドレスに有効時間を設定
 - 有効時間はMANET内で同時刻にタイムアウト
- ◆ 最終的に初期ノードがすべてのアドレスを回収

有効時間内でのアドレス解放手法

- ◆ Amebaでは明示的なアドレス解放のみを提供
 - 他ノードによって離脱が判断されアドレスを再利用するとアドレスの重複を招くことになる



離脱ノードの貯蔵アドレスが最も少ない場合はその貯蔵アドレスは有効期限内では利用されない

メッセージ構成

Mtype	Rtype	Tid(1)	LTime(2)
Saddr(4)			
Aaddr(4)			
Highaddr(4)			
Lowaddr(4)			
LLaddr(16)			

()内はオクテッド

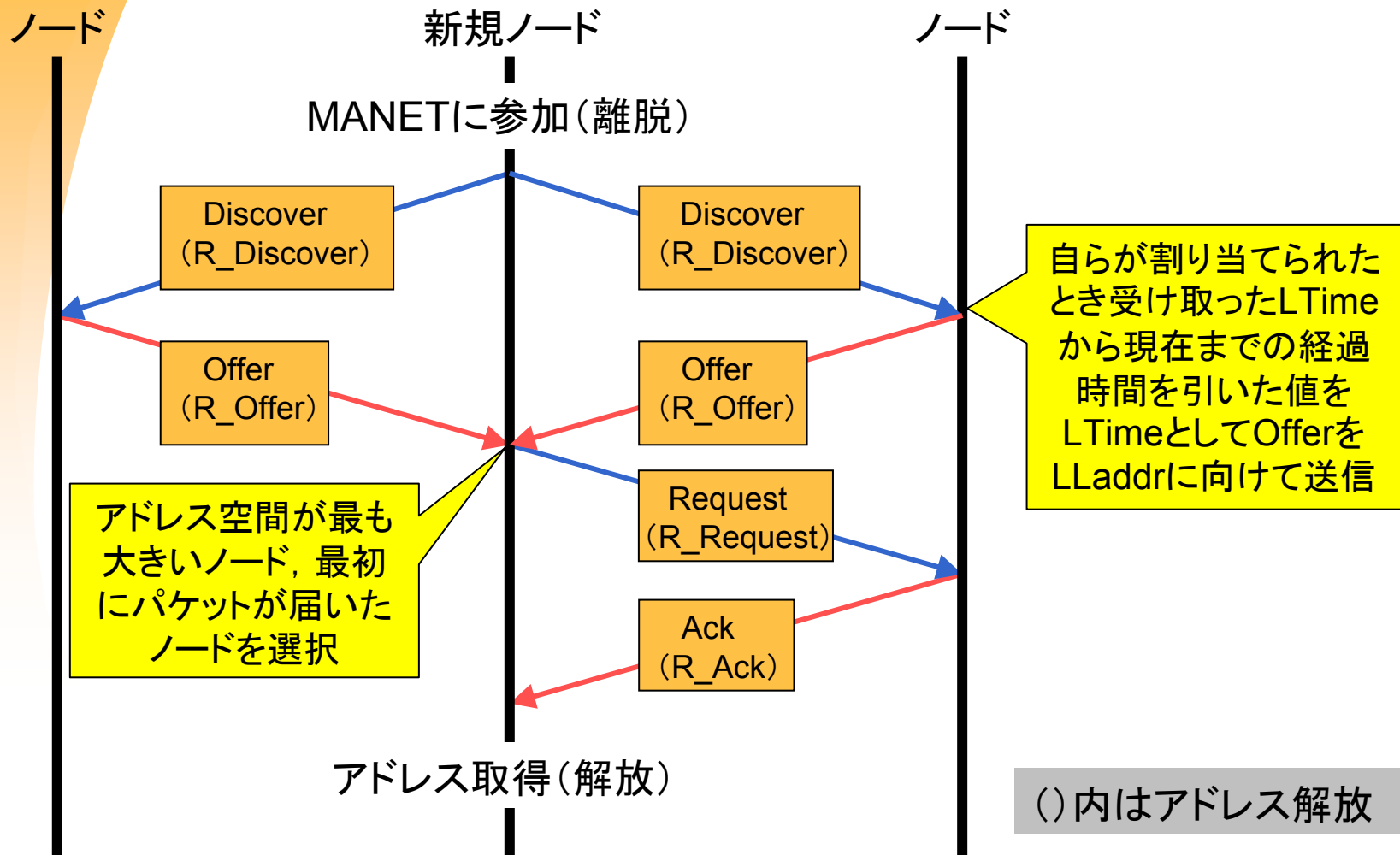
メッセージタイプ	
Discover Offer Request Ack	アドレス割当てに使用
Lease	動的割当てでアドレスを更新するのに使用
R_Discover R_Offer R_Request R_Ack	アドレス解放に使用

- ◆ Mtype:メッセージタイプ
- ◆ Rtype:MtypeがDiscoverのときだけ有効. 割当て方式の選択に使用
- ◆ Tid: 他のアドレス割当てパケットと区別するのに使用
- ◆ LTime:有効時間を定義
- ◆ Saddr:アドレスを割り当てるノードのアドレスを格納
- ◆ Aaddr:割り当てるアドレスを格納
- ◆ Highaddr, Lowaddr:割り当てる貯蔵アドレスの先頭と最後尾のアドレス値を格納
- ◆ LLaddr:リンクローカルアドレスを格納

初期ノード

- ◆ ネットワーク構築を要求する初期ノードが存在が必要不可欠
 - ネットワークを利用してサービスを提供する組織
- ◆ 初期ノードの設定
 - MANET全体で使用するアドレス空間の大きさ
 - アドレスの有効時間
- ◆ 連続的なアドレス空間

新規ノードへのアドレス割当て



制御パケット数の比較

◆ 想定条件

- すべてのノードはマルチホップで到達可能
- ノード数: n
- 電波到達距離内の平均ノード数: k
- ブロードキャストパケットは1パケットとする

フラッディング方式

送信処理: $n(n+1)/2$ 回

受信回数: $kn(n+1)/2$ 回

送受信数: $(k+1)n^2/2 + (k+1)n$

Ameba

送信処理: $n(2+k) + \log_k n$ 回

受信回数: $kn(2+k) + k \log_k n$ 回

送受信数: $(k^2 + 3k + 2)n + (k+1) \log_k n$

よって

フラッディング方式では二乗に比例してネットワークトラフィックが増加

Amebaでは直線的にトラフィックが増加

することがわかる

k が固定された場合, n が増加するに従いAmebaがフラッディングと比べてより少ない数のパケットになる.

例 $k=2, n=500$ のときフラッディングはAmebaに比べて60倍以上のパケットが必要

シミュレーションによる性能解析

- ◆ 一辺がMブロックの正方形の領域をN台のノードが動き回る環境を想定
- ◆ ノードの移動にはRandomWayPointModelを利用
 - ノードが任意に目的を設定し,1ティック3ブロックの速さで移動.目的地に到着すると10ティック停止.その後同じ動作を繰り返す
- ◆ 一台の電波到達距離に存在するノードの平均を4ノード
- ◆ 電波到達距離は50ブロック

- ◆ 以上の環境でフラッディング方式とAmebaのパケット数と消費電力についてシミュレーションを行う

ネットワーク全体のパケット数

- ◆ フラッディング方式
 - ネットワーク全体としてパケット数が二乗に比例して増加
- ◆ Ameba
 - ノード数が増加してもネットワーク全体のパケット数はほぼ直線的に増加
- ◆ これより二乗に比例して増加するフラッディング方式と比べてスケーラビリティがあるといえる

消費電力の比較

- ◆ フラッディング方式
 - 全体のホスト数が増加すると全消費電力が二乗に比例して増加
- ◆ Ameba
 - ホストあたりの消費電力は一定. 全体の消費電力も全体のホスト数に対して直線的に増加
- ◆ これよりAmebaは消費電力の点でも従来のフラッディング方式よりも効率的であるといえる

考察と今後の課題

◆ 貯蔵アドレスの分配方法

- 現在はブロック割当てでは貯蔵アドレスの半分を割り当てているが、ノード密度の高い範囲に貯蔵アドレスが多く分配されると効率的
- 分配方法を最適化することが今後の課題

◆ セキュリティ技術との融合

- Amebaはセキュリティに関して触れていない。現在ではMANETに加わりたいノードすべてにアドレス割当てを許可してしまう
- 安全性の高いアドレス割当てが今後の課題

まとめ

- ◆ 従来方式の多くはアドレス割当てにおいて唯一性を保証するためにフラッディングを利用している。しかし、フラッディングでは完全な唯一性を保証することは出来ない。本稿のAmebaではアドレスを割り当てた時点でアドレスの唯一性を保証できる。
- ◆ シミュレーション実験よりフラッディングを利用した方式と比べAmebaでは唯一性を保証するだけでなくノード数が多いネットワークにおいてアドレス割当ての際に利用するパケットを大幅に減少させ、その結果消費電力を減少させることがわかった。