

# MANETにおけるモバイルDBデータ転送時の中継ノード高機能化

松井愛子<sup>†</sup> 神坂紀久子<sup>†</sup>  
山口実靖<sup>††</sup> 小口正人<sup>†</sup>

モバイルDB機能を持つノードから構成される無線LANマルチホップ通信において、データ中継ノードに単にデータの中継するだけでなく、中継データを自身のDBにキャッシュしたり、アグリゲーションを行う等の高度な処理をさせる手法を検討する。無線LANは帯域が狭く下位層の処理に比べ上位層の処理に余裕があり、中継ノードにそのような高度な処理をさせても通信性能にさほど影響を及ぼさない、ということが予想される。本研究では従来のディスクタイプDBではなく、よりモバイルDBに適すると思われるインメモリの組み込みDBをモバイル端末に搭載し、無線LANマルチホップ通信環境において中継ノードにDB処理をさせた場合とさせなかった場合について通信性能を比較する。

## High Performance Relay Node on Mobile DB Data Transfer in MANETs

AIKO MATSUI,<sup>†</sup> KIKUKO KAMISAKA,<sup>†</sup> SANEYASU YAMAGUCHI<sup>††</sup>  
and MASATO OGUCHI<sup>†</sup>

In Wireless LAN multi hop communication performed by mobile DB node, relay nodes can not only act as a router but also execute advanced DB processing, like data caching and data aggregation. Since the bandwidth of wireless LAN is narrow, processing of higher layer has enough time compared with that of lower layer. So we think communication performance would not be influenced even if the relay node does advanced processing. We adopt not only traditional disk-DB but In-memory DB that is more suitable for mobile environment. We have executed experiments on both situation; the relay node does advanced DB processing, and the relay node acts as only a router. And we have analyzed the difference of communication performance between two cases.

### 1. はじめに

ノートPCやPDA、携帯電話などといった移動体端末の高機能化、小型化に伴ってこれらの機器が社会に急速に浸透し、それに伴い日常的にデータベースを携帯することも可能となった。これらのモバイルDBは有線および無線通信によって、お互いにデータの同期を取ったり、複数のDB間でデータを収集、分散したりするなどのやりとりが可能である。これらのやりとりは有線環境、無線環境、単一方向、相互方向、複数端末間などさまざまな通信形態においてなされることが予想される。

近年、無線通信に対する需要が益々高まり、様々な場面における種々の形態での活用が期待されている。無線通信の利用形態のひとつに、既存の通信インフラに

頼らずに移動無線端末同士で動的にネットワークを構築する、モバイルアドホックネットワーク(MANET)がある。現在MANETを用いたセンサネットワーク<sup>1)</sup>など様々な分野においてMANET関連の研究が多くなされており、MANET環境におけるモバイルDBについても多くの研究がなされている<sup>2)</sup>。

またMANETにおいて期待される手法として、マルチホップ通信がある。マルチホップ通信を用いれば、端末同士が直接通信するだけでなく、他の端末を経由することでより広い範囲の端末と複数の経路を介した通信が可能となる。

現在無線LANの技術は進歩しており、帯域幅も広がってきている。しかし、例えばこのようなセンサネットワークやモバイルDB等で用いられる無線ネットワークはノードの数が多いことや、バッテリーの問題、扱うデータのサイズ等を考えると、あまり帯域幅の大きな無線LANは使われない可能性も高い。

筆者らは、これまでに無線LAN通信および、無線LANを含むマルチホップ環境等におけるTCPパ

<sup>†</sup> お茶の水女子大学  
Ochanomizu University  
<sup>††</sup> 工学院大学  
Kogakuin University

ラメータの振舞やスループット等を詳細に調べてきた<sup>3)4)5)</sup>。その結果、有線 LAN と比べ極めて小さい無線 LAN の帯域幅が、無線 LAN 独自の通信を行わせることが分かった。

その経験から、無線 LAN の帯域幅が小さい時には、無線 LAN をインタフェースとして持つ端末は下位層の通信処理に比べ上位層において余裕がある場合が多いことが予想される。従ってマルチホップ通信におけるデータ中継ノードに、データをキャッシュしたりアグリゲートする等の高度な処理を行わせたとしても、通信性能にはさほど影響が出ない可能性がある。

そこで、本論文では、無線 LAN マルチホップ通信環境において、中継ノードに単にデータを中継させる場合と、中継ノードに高度な処理を行わせた場合との通信性能の差について検討する。まず始めに、中継ノードの IP 層で単にデータを中継させる場合と中継ノードのアプリケーション層にデータを read/write(send/receive) させた場合の通信性能の差を評価する。次に無線 LAN マルチホップ通信環境において、2 台の端末の DB 間でデータ転送を行う際に、中継ノードに単にデータを中継させる場合と、中継ノード自体に DB 動作（データキャッシュ）を行わせた場合との通信性能の差を評価する。そしてそれらの実験結果を踏まえた上で今後の展望について議論を進める。

## 2. 研究背景

### 2.1 MANET に関する技術動向

昨今の無線通信技術の発展と計算機の小型化に伴い、ルーティング機能を持つ移動体のみで一時的なネットワークを構築する MANET への関心が高まっている。MANET では個々のノードのワイヤレス回線の範囲が近距離に限定されているのに対し、あるノードが自身のワイヤレス回線の到達範囲外にあるノードと通信する場合は、図 1 に示すようにあるノードが送信したパケットを他のノードが転送するマルチホップ通信によって実現される。

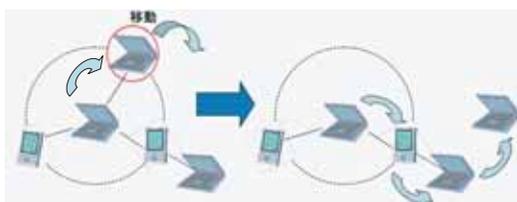


図 1 MANET におけるマルチホップ通信

MANET を用いたセンサネットワークやモバイル

DB などの研究も盛んに行われている。センサネットワークとは超小型無線装置を内蔵した多数のセンサを相互に連携させるネットワークのことで、人およびモノの状況や周辺環境などをセンサが認識し、センサ同士の自律的な情報の通信によって状況へのリアルタイムな対応を可能とするものである。セキュリティ、食品、医療、福祉などの多岐にわたる分野での活用が期待されている。例えば、家庭内に設置したセンサで高齢者の脈拍や血圧などのデータを収集し高齢者の健康状態を把握をしたり、自然環境の状態を監視する環境テレメータシステムの構築といったことが可能になる。

また MANET 環境に分散したノードがそれぞれデータベースの一部を保有する状況で、問い合わせ処理やデータ収集などを行うモバイル DB の研究も盛んに行われている。この場合にも、データを転送して収集する時などに、マルチホップ通信が活用される。

### 2.2 輻輳ウィンドウ

TCP では、輻輳制御において輻輳ウィンドウという概念を用いている。Linux OS の TCP は状態機械として実装されており、パケット送受信における処理は、その状態によって変化する。

輻輳ウィンドウとはネットワークの輻輳制御を目的としてデータ送信側が自主的に制限するための TCP パラメータで、受信側からの確認応答パケット (Acknowledgement: ACK) 無しに連続送信できる最大のパケット数を表す。正常な通信時には確認応答を 1 つ受信するごとに輻輳ウィンドウが増加し、エラーが起ると急激に減少する。エラーには

- ・送信側デバイスドライバのバッファが溢れることによる Local Congestion エラー (CWR)
- ・重複 ACK, SACK を受信 (Recovery)
- ・タイムアウトを検出 (Loss)

の 3 つがある。これらのエラーにより輻輳ウィンドウが急激に減少した後、回復と判断されると再度正常な状態に遷移し増加を始める。単位時間あたりに受信する確認応答の個数によって差はあるものの、輻輳ウィンドウの減少時に比べると緩やかに増加していくため、TCP 輻輳ウィンドウ制御において、輻輳ウィンドウの時間変化は鋸型となる。また、Linux の TCP 実装では、通信中にひとたび設定された輻輳ウィンドウは、その値を使い切らない限りは変化せず、その間のスループットはほぼ一定の値で安定する、ということが分かっている。

### 2.3 既存研究

筆者らはこれまで、無線 LAN 通信環境における TCP パケットの振舞を可視化し、解析を重ねてきた。

具体的には、図2に示すように、まずカーネル内部のTCPソースファイルにモニタ関数を挿入しカーネルを再構築することで、TCPパラメータのリアルタイムの可視化を実現した。さらにパケット転送タイミングとスループット時間変化を可視化し、これらを元に有線LAN、無線LAN環境における、あるいはその両方を含むマルチホップ環境におけるTCPパラメータの振舞を比較、解析してきた。

無線LAN環境においては広告ウィンドウサイズを変えることで輻輳ウィンドウを制御し通信効率を上げ<sup>3)</sup>、有線LANと無線LANからなる無線LANアクセスポイントの使用を想定した無線LANアクセス環境においては、現実の通信で輻輳ウィンドウがどのように振舞うかを解析することが出来た<sup>4)</sup>。

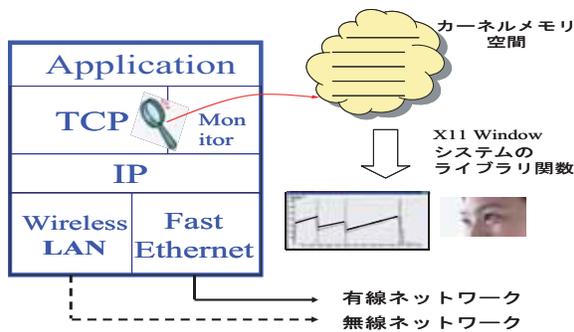


図2 TCPパラメータの可視化手法の概念図

### 3. 中継ノードの高機能化

#### 3.1 マルチホップ通信に関する考察

2.で述べたような、データの分散および収集処理における通信環境の例として、モバイルノード群がデータを収集してデータ管理端末がそれを回収したり、逆にデータ管理端末があるデータを分散したモバイルノード群に一齐に送信する、すなわちデータ管理端末がデータのaggregation & disseminationを行う図3のような場面が想定される。

ノードAがデータ管理端末でインターネットに接続しており、モバイル環境へのゲートウェイにもなっている。ノードB~ノードDがモバイルノード群で、ノードBは他のモバイルノード群のデータ中継の役割も果たす。

このような場面において、ノードBのような中継ノードは、単にデータを中継するだけではなく、図4のように中継データをキャッシュしたり、各ノードが集めてきたテーブルをジョインしてからデータ管理端末に送るなどといったデータのアグリゲートを行うこと

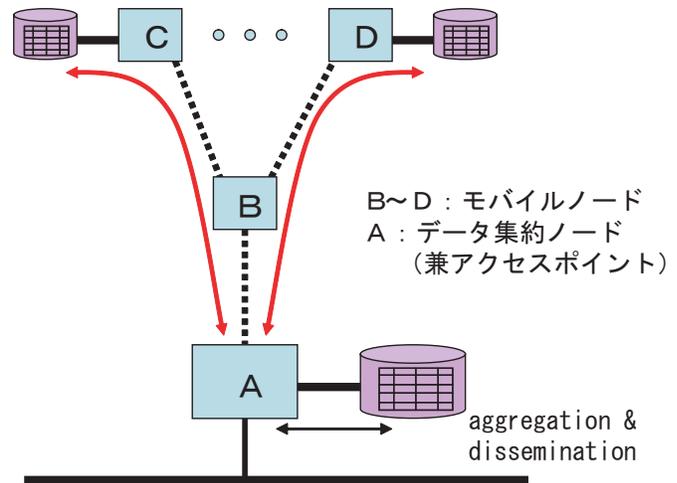


図3 無線LANマルチホップ環境一例

も可能である。MANETは無線LANの一部の区間が断続的に切断してしまうような状況が多く起こると考えられるが、その場合このような形態は極めて有効である。

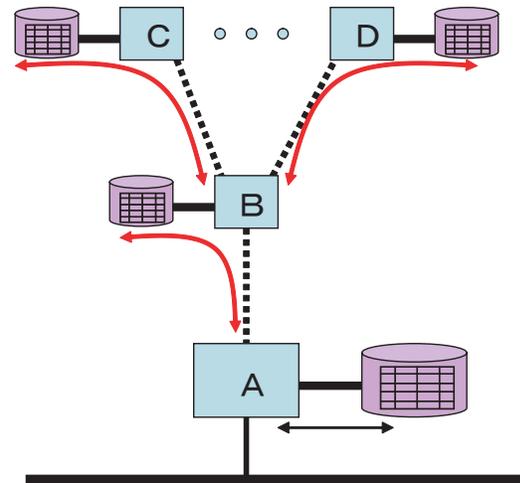


図4 中継ノードの高機能化

また、無線LANは有線LANと異なり、帯域に大きな制約があることから、通信は出来るだけ節約し、中継ノードの機能を出来るだけ高機能化させる事が有効なのではないかと考えられる。

これらのことから、MANETやセンサなどの環境において、中継ノードは上位層における処理を例えば

- (1) マルチホップ仲介
- (2) データキャッシュ
- (3) データアグリゲート

というように高機能化していくことが望ましいと考えられる。

### 3.2 本研究の実験概要

一般に、マルチホップ通信において中継ノードにアプリケーション層で高度な処理を行わせた場合、アプリケーション層までデータを引き上げての処理に時間がかかるため、中継ノードに IP 層で折り返してデータの中継のみをさせる場合よりもスループットが低下することが予想される。そこで本研究では、モバイル中継ノードを単にデータを中継するノードとして動作させた場合と、中継ノード自身がアプリケーション層で処理を行ってからデータを中継する場合とでどのくらい通信性能に差が出るのか実験を行い、TCP パラメータの振舞、スループット、パケットの流れなどを元に解析を行う。その結果を元に中継ノードの高機能化について、今後の可能性を探る。

## 4. 通信性能評価

### 4.1 中継ノードにおける read/write 処理

#### 4.1.1 実験概要

まず、中継ノードにおいてアプリケーション層で read/write 処理をさせた場合とさせずに IP 層でデータの中継のみを行う場合の比較実験を行う。実験環境は図 5 に示すように 3 台の端末を IEEE802.11b 無線 LAN のツーホップ通信で接続したものをを用いた。実験においてはデータ送信側ノードおよび受信側ノードがお互いに直接ではなく必ずマルチホップ通信でパケットをやりとりするように、iptables コマンドによって、中継ノード経由以外のパケットを遮断するようにした。各マシンのスペックを図 6 に示す。

両端のノードのデータのやり取りとしては、送信側が受信側に接続要求を出し、受信側が許可すると、送信側が一定のブロックサイズのデータを繰り返し送信し続けるという形のソケット通信プログラムを用いた。このような環境において、中継ノードが受信したデータを自身のアプリケーション層で read/write してから転送（データを一旦 receive してから次のノードへ send）するようにさせた場合と、受信したデータをマルチホップ用ルーティングテーブルを元に IP 層で折り返し、すぐにもう一方の端末に転送するというように中継ノードをルータとしてのみ動作させた場合の、両方について測定を行った。前述の場合、パケットが通るプロトコルレイヤは図 7 の (1) のようになり、後述の場合は (2) のようになる。

#### 4.1.2 通信性能測定

実験の結果を図 8 ~ 図 13 に示す。



図 5 実験環境

	CPU	メインメモリ	OS
NodeA NodeB	Pentium III 800MHz	640MB	Linux 2.4.18-3
Relay	Pentium M 1.3GHz	512MB	Linux 2.4.18-3

図 6 マシンスペック 1

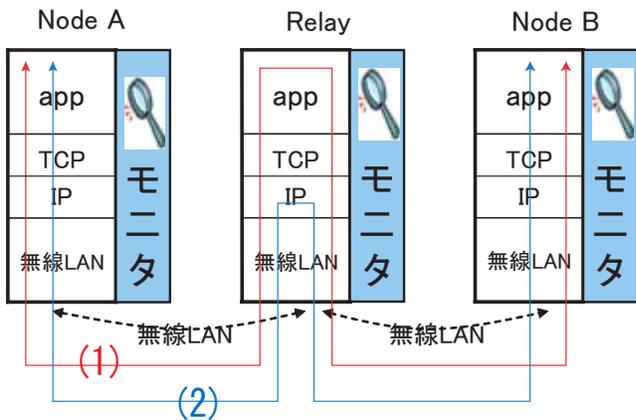


図 7 パケットの転送経路

図 8 は横軸がブロックサイズ [KB]，縦軸がスループットの平均値 [Mbps] を表している。ブロックサイズは 1KB から 1024KB まで 9 段階の値で測定した。中継ノードに read/write 処理をさせた場合とさせなかった場合の両方ともブロックサイズによってスループットの値に変化は見られなかった。また、中継ノードに read/write をさせた場合とさせなかった場合でスループットの差は殆ど見られなかった。

図 9 は中継ノードに read/write をさせなかった場合、図 10 は read/write をさせた場合の輻輳ウィンドウの時間変化のグラフである。ブロックサイズは 1024KB である。グラフ中の縦の点線は重複 ACK 受信エラーが発生したことを示している。すなわち、中継端末に read/write をさせた場合もさせなかった場合も同様に、パケットロスによる重複 ACK が頻発していることが分かる。

図 11 はブロックサイズを 1024KB としたときのスループットの時間変化のグラフである。横軸が時間 [sec]，縦軸がスループット [Mbps] を表している。中継

ノードに read/write をさせた場合とさせなかった場合では、read/write をさせなかった場合の方が多少スループットは大きいものの、両者に殆ど差は見られなかった。また、両方とも輻輳ウィンドウの時間変化と異なり、スループットの時間変化には大きな変動は見られず、安定して一定のスループットを出していることが分かった。

図 12 は中継ノードに read/write をさせなかった場合、図 13 は read/write をさせた場合の packets traces の図である。各端末において tcpdump コマンド<sup>6)</sup> を用いてパケットをキャプチャしパケットの送受信の様子をグラフ化して可視化した。横軸が経過時間、縦軸がノードを表していて、各パケットが各ノードをどのタイミングで通過しているかが分かる。中継ノードにおいて read/write をした方が read/write をしなかった場合より受け取ったデータを転送するのに時間がかかっていることが分かる。しかし、それは端末間のデータ転送にかかる時間と比べるとわずかな差と言える。また、パケットの転送は大体同じペースでされており、パケット送信タイミングがほぼ同じであることが分かる。

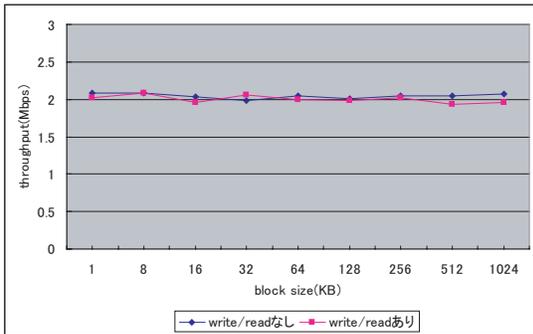


図 8 blocksize に対するスループット (read/write)

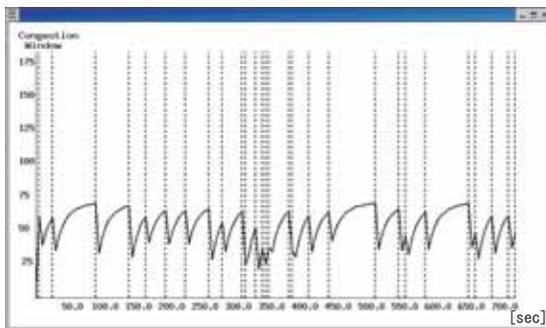


図 9 read/write なしの場合の輻輳ウィンドウ時間変化

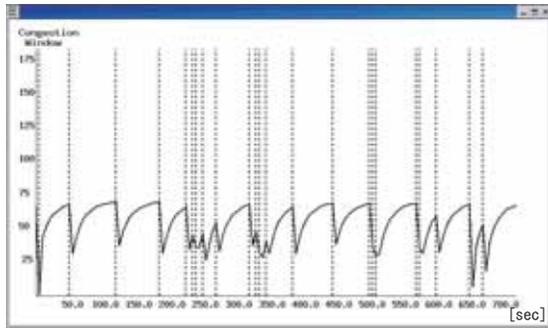


図 10 read/write ありの場合の輻輳ウィンドウ時間変化

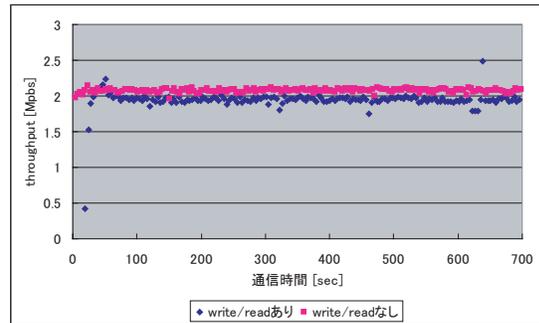


図 11 スループット時間変化

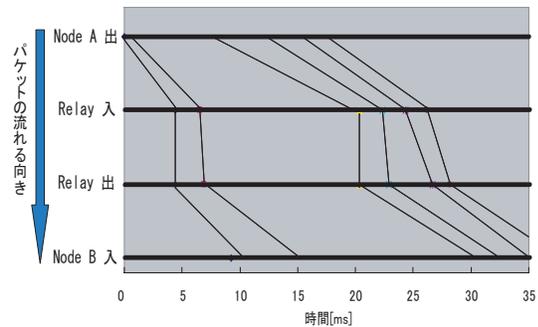


図 12 read/write なしの場合の packets traces

#### 4.1.3 測定結果に対する考察

以上見てきたように、中継装置にアプリケーション層レベルで read/write をさせた場合とさせずに IP 層でデータの中継のみを行う場合とを比較すると、read/write をさせた場合の方が中継ノードに負荷がかかり通信性能が下がる可能性が予想されたが、実際には通信性能には殆ど影響しないことが分かった。

図 12、図 13 から分かるように、上位層での処理は

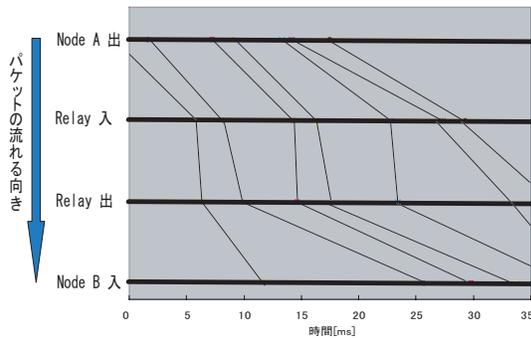


図 13 read/write ありの場合のパケットトレース

何もしないときよりも時間はかかっているが、IEEE 802.11b 無線 LAN の帯域幅が狭いためデータの転送速度が遅く、またパケットの送信ペースもゆっくりであるため、上位層の処理によってかかる時間は通信性能に大きな影響を与えずに、図 8、図 11 のように実際のスループットに差が殆ど出ないのであると考えられる。

#### 4.2 中継ノードにおける DB 処理

次節から DB を搭載した 2 台の端末の間に中継ノードを設けてマルチホップ通信を行い、2 台の端末間で DB 転送を行う際に、中継ノードに受け取ったデータを自身の DB に書き込みを行いキャッシュさせた場合と DB 処理を行わずに単に IP 層でデータを中継させた場合について通信性能を比較する。まず、7.3 では一般的なディスクを使用するタイプの DB (以降ディスク型 DB と呼ぶ) を用いて実験を行い、次に 7.4 ではよりモバイル DB に適すると考えられるインメモリアイプの DB を用いて実験を行う。

#### 4.3 中継ノードにおけるディスク型 DB 処理

##### 4.3.1 実験概要

ディスク型 DB を搭載した 2 台の端末の間に中継ノードを設けてマルチホップ通信を行い、2 台の端末間で DB 転送を行う際に、中継ノードに受け取ったデータを自身のディスク型 DB に書き込みを行いキャッシュさせた場合と DB 処理を行わずに単にデータを中継させた場合について通信性能を比較する。実験環境は 7.1 と同様図 5 のように 3 台の端末を用意し、それぞれに DB を搭載した。各端末のスペックは図 6 のようになっている。DB は PostgreSQL7.4.14<sup>7)</sup> を用いた。DB 転送におけるデータのやりとりとしては以下のような DB アクセスを伴ったソケット通信プログラムを用いた。

まず受信側が送信側に送信要求を出すと送信側が自身の DB から一行ずつ record を取り出し、それらを

まとめてソケット通信で送信するという動作を繰り返す。一方受信側はまとめてソケット通信で受けたデータを一行ずつ DB に insert していくという動作を繰り返す。このような環境において、中継ノードに送信側から来たデータを自身の IP 層で折り返して単にデータを転送させる場合と、受けたデータを自身のディスク型 DB に insert (データキャッシュ) をしてから受信側ノードに転送する場合との両方について実験を行い、ブロックサイズに対するスループット、輻輳ウィンドウの振舞、パケットトレースを元に解析を行う。

##### 4.3.2 性能測定

実験の結果を図 14~図 16 に示す。

図 14 は横軸がブロックサイズ [byte]、縦軸がスループットの平均値 [Mbps] を表している。ここでいうブロックサイズとは、ソケット通信において一度に write/read するデータサイズのこと、DB の 16byte のレコードを複数まとめて write/read した時の大きさを示している。また、スループットとは、受信側が受信したデータ量を、受信し始めてからそのデータ (record の集まり) を自身の DB への insert が完了するまでの時間で割って算出しており、通信性能だけでなく DB の読み書きも含んだ DB 転送処理性能を表している。ブロックサイズは 16byte から 2048byte まで 8 段階の値で測定した。

中継ノードの DB にデータをキャッシュさせた場合とさせなかった場合の両方ともブロックサイズに対してスループットが比例に近い形で増加しているのを見て取れる。これは、ブロックサイズが小さいときには帯域を効率的に利用出来ていないのに対し、ブロックサイズが 1024byte、2048byte と大きくなると、帯域を有効に活用できていることを表していると言える。また、ブロックサイズが小さい時には、中継ノードにおいて DB 処理をさせなかった方が大きなスループットを出しているが、ブロックサイズが大きくなるにつれ、その差は殆ど無くなる。

図 15 は中継ノードにおいて DB 処理をさせた場合の輻輳ウィンドウの時間変化のグラフである。ブロックサイズは 1024byte である。中継ノードに DB 処理をさせなかった場合も図 15 と同様な振舞を見せた。輻輳ウィンドウの値はある程度まで成長したのち、その値を使い切らずに一定となっている。これは、送信側ノードは DB からレコードを取り出しソケットに write するという中継ノードや受信側ノードに比べて軽い処理を行っており、中継ノードおよび受信側ノードでの実行時間に比べ短時間のうちに安定した状態で終了するためであると考えられる。

図 16 は中継ノードにおいて DB 処理をさせた場合のパケットトレースの図である。説明のため、パケットを 6 つのみ抽出し簡素化してある。4.1 で調べた図 12、図 13 では、中継ノードにおいて read/write 処理をさせたとしても、かかる時間は端末間のデータ転送に比べると極めて短かったが、中継ノードにおいてデータをディスク型 DB にキャッシュさせた場合はかなりの処理時間がかかっていることが分かる。

#### 4.3.3 測定結果に対する考察

以上見たように、中継ノードにおいてデータを自身のディスク型 DB に書き込みを行いキャッシュさせるという処理はかなりの負担がかかることが分かった。図 14 で見たように、中継ノードにおいて処理をさせた方がさせなかった場合よりもスループットは少し低くなった。しかし、ブロックサイズが大きくなるにつれ、その差が殆ど無くなるという現象が見られた。これは、ブロックサイズが大きくなるほど、レコード数が増えて DB の処理も増えることで、中継ノードと受信ノードにおける DB の insert 処理がオーバーラップする時間が増加し DB の転送処理全体の効率上がるためではないかと考えられる。

この実験では、中継ノードにおいて DB にデータをキャッシュさせることを試みた。実験を行いパケットトレースを見ると、中継ノードにおける処理にかなりの時間がかかっており、やはりソケット通信と比較してディスク型 DB の書き込み処理は重いことが分かる。このことから、中継ノードにおいて処理をさせた方がさせなかった場合よりも DB 転送処理性能が少しだけ低くなるが、中継ノードと受信ノードにおける DB 処理がオーバーラップしてくるにつれ、その差は殆ど無くなるという興味深い結果が得られた。

しかし図 16 の通り、中継ノードにおけるディスク型 DB 処理は重く、多数のノードからなるモバイル DB マルチホップ通信環境において更に高度な処理をさせるには不安が残る。そこで、よりモバイル DB に適していると考えられるインメモリタイプの DB を用いて同様の実験を行い、中継ノードの高機能化についての更なる考察を行う。

### 4.4 中継ノードにおけるインメモリ型 DB 処理

#### 4.4.1 実験概要

4.3.3 で述べたように本節ではインメモリ型 DB を搭載した 2 台の端末の間に中継ノードを設け、2 台の端末間で DB 転送を行う際に、中継ノードに受け取ったデータを自身のインメモリ型 DB に書き込みを行いキャッシュさせた場合と DB 処理を行わずに単に IP 層でデータを中継させた場合について通信性能を比較

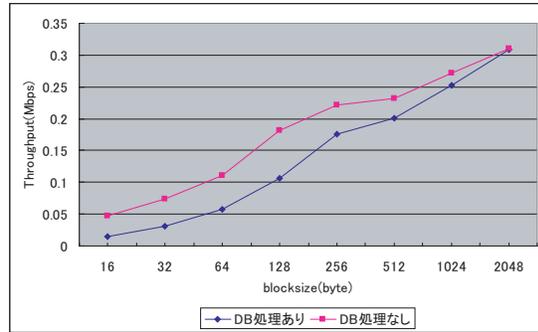


図 14 blocksize に対するスループット (DB 転送)

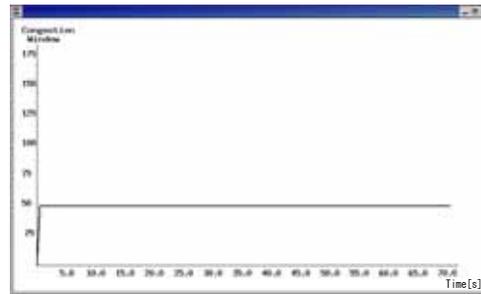


図 15 輻輳ウィンドウ時間変化

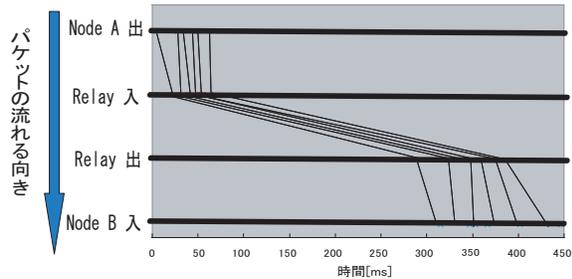


図 16 中継装置に DB 処理をさせた場合のパケットトレース  
する。

	CPU	メインメモリ	OS
NodeA, B Relay	Pentium 4 3.06GHz	512MB	Linux 2.6.9-1

図 17 マシンスペック 2

実験環境は図 5 のように 3 台の端末を用意し、それぞれにインメモリ型 DB を搭載した。各端末のスペックは図 17 のようになっている。インメモリ型 DB は Encirq DeviceSQL3.01<sup>8)</sup> を用いた。DB 転送におけるデータのやりとりとしては 4.3 節で用いたプログラ

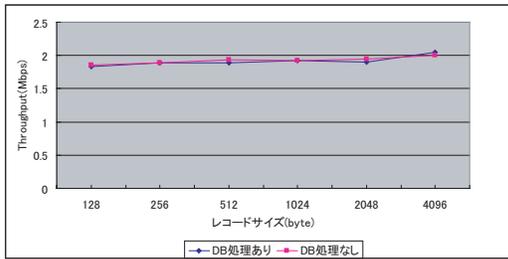


図 18 blocksize に対するスループット (DB 転送)

ムと同じような振る舞いの DB アクセスを伴ったソケット通信プログラムを用い、中継ノードに送信側から来たデータを自身の IP 層で折り返して単にデータを転送させる場合と、受けたデータを自身のインメモリ型 DB に insert (データキャッシュ) をしてから受信側ノードに転送する場合との両方について実験を行い、ブロックサイズに対するスループット、パケットトレースを元に解析を行う。

#### 4.4.2 性能測定

実験の結果を図 18 ~ 図 20 に示す。

図 18 は横軸がブロックサイズ [byte]、縦軸がスループットの平均値 [Mbps] を表している。ここでいうブロックサイズとは、ソケット通信において一度に write/read するデータサイズのこと、DB の 128byte のレコードを複数まとめて write/read した時の大きさを示している。また、スループットとは、受信側が受信したデータ量を、受信し始めてからそのデータ (record の集まり) を自身の DB への insert が完了するまでの時間で割って算出しており、通信性能だけでなく DB の読み書きも含んだ DB 転送処理性能を表している。ブロックサイズは 128byte から 4Kbyte まで 6 段階の値で測定した。中継ノードにおいて DB 処理をさせた場合とさせなかった場合とでスループットに差が生じないことが確認できる。

図 19 は中継ノードに DB 処理をさせなかった場合、図 20 は DB 処理をさせた場合のパケットトレースの図である。やはり、中継ノードにおいてインメモリ型 DB 処理をさせた方が中継ノードにおいて受け取ったデータを転送するのに時間がかかっているが、前節の図 16 と比べると、明らかに中継ノードにおける DB 処理が格段に軽くなっている。

#### 4.4.3 測定結果に対する考察

以上見たように、中継ノードにおけるインメモリ DB 処理は中継ノードを単にルータとして動作させた場合よりも若干時間がかかることが分かったが、ディスク型 DB 処理に比べると極めて処理が軽く、事実、図

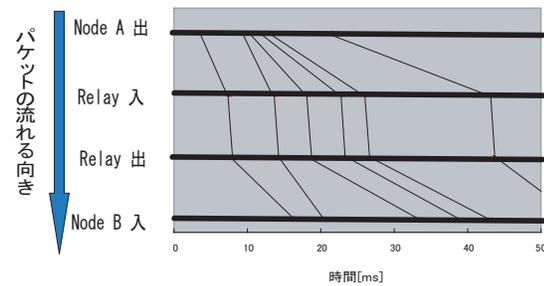


図 19 DB 処理をさせなかった場合のパケットトレース

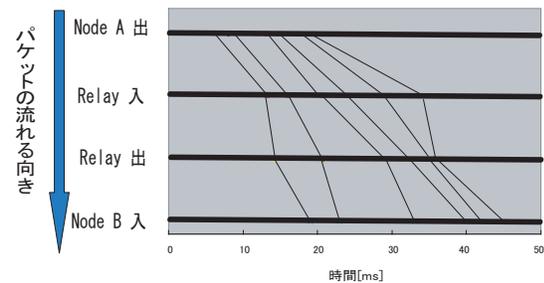


図 20 DB 処理をさせた場合のパケットトレース

18 のように、実際のスループットにはほとんど影響を及ぼさなかった。

## 5. まとめと今後の課題

本研究では無線 LAN ツーホップ通信環境を構築し、中継ノードにおいて高度な処理をさせた場合とさせなかった場合とでデータ転送性能の比較を行った。まず始めに、中継ノードにおいてアプリケーション層レベルで受信したデータを read/write してから転送させた場合と単にルータとして動作させた場合について比較を行い、次に中継ノードにおいて自身の DB にデータをキャッシュさせた場合とさせなかった場合との比較を行った。DB はディスク型 DB とインメモリ型 DB の両方について実験を行った。スループットの平均値、時間変化と輻輳ウィンドウの時間変化、中継ノードにおけるパケットの流れ等を元に議論を行った結果、本実験環境では中継ノードにおいて read/write 処理をさせる場合、させなかった場合と通信性能の差が殆ど生じず、DB にデータをキャッシュさせる場合も、させなかった場合と比べ、さほど大きく DB の転送処理性能が低下しないことが分かった。特に、インメモリ型 DB を用いた実験では良好な結果を得て、モバイル DB マルチホップ通信環境における中継ノードの高機能化の有効性を確認することが出来た。

今後の課題としては、まずツープではなくホップ数を増やしたマルチホップ環境を構築し実験を行う。ホップ数の増えた環境においては、3. で述べたように MANET 環境における通信の不安定さゆえに、中継ノードにおけるデータキャッシュやデータアグリゲートといった処理がより意義のあるものになると考えられる。

そして、インメモリDB型転送システムにおける具体的なアプリケーションを考案し、その環境に応じた実験を行い、各端末のアプリケーション層とトランスポート層、ネットワーク層をパケットが通過する時間についてタイムスタンプ記録コードを各層に挿入することで取り出し、パケットの流れを解析できるようなパケットトレースシステムを構築するなどして、より詳細な解析を行うことを目標とする。

### 参 考 文 献

- 1) Yingqi Xu, Wang-Chien Lee, Jianliang Xu, Gail Mitchell: "Processing Window Queries in Wireless Sensor Networks", The IEEE 22nd International Conference on Data Engineering (ICDE 2006), Atlanta, Georgia, USA, pp529-538, April 2006.
  - 2) Zhiyong Huang, Christian S. Jensen, Hua Lu, Beng Chin Ooi: "Skyline Queries Against Mobile Lightweight Devices in MANETs", The IEEE 22nd International Conference on Data Engineering (ICDE 2006), Atlanta, Georgia, USA, pp.403-413, April 2006.
  - 3) 松井愛子, 豊田真智子, 小口正人: "無線 LAN 通信における TCP パラメータの一解析", 第 68 回情報処理学会全国大会, IS-9, 2006 年 3 月
  - 4) 松井愛子, 神坂紀久子, 山口実靖, 小口正人: "無線 LAN 上のデータ転送時の TCP パラメータの一解析", 情報処理学会研究報告, 2006-DBS-140(II), pp. 443-449, 2006 年 7 月
  - 5) 松井愛子, 神坂紀久子, 山口実靖, 小口正人: "無線 LAN マルチホップ環境下における階層プロトコル上のパケットトレース", 電子情報通信学会 ITS 研究会, ITS2006-39, pp. 25-30, 2006 年 12 月
  - 6) tcpdump: <http://www.tcpdump.org/>
  - 7) PostgreSQL: <http://www.postgresql.org/>
  - 8) Encirq DeviceSQL: <http://www.encirq.co.jp/products/index.html>
  - 9) Ye Tian, Kai Xu, Nirwan Ansari: "TCP in Wireless Environments: problems and Solutions", IEEE Radio Communications, Vol.2, No.1, pp.s27-s32, March 2005
-