

無線ネットワークシミュレータ GloMoSim の並列化手法

東京電機大学 理工学部 情報システム工学科

加藤 剛史 小野 真和 桧垣 博章

E-mail: {kato, masa, hig}@higlab.net

無線ネットワークの高度化にともない、より複雑で高機能のプロトコル開発が必要とされてきており、その開発環境として無線ネットワークシミュレータの重要性が高まっている。より短時間でのシミュレーション実験を可能とするために、シミュレーションを複数の実行実体によって並列実行することが考えられる。本論文では、無線ネットワークシミュレータ GloMoSim において、各ノードに実行実体を割り当てる手法と、これを改造してすべてのノードのラジオ層の処理のみを独立の実行実体で行う手法を提案し、その性能評価を行う。

Parallelization of GloMoSim Wireless Network Simulator

Takeshi Kato, Masakazu Ono and Hiroaki Higaki

Department of Computers and Systems Engineering

Tokyo Denki University

E-mail: {kato, masa, hig}@higlab.net

GloMoSim is one of the widely available simulators for development and evaluation of protocols in mobile networks. For supporting simulation of highly developed protocols and for large scale mobile networks, high performance simulation environment is required. In order to solve this problem, a simulation program is parallelized and executed by multiple entities. This paper proposes a parallelization method for GloMoSim by which each mobile node is assigned an entity and radio layer in every mobile node is supported by a dedicated entity. By this method, utilization of processors is improved by parallelization and overhead for inter-entity communication is reduced.

1 背景と目的

ネットワーク環境のトラフィックの予測や通信プロトコルの研究用に様々なネットワークシミュレータが利用されている。通信プロトコルに組み込まれる機能の複雑化やシミュレーション対象のネットワークの大規模化等によりシミュレーションに要する時間は増加する傾向にある。そのため、並列化することによりシミュレーション時間の短縮を可能とするネットワークシミュレータが開発されている。

無線モバイルネットワークにおいては、ネットワークを構成するノードが時間とともに移動し、あるノードが送信した無線信号がどのノードに受信されるかは、そのときのノードの位置によって定まる。この性質を考慮すると、シミュレーションを並列化するためにシミュレーション領域の全体を地理的に複数の小領域に分割し、それぞれにシミュレーションの実行実体（プロセス、スレッド、CPU など）を割り当てる方法が考えられる。無線ネットワークシミュレータ GloMoSim [1] では、多ノードからなる広域無線ネットワークのシミュレーションを実現するために、シミュレーション領域を方形の複数の領域であるセルの集合に分割し、それぞれに別々の実行実体を割り当てるという方法で並列化を行っている（図 1）。しかし、無線モバイルネットワークにおいては、ネットワークを構成するノード群がシミュレーション領域全体に均一に分布することは稀であり、また通信要求の発生頻度の高い領域も偏在することが考えられる。そのため、一部のセルに割り当てられた実行実体に処理負荷が集中し、処理時間に大きな偏りを生じる結果、CPU 使用率の低下を招くことが考えられる。これを解決するためのひとつの方法として、より細粒度の並列実行環境とすることが考えられる。本論文では、GloMoSim の並列化をノードごとに実行実体を割り当てることで実現する手法および無線信号伝達の特性に基づいて、すべてのノードのラジオ層を分離してひとつの実行実体を割り当てる複合型の実現手法を提案し、性能評価結果を報告する。

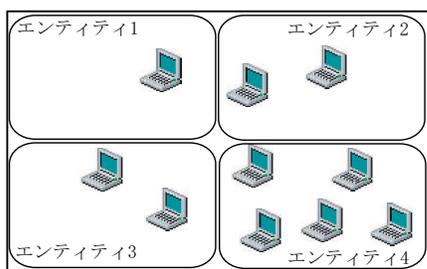


図 1: 地理的分割による並列実行

2 GloMoSim

GloMoSim は並列シミュレーション環境 PARSEC [2] 上に開発された無線ネットワークシミュレータである。各ノードは、ラジオ層、MAC 層、ネットワーク層、トランスポート層、アプリケーション層から構成され、各層が独立に実装される（図 2）。GloMoSim における実行実体の割り当て単位はエンティティと呼ばれ、ひとつ

のエンティティは複数のノードの動作をシミュレーションする（図 3）。前章で述べたように、GloMoSim では、シミュレーション領域を地理的に分割し、分割された小領域ごとにエンティティが割り当てられる。したがって、地理的分割の境界をまたいだ無線通信やノード移動時にエンティティ間の通信が必要となるが、小領域内の通信や移動はエンティティ内部で処理される。シミュレーションはイベントドリブン方式で進行し、ノード間の通信および同一ノード内の各層間の通信は、図 4 のようにすべてイベントを介して行われる。イベントの発行時にはイベント発生までの時間、イベントが発生するノードとその層が指定され、イベントスケジューラに登録される。イベントスケジューラの管理するシミュレーション時刻がイベント発生時刻と等しくなったとき、イベントスケジューラから登録時に指定されたノードの指定された層に対して、指定された種類のイベント発生要求が出される。ノード間の無線信号配送は、送信元ノードのラジオ層が受信イベントを発行し、この信号を受信可能なノードのラジオ層で無線信号到達時刻に受信イベントが発生する。

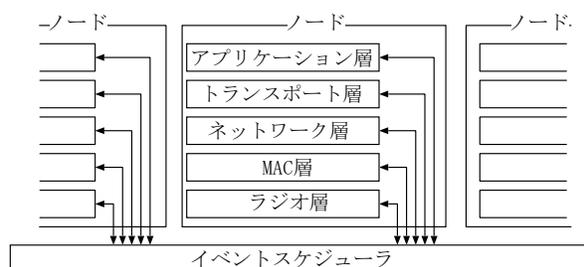


図 2: GloMoSim の構造

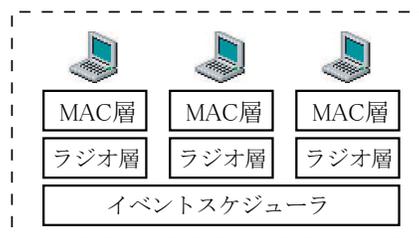


図 3: GloMoSim におけるエンティティ

3 提案並列化技法

前述したように、GloMoSim に実装されているセル単位での分散処理は、CPU 使用率を低下させる場合があるが、処理単位を細分化することによってプロセッサの空き時間を短くすることができる。そこで、図 5 に示すように各ノード単位での並列化、すなわちノード単位で実行実体を割り当てる方法を考える。ここでは、イベントスケジューラは複数のノードの複数の層の間のイベントの発行と発生をスケジュールするのではなく、単一ノード内の層間のイベントのやり取りをイベントを介して行うとともにエンティティ間通信によって他ノードのラジオ層と自ノードのラジオ層とのイベントのやり取りをスケジュールする（図 6）。つまり、ス

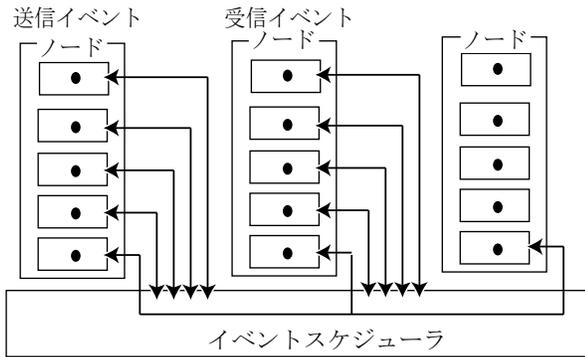


図 4: GloMoSim のメッセージ配送

ケジューラはノード内のイベント管理を行うこととする。このような実装を行った場合には、以下の問題が発生しうる。

- 実行実体間の通信が頻繁に発生するとその処理オーバーヘッドによってシミュレーションの性能が低下する。
- ノード座標情報を各ノードに実装されたイベントスケジューラ間で共有する機構が必要となる。

後者の問題は、あるノードが送信した無線信号がいつ、誰に、どの程度の強度で受信されるのかを決定するために必要となる。共有メモリ型のマルチプロセッサ環境であれば、座標情報を共有メモリに格納する方法は合理的である。しかし、グリッドなどの分散メモリ型マルチプロセッサ環境の場合は実装に工夫が必要であり、分散共有メモリやメッセージによる実現を検討しなければならない。また、細分化されたエンティティ間の通信が頻繁に発生するとオーバーヘッドが大きくなる可能性がある。

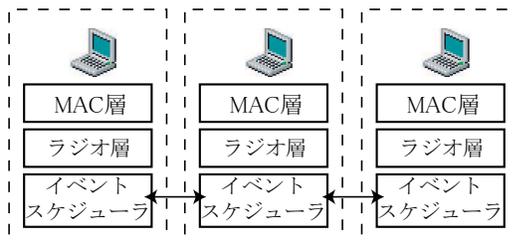


図 5: ノード単位のエンティティ割り当て

一方、前者の問題も各ノードに割り当てられた実行実体間の情報交換に関する問題である。この問題は、ノード間の通信の頻度に依存するが、無線モバイルネットワークではノード間の通信はブロードキャストベースで行われることから、図 7 に示すようにあるノードが無線信号を送信するとすべてのノードで無線信号を受信するイベントが発生する。これは、信号の受信の可否が各ノードの受信電力を元に決定されることと、到達した電波による雑音を考慮するために各ノードが受信電力の強さを知るための実装である。したがって、ラジオ層を含めて各ノードごとにエンティティを割り当てる実装を行う場合、送信元ノードに対応するエンティティから他のすべてのノードに対応するエンティティに

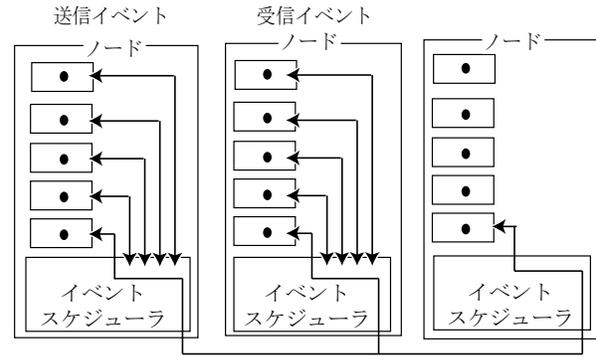


図 6: ノード単位並列化におけるメッセージ配送

対してエンティティ間通信が行われる。無線信号を受け取ったノードが受信電力の弱さや雑音の強さによって受信不能である場合はラジオ層で情報が破棄され、上位の MAC 層には通知されない。したがって、ノード間通信のメッセージ数よりラジオ層と MAC 層との通信のメッセージ数の方が少なくなる。そこで、実行実体間通信のオーバーヘッドが大きい場合は、すべてのノードのラジオ層だけを各ノードから切り離し、ラジオ層だけを処理するエンティティにまとめることによって図 8 のように実行実体間通信のメッセージ数を削減し、オーバーヘッドを抑えることができる。これは、MAC 層以上のイベントスケジューリングをノード単位に分割した実行実体に分散実装し、ラジオ層のみを従来の集中型あるいは地理的分割型で実現する複合実装方式であるといえる。

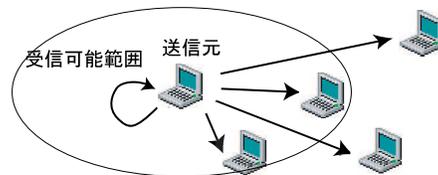


図 7: 無線信号のブロードキャスト

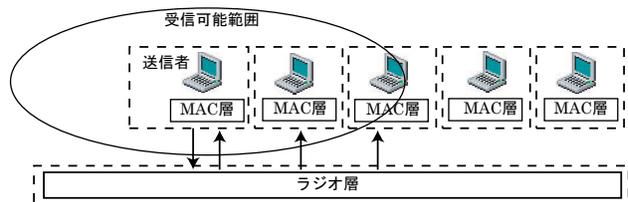


図 8: ラジオ層専用エンティティの導入

4 評価

GloMoSim の並列化にともなうオーバーヘッドを測定するため、図 5 のように各ノードごとにスレッドを対応付ける実装と、図 8 のようにラジオ層を 1 つのエンティティにまとめる実装を GloMoSim に行い、単一 CPU 環境でシミュレーションに要する時間を測定した。測定環境は、CPU は Opteron 850、メモリは 4GB、OS

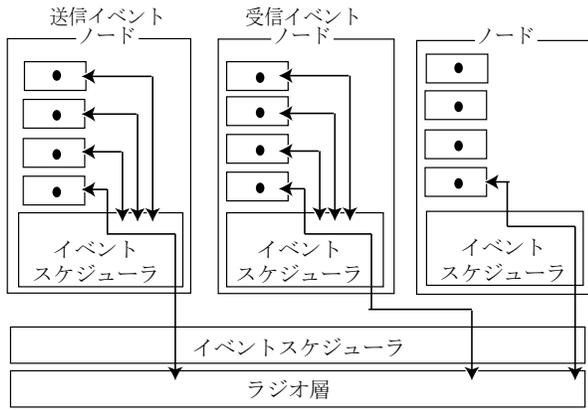


図 9: ラジオ層専用エンティティを用いたメッセージ配送

は Linux である。ここでは、GloMoSim のパッケージに付属する以下の 11 のシナリオを用いて測定を行う。

- wiredlink-small
- mixed-wired-wireless
- string
- routing-odmrp
- multi-mac-radio-test
- hidden
- ring
- exposed
- grid
- routing-fisheye
- mobihoc2001

測定結果を図 10、図 11、図 12 に示す。比較の結果、ノード単位の並列化のみを実装すると変更前に比べてオーバーヘッドが大きくなる傾向にある。オリジナルの GloMoSim による実行時間を基準 (100%) とすると最短で 98.7%、最長で 172.4%、平均で 118.7% である。一方、ノード単位並列化にラジオ層集約を組み合わせると並列化の効果とエンティティ間通信オーバーヘッドの削減によりオリジナルの GloMoSim よりもオーバーヘッドが削減される傾向となる。この場合のシミュレーション時間は、最短で 69.4%、最長で 115.2%、平均で 93.49% である。

5 まとめと今後の課題

GloMoSim の並列実行技法として、ノードを単位として分割し、実行実体を割り当てる分散型イベントスケジューラ技法と、無線通信の特性を考慮してラジオ層のみを集中型で実装する複合型イベントスケジューラ技法を提案し、実装と性能評価を行なった。今後はラジオ層を複数のエンティティで構成する手法、および異なる時刻にスケジュールされた複数のイベントを一貫性を失わずに並列処理する方法を提案していく。

参考文献

- [1] GloMoSim, <http://pcl.cs.ucla.edu/projects/gloMosim/>.
- [2] PARSEC, <http://pcl.cs.ucla.edu/projects/parsec/>.

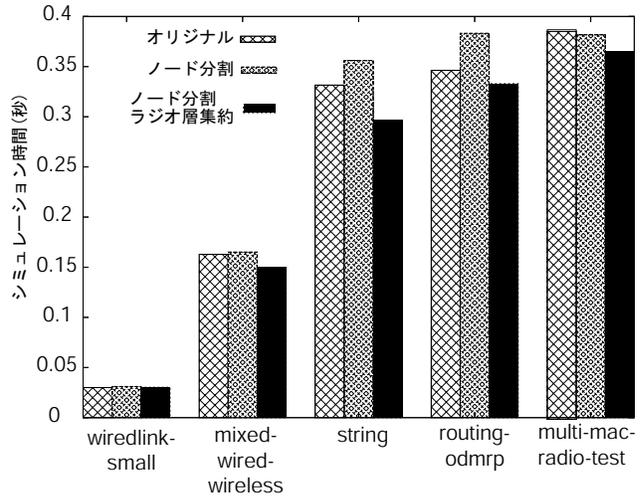


図 10: シミュレーション実験結果 (1)

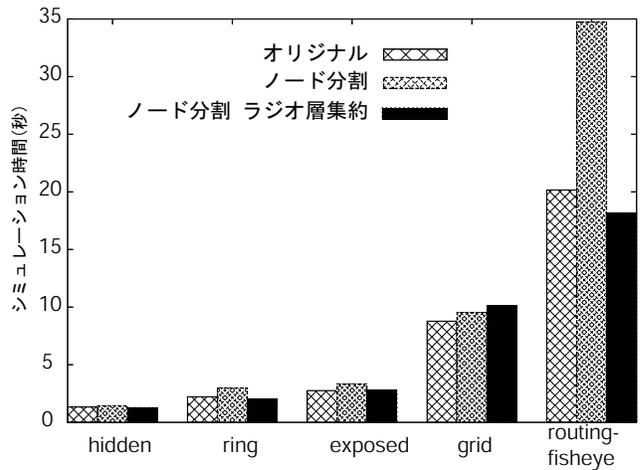


図 11: シミュレーション実験結果 (2)

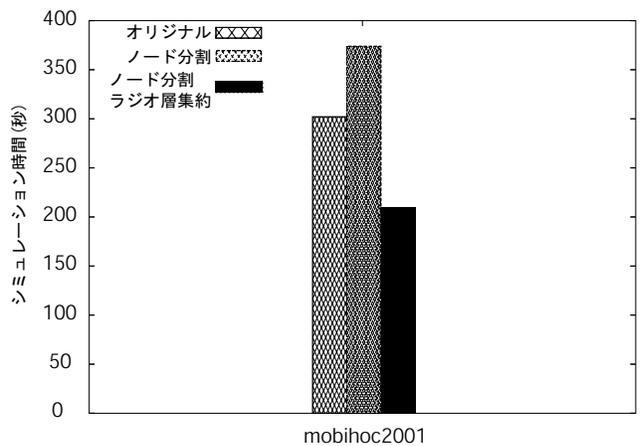


図 12: シミュレーション実験結果 (3)