

ネットワークシミュレータ GloMoSim の並列化*

東京電機大学 理工学部 情報システム工学科†
加藤 剛史 桧垣 博章‡ §

1 背景と目的

ネットワーク環境の負荷テスト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、それぞれに別々のスレッドを割り当てるといった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スレッド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いた上記2手法の複合型実現手法を提案し、一部の性能評価結果を報告する。

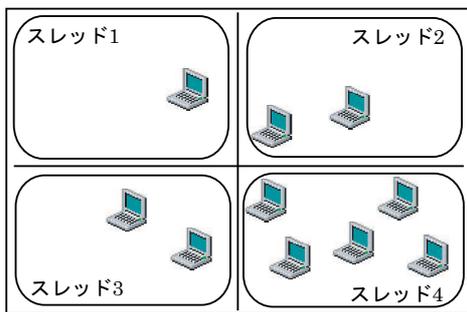


図 1: 地理的分割による並列実行

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。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。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、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。

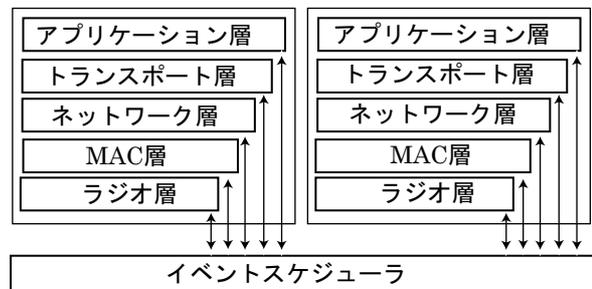


図 2: GloMoSim の構造

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3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と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。つまり、スケジューラは

*Distributed Event-Scheduler for Node-Based Parallelized GloMoSim

†Tokyo Denki University

‡Takeshi Kato and Hiroaki Higaki

§{kato, hig}@higlab.net

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（スレッド等）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。一方、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。一方で、細分化されたスレッドが頻繁にスレッド間通信を行うとオーバーヘッドが大きくなる可能性がある。

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 3 のように各ノードごとにスレッドを対応付けるよう GloMoSim を図 4 のようにし、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。測定環境のスペックは、CPU は Opteron 850、メモリは 2GB で、4 つのシミュレーションシナリオで測定を行った。同じシナリオで変更前の GloMoSim による実行時間の比較結果を表 1 に示す。表より並列化にともなうオーバーヘッドは 20 % 程度である。これは複数の CPU に分散することによってシミュレーション時間を短縮できる程度に小さいものであると考えられる。

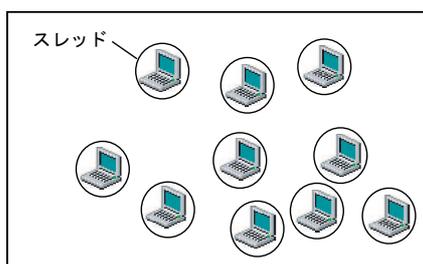


図 3: ノード単位の並列実行

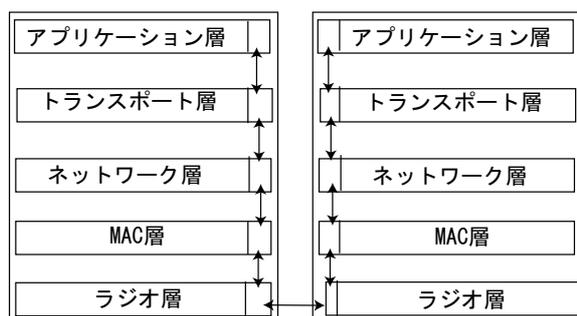


図 4: イベントスケジューラの分散実装

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問題であるが、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 5 のように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。

表 1: 実行時間の比較

シナリオ	変更前 [s]	変更後 [s]	比率
A	38.12	45.45	1.19
B	23.26	27.13	1.17
C	2.46	2.51	1.02
D	1.72	1.99	1.15

したがって、送信元ノードに対応するスレッドから他のすべてのノードに対応するスレッドに対してスレッド間通信が行われる。無線信号を受け取ったノードが受信電力の大きさや S/N 比によって受信不能と判断した信号はラジオ層で破棄され、上位層である 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、ノードに割り当てられた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実行実体にまとめることによって図 6 のようにプロセス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復号実装方式であるといえる。

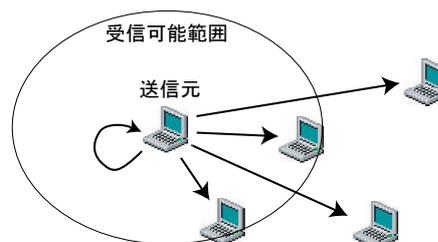


図 5: ブロードキャスト

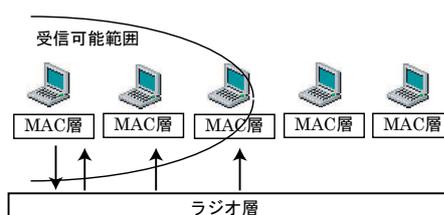


図 6: プロセス間通信の削減

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、その可能性を検証する実験を行なった。今後は、提案手法を実装し、並列化によるシミュレーション時間短縮の効果を検証する。

参考文献

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