社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

並列シミュレーションの高速化のためのリンク間トラヒック量にもとづくネットワークモデル分割法に関する検討

吉田　信平† 　大崎　博之† 　今瀬　真†

† 大阪大学 大学院情報科学研究科　〒 560-8531 大阪府豊中市待兼山町 1-3
E-mail: †{s-yosida,oosaki,imase}@ist.osaka-u.ac.jp

あらまし　近年、大規模ネットワークの性能評価手法に対する要求が高まっている。本稿では、大規模ネットワークのシミュレーションを分散計算機環境上で高速に実行するための、リンク間トラヒック量にもとづくネットワークモデル分割法を提案する。提案するネットワークモデル分割法の基本的なアイディアは、(1) シミュレーションを実行したいネットワークモデルに対して定常状態解析を実行し、定常状態においてリンク上を通過するトラヒック量を推測する、(2) 推測したトラヒック量にもとづき、最小カットアルゴリズムを利用して、定常状態において通過するトラヒック量が少ないリンクでネットワークモデルを分割する、(3) ノードを通過するトラヒック量の合計が均等になるように、(1)、(2) を繰り返し実行することによりシミュレーションモデルを N 個に分割する、というものである。
キーワード　大規模ネットワーク、ネットワークシミュレータ、並列シミュレーション、クラスタ計算機

On Network Model Division Method Based on Link-to-Link Traffic Intensity for Accelarating Parallel Simulatioin

Shinpei YOSHIDA†, Hiroyuki OHSAKI†, and Makoto IMASE†

† Graduate School of Information Science and Technology, Osaka University, Japan
E-mail: †{s-yosida,oosaki,imase}@ist.osaka-u.ac.jp

Abstract　In recent years, requirements for performance evaluation techniques of a large-scale network have been increasing. However, the conventional network performance evaluation techniques, such as mathematical modeling and simulation, are suitable for comparatively small-scale networks. Research on parallel simulation has been actively done in recent years, which might be a possible solution for simulating a large-scale network. However, since most existing network simulators are event-driven, parallelization of a network simulator is not easy task. In this paper, a novel network model division method based on link-to-link traffic intensity for accelerating parallel simulator of a large-scale network is proposed. The key ideas of our network model division method are as follows: (1) perform steady state analysis for the network model that is to be simulated, and estimate all traffic intensities along links in steady state, (2) repeatedly apply the minimum cut algorithm from graph theory based on the estimated traffic intensities, so that the simulation model is divided at the link that has little traffic intensities in steady state.
Key words　Large-Scale Network, Network Simulator, Parallel Simulation, Cluster Computer

1　はじめに

近年、インターネットの大規模化および複雑化に伴い、大規模ネットワークの性能評価手法に対する要求が高まっている [1, 2]。現在のインターネットは、ベストエフォート型のネットワークであり、エンド–エンド間の通信品質はまったく保証されない。もちろん、現状のインターネットでも、OSPF や BGP のような動的ルーティングを用いることにより、ある程度の堅牢性

は実現されている。しかし、インターネット自体が、社会の通信インフラストラクチャとして不可欠なものになっているため、ネットワークの信頼性・安全性・堅牢性の確保や、将来のネットワーク拡張および設計、テロや災害などの影響評価などを行うために、大規模ネットワークの性能評価手法が強く要求されている。

しかし、数学的解析手法やシミュレーション手法など、従来のネットワーク性能評価手法は、比較的小規模のネットワークを対象としている。例えば、従来のコンピュータネットワークの性能評価に広く用いられてきた待ち行列理論 [3] は、大規模かつ複雑なインターネットの性能評価に適用することは困難である。待ち行列理論を用いて、ネットワークの性能を厳密に解析する場合、ネットワークに接続されているノード数が増加するにつれ、解析における状態数が爆発的に増加する。

一方、待ち行列理論においても、Jackson ネットワークのように、多段接続されたネットワークを近似的に解析する手法が提案されている [3] が、ノードに到着するパケットがポアソン到着であることを仮定している。しかし、インターネットの通信プロトコルである TCP/IP は、階層化された複雑な通信プロトコルであり、複雑なトラヒック制御アルゴリズムやルーティングアルゴリズムが動作している。例えば、インターネットでは、Ethernet、FDDI、ATM など、さまざまな通信方式を用いており、このような複雑なシステムの厳密な数学的モデルを作成することは事実上不可能である。もちろん、数学的解析手法は計算時間という点では非常に有利であるため、他の性能評価手法を補完する方法として重要である。

一方、シミュレーション手法は、数学的解析手法よりも複雑なネットワークの性能評価が可能である [4]。特に、近年のコンピュータの高速化および大容量化により、中規模程度のネットワークの性能評価が可能となりつつある。しかし、インターネットの通信プロトコルは非常に複雑であることから、ネットワークのシミュレーションには大量の計算機資源が必要となり、今なお大規模ネットワークのシミュレーションは困難である。現在、広く用いられているネットワークシミュレータの大部分は、パケットレベルの挙動をシミュレートするため、イベント駆動型のアーキテクチャを採用している。単体の計算機上で動作する、ネットワークシミュレータの高速化手法も提案されている [5] が、大規模ネットワークのシミュレーションを行うためには、これらとは異なるアプローチが必要となる。

大規模ネットワークのシミュレーションを可能にする技術として、並列シミュレーションに関する研究が近年さかんに行われている [6-8]。デスクトップ型計算機の高速化および低価格化や、ギガビットイーサネット等の高速なネットワークインターフェースの普及により、比較的安価なクラスタ計算機の構築が容易となりつつある。また、広域ネットワークを用いて、世界中の計算機資源を統合する Grid コンピューティングも注目を浴びている。しかし、大部分のネットワークシミュレータは、イベント駆動型のアーキテクチャであるため、ネットワークシミュレータの並列化は容易ではない。

そこで本稿では、大規模のネットワークのシミュレーションを分散計算機環境上で高速に実行するための、リンク間トラヒック量にもとづくネットワークモデル分割法を提案し、その有効性を評価する。提案するネットワークモデル分割法の基本的なアイディアは以下の通りである。

(1) シミュレーションを実行したいネットワークモデル (シミュレーションモデル) に対して、文献 [9] で提案されている定常状態解析を実行し、定常状態においてリンク上を通過するトラヒック量を推測する。

(2) 推測したトラヒック量にもとづき、文献 [10] の最小カットアルゴリズムを利用して、トラヒック量の少ないリンクでシミュレーションモデルを分割する。

(3) ノードを通過するトラヒック量の合計が均等になるように、(1)、(2) を繰り返し実行することにより、シミュレーションモデルを N 個に分割する。

上記のような分割法により、シミュレーションモデルを N 個に分割し、それぞれの部分ネットワークモデルを N 個の計算機上で実行する。

本稿の構成は以下の通りである。まず、2 章においてネットワークの並列シミュレーションに関する関連研究について述べる。3 章において、提案するリンク間トラヒック量にもとづくネットワークモデル分割法を説明する。4 章では、提案するネットワークモデル分割法の実行例を示す。また、5 章では、簡単な実験により、提案するネットワークモデル分割法により並列シミュレーションがどの程度高速化されるかを評価する。最後に、6 章において、本稿のまとめと今後の課題について述べる。

2 関 連 研 究

代表的なネットワークシミュレータのうち、並列シミュレーションに対応しているものとして QualNet [11]、OPNET [12]、PDNS [13] などが存在する。QualNet は Scalable Network Technologies 社の商用シミュレータであり、単一の SMP (Symmetric Multi Processing) 計算機上での実行をサポートしている [11]。並列シミュレーションを高速化するために、シミュレーションモデルの分割 (Smart Partitioning)、CPU ごとの負荷分散 (Load Balancing)、シミュレーション先読みの最大化 (Maximization of Lookahead) などの手法を採用している。しかし、クラスタ計算機など、複数の計算機上での実行はサポートしておらず、大規模ネットワークのシミュレーションには使用できない。

一方、OPNET は、OPNET Technologies 社の商用シミュレータであり、単一の SMP 計算機上での実行をサポートするが、クラスタ計算機など、複数の計算機上での実行はサポートしてい

ない [12]。また、並列シミュレーションが可能であるのは、無線ネットワーク用の特定のモジュールだけであり、大規模ネットワークのシミュレーションには使用できない。

PDNS (Parallel/Distributed NS) は、米国ジョージア工科大の PADS 研究グループが開発したネットワークシミュレータである [13]。PDNS は、TCP/IP ネットワークの性能評価に広く用いられている、ns2 シミュレータ [14] を並列計算機上で動作するように拡張したものである。PDNS では、シミュレーションのノードを、異なる計算機上で分散して動作させることが可能である。ただし、並列シミュレータとしては非常に限定的な機能のみを実装している。単純に、大規模なネットワークのシミュレーションを、複数の計算機上で実行させた場合、シミュレーションを行う計算機間の通信によるオーバヘッドにより、シミュレーション速度が大幅に遅くなるという問題が指摘されている [13]。このため、PDNS をそのまま用いて、大規模ネットワークのシミュレーションを行うことも困難である。

3 リンク間トラヒック量にもとづくネットワークモデル分割法

提案するネットワークモデル分割法の概要を説明する。以下では、シミュレーションを行いたいネットワーク全体のモデルを「ネットワークモデル」、ネットワークモデルの分割によって得られるモデルを「部分ネットワークモデル」と呼ぶ。まず、シミュレーションを行うネットワークモデルを、重みつき無向グラフで表現する。グラフの頂点がノード (ルータもしくは端末) に相当し、グラフの辺がノード間のリンクに相当する。さらに、グラフの辺の重みとして、定常状態においてリンクを通過するトラヒック量を用いる。基本的なアイディアは、(1) シミュレーションを実行したいネットワークモデル (シミュレーションモデル) に対して、文献 [9] で提案されている定常状態解析を実行し、定常状態においてリンク上を通過するトラヒック量を推測する、(2) 推測したトラヒック量にもとづき、文献 [10] の最小カットアルゴリズムを利用して、トラヒック量の少ないリンクでシミュレーションモデルを分割する、(3) ノードを通過するトラヒック量の合計が均等になるように、(1)、(2) を繰り返し実行することによりシミュレーションモデルを N 個に分割する、というものである。

具体的には、まず、シミュレーションを行いたいネットワークモデルとノード間のトラヒック要求が与えられた場合に、文献 [9] で提案されている定常状態解析を用いて、定常状態において各リンクを通過するトラヒック量を導出する。さらに、文献 [10] で提案されている最小カットアルゴリズムを利用して、ネットワークモデルのカットの候補を複数求める。この中から、カットの容量 (部分ネットワークモデル間を通過するトラヒック量) が小さく、なおかつ 2 つの部分ネットワークモデルのシミュレーション計算時間 (部分ネットワークモデル内の全トラヒック量によって推測) ができるだけ均等となるカットを採用する。次に、各部分ネットワークモデルのうち、シミュレーション計算時間が最も大きいと考えられるネットワークモデルに対して、再度、上記のような分割アルゴリズムを適用する。上記のような分割を $N-1$ 回繰り返すことにより、部分ネットワークモデル間を通過するトラヒック量が少なく (つまり、並列シミュレーションのオーバヘッドが少ない)、かつ各部分ネットワークモデルのシミュレーション計算時間ができるだけ均等 (つまり、シミュレーションを実行する計算機の負荷が均等となる) となるような、N 個の部分ネットワークモデルが得られる。

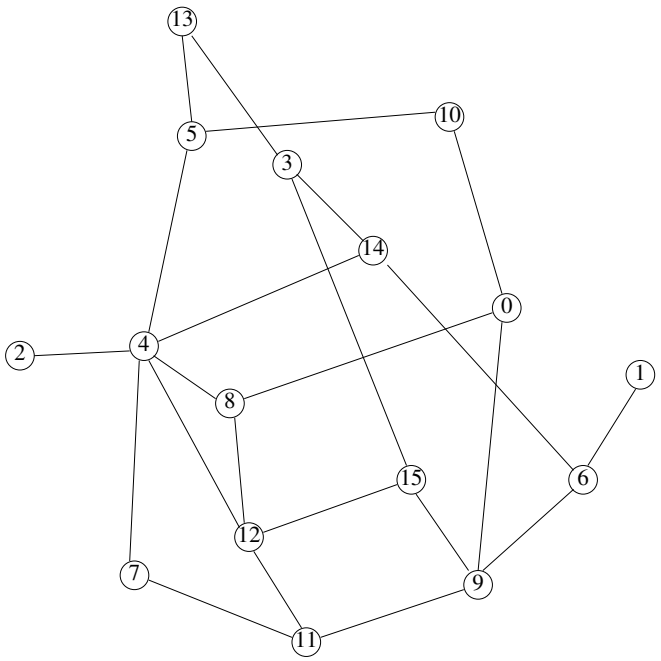

図 1: ネットワークモデル分割法の実行例 (アルゴリズム実行前)

次に、提案するネットワークモデル分割法のアルゴリズムを説明する。まず、アルゴリズム説明に先立ち、いくつかの記号を定義する。ネットワークモデルとして、無向グラフ $G=(V,E)$ を考える。ここで、$V=\{v_1,v_2,\ldots,v_n\}$、$E=\{e_1,e_2,\ldots,e_m\}$ である。辺 (v_i,v_j) の重みを $w_{i,j}$ とする。さらに、最終的なネットワークモデルの分割数 (部分ネットワークモデルの数) を N とする。さらに、シミュレーションで用いるトラヒックモデルを、トラヒック行列 $L=(l_{i,j})$ および $M=(m_{i,j})$ によって表記する。ここで、$l_{i,j}$ は頂点 v_i から頂点 v_j への UDP トラヒックの転送レート、$m_{i,j}$ は頂点 v_i から頂点 v_j への TCP コネクション数である。本稿では、簡単のため、連続的にデータ転送を行う TCP トラヒックおよび UDP トラヒックを対象とする。

提案するネットワークモデル分割法のアルゴリズムは以下の通りである。

(1) 定常状態解析によるリンク間トラヒック量の導出

ネットワークモデル G とトラヒック行列 L, M に対して定常状態解析を実行し、定常状態におけるリンク間のトラヒック

量を導出する。ネットワークの定常状態解析には、文献 [9] で提案されている解析手法を用いる。これにより、定常状態における TCP トラヒックのスループット $T_{i,j}$ および UDP トラヒックのスループット $L_{i,j}$ が求まる。ここで $T_{i,j}$ および $L_{i,j}$ は、定常状態において辺 (v_i, v_j) を通過する TCP トラヒックおよび UDP トラヒックのスループットである。

(2) 辺の重み $w_{i,j}$ の決定

辺 (v_i, v_j) の重み $w_{i,j}$ を以下のように定義する。

$$w_{i,j} = \sum_l \sum_m C_{l,m} T_{l,m} + \sum_l \sum_m D_{l,m} L_{l,m} \tag{1}$$

ここで $C_{i,j}$ は、$m_{i,j}$ が辺 (v_i, v_j) もしくは辺 (v_j, v_i) を通過すれば 1 となり、それ以外ならば 0 の値を取る。また、$D_{i,j}$ は、$l_{i,j}$ が辺 (v_i, v_j) もしくは辺 (v_j, v_i) を通過すれば 1 となり、それ以外ならば 0 の値を取る。これにより、重み $w_{i,j}$ は、定常状態において辺 (v_i, v_j) および辺 (v_j, v_i) を通過する全トラヒックのスループットの和を意味する。

(3) 部分グラフの集合 M の初期化

分割により得られる部分グラフの集合をネットワークモデル G により初期化する。

$$M \leftarrow \{G\} \tag{2}$$

(4) 最小カットアルゴリズムを利用したモデル分割

ネットワークモデルの分割数を N とする。以下の処理を $|M| = N$ になるまで繰り返し実行する。

(a) 部分グラフの集合 M から辺の重みの和 $W(E)$ が最大の部分グラフ $G' = (V', E')$ を選択する。重みつき無向グラフ $G = (V, E)$ における辺の重みの和 $W(E)$ は次式により定義される。

$$W(E) = \sum_{(v_i, v_j) \in E} w_{i,j} \tag{3}$$

(b) 重みつき無向グラフ G' に対して、文献 [10] で提案されている最小カットアルゴリズムを実行する。これにより、$|V'| - 1$ 個のカット $(S, \overline{S})$ が得られる。ここで、カットの容量が n 番目に小さいカットを $(S_n, \overline{S}_n)$ $(1 \leqq n \leqq |V'| - 1)$ と表記する。

(c) $(S_n, \overline{S}_n)$ の中から、カットの容量が小さく、なおかつ辺の重みの和ができるだけ均等な部分グラフを選択する。具体的には、

$$\frac{|W(S_n) - W(\overline{S}_n)|}{W(S_n) + W(\overline{S}_n)} \leqq \alpha \tag{4}$$

(α は定数) を満たし、なおかつ n が最小 (つまり、カットの容量が最小) となる部分グラフ S_n、$\overline{S}_n$ を選択する。そして、部分グラフの集合 M 中の G' を $\{S_n, \overline{S}_n\}$ で置き換える。

本稿で提案するネットワークモデル分割法は、以下のような特徴を持つ。まず、提案する分割法のアルゴリズムは、ヒューリスティックなアルゴリズムであり、グラフ理論の最小カットアルゴリズムを利用する。また、部分ネットワークモデルへの分割時に、全辺の重みの和 $W(E)$ を計算することによって、各部分ネットワークモデルのシミュレーションに必要な計算量を推定している。これにより、単にリンク間のトラヒック量 $T_{i,j}$、$L_{i,j}$ だけを用いて分割する場合と異なり、部分ネットワークモデルごとのシミュレーションに必要な計算量が均等になることが期待できる。一方、提案するネットワークモデル分割法は、定常性がある連続的な TCP トラヒックおよび UDP トラヒックを前提としており、バースト的なトラヒックを扱うことができない。また、重み $W(E)$ を用いて、部分ネットワークモデルのシミュレーションに必要な計算量を推定しているが、この推定方法の妥当性の検証が必要である。

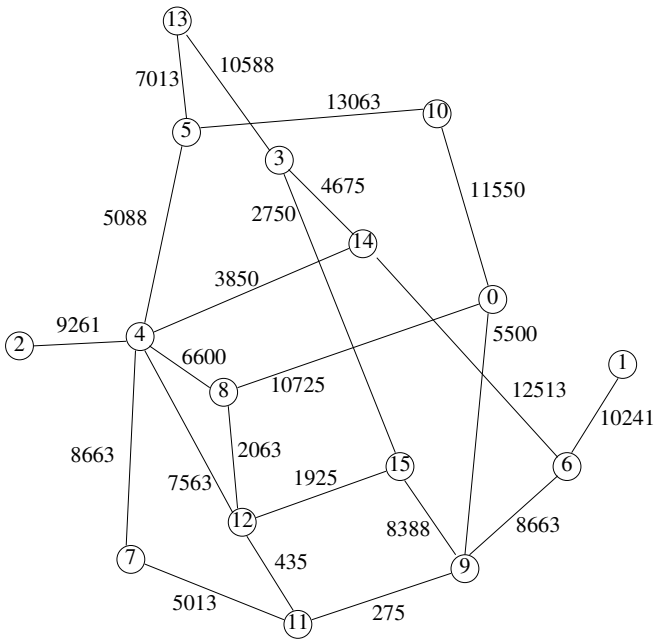

図 2: ネットワークモデル分割法の実行例 (定常状態解析により辺の重み $w_{i,j}$ を計算。各辺の値はリンクを通過するトラヒック量 [kbyte/s])

4 提案するネットワークモデル分割法の実行例

本章では、提案するネットワークモデル分割法の実行例を示す。ここでは、図 1 のような、ノード数 16、平均次数 3 のネットワークモデルを例として取り上げる。各リンクの帯域は 1 ~ 100 [Mbit/s] のランダムな値、各リンクの伝搬遅延は 10 ~ 200 [ms] のランダムな値である。また、4 台の並列計算機上でシミュレーションを実行することを考え、ネットワークモデルの分割数を $N = 4$ とする。ここでは、1000 本の TCP コネクションをランダムに生成した場合の結果を示す。

まず、図 1 に対して、文献 [9] の定常状態解析を実行し、定常状態において各リンクを通過する各トラヒックのスループット $T_{i,j}$ および $L_{i,j}$ を導出する。これをもとに、式 (1) より、各辺の重み $w_{i,j}$ を計算する (図 2)。

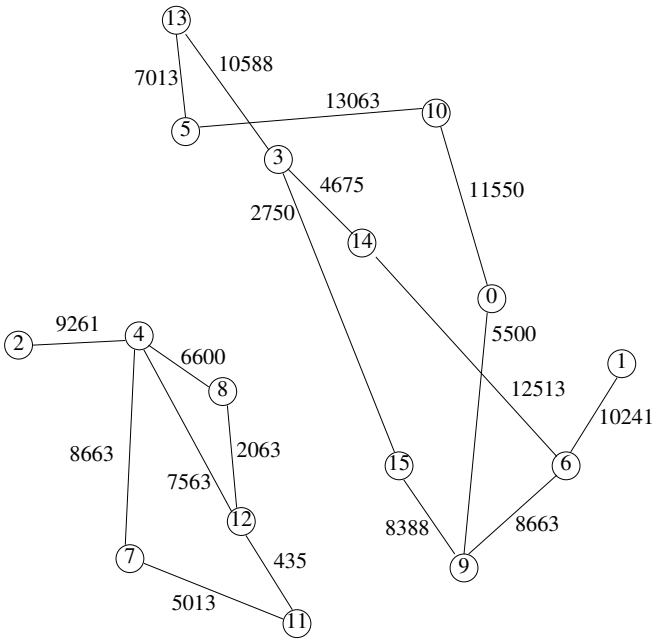

図 3: ネットワークモデル分割法の実行例 (最小カットアルゴリズムを利用し、2 つの部分ネットワークモデルに分割。$(S, \overline{S}) = (\{2, 4, 7, 8, 11, 12\}, \{0, 1, 3, 5, 6, 9, 10, 13, 14, 15\})$ のカットを適用)

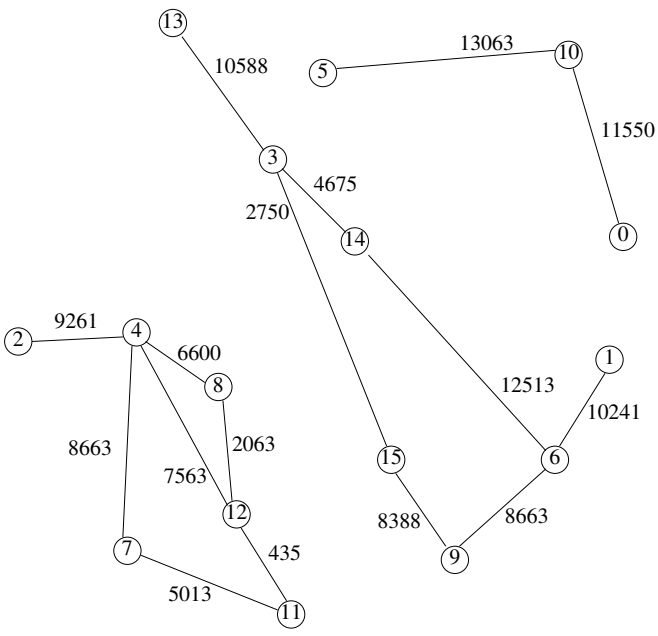

図 4: ネットワークモデル分割法の実行例 ($W(S) < W(\overline{S})$ であるため、$\overline{S} = \{0, 1, 3, 5, 6, 9, 10, 13, 14, 15\}$ を、さらに 2 つの部分ネットワークモデルに分割。$(T, \overline{T}) = (\{1, 3, 6, 9, 13, 14, 15\}, \{0, 5, 10\})$ のカットを適用)

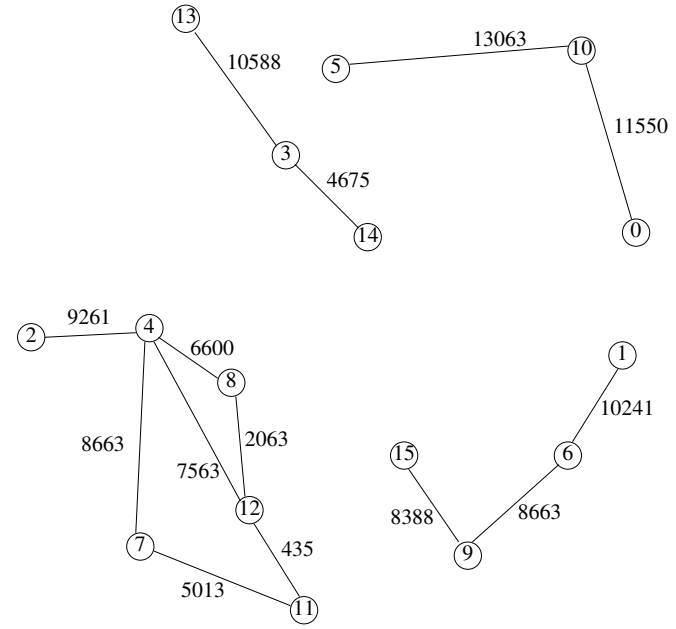

図 5: ネットワークモデル分割法の実行例 ($W(T) > W(\overline{T})$ であるため、$T = \{1, 3, 6, 9, 13, 14, 15\}$ を、さらに 2 つの部分ネットワークモデルに分割。$(U, \overline{U}) = (\{3, 13, 14\}, \{1, 6, 9, 15\})$ のカットを適用)

図 2 の重みつき無向グラフ $G' = (V', E')$ に対して、文献 [10] の最小カットアルゴリズムを実行し、$|V'| - 1 = 15$ 個のカット $(S, \overline{S})$ が得られる。これらのカットのうち、部分グラフ S および $\overline{S}$ における辺の重みの和 $W(S)$ および $W(\overline{S})$ が式 (4) を満たし、なおかつカットの容量が最小となるカットを適用する (図 3)。この例では、$(S, \overline{S}) = (\{2, 4, 7, 8, 11, 12\}, \{0, 1, 3, 5, 6, 9, 10, 13, 14, 15\})$、カットの容量が 21,863 [kbyte/s]、$W(S)$ =39,598 [kbyte/s]、$W(\overline{S})$ =94,944 [kbyte/s] となるカットが適用されている。

図 3 において、$W(S) < W(\overline{S})$ であるため、$\overline{S} = \{0, 1, 3, 5, 6, 9, 10, 13, 14, 15\}$ を、さらに 2 つの部分ネットワークモデルに分割する (図 4)。この例では、$(T, \overline{T}) = (\{1, 3, 6, 9, 13, 14, 15\}, \{0, 5, 10\})$、カットの容量が 12,513 [kbyte/s]、$W(T)$ =57,818 [kbyte/s]、$W(\overline{T})$ =24,613 [kbyte/s] となるカットが適用されている。

さらに、同様の手順を繰り返すことにより、N (= 4) 個の部分ネットワークモデルが得られる。ここでは、$W(T) > W(\overline{T})$ であるため、$T = \{1, 3, 6, 9, 13, 14, 15\}$ が、さらに 2 つの部分ネットワークモデルに分割される (図 5)。この例では、$(U, \overline{U}) = (\{3, 13, 14\}, \{1, 6, 9, 15\})$、カットの容量が 15,263 [kbyte/s]、$W(U)$ =15,263 [kbyte/s]、$W(\overline{U})$ =27,292 [kbyte/s] となるカットが適用されている。

5 提案するネットワークモデル分割法の評価

本章では、簡単な実験により、提案するネットワークモデル分割法を用いることにより、ネットワークの並列シミュレーショ

ンがどの程度高速化されるかを評価する。実験では、提案するネットワークモデル分割法を用いて 2 つの部分ネットワークモデルに分割した場合と、単純に部分ネットワークモデルのノード数が等しくなるように分割した場合について、並列シミュレーションの実行時間 (シミュレーションを開始してから、シミュレーションが終了するまでの時間) を計測した。

実験では、ノード数 16、平均次数 2 のランダムグラフによってネットワークモデルを生成した。ネットワークモデル中の、各リンクの帯域はすべて等しく 10 [Mbit/s] とし、各リンクの伝搬遅延は 10 ~ 200 [ms] のランダムな値とした。また、16 本の TCP コネクションをノード間にランダムに生成した。以上のような条件で、ネットワークモデルを 3 種類生成し、それぞれについて提案するネットワークモデル分割法を用いて 2 つの部分ネットワークモデルに分割した場合と、単純に部分ネットワークモデルのノード数が等しくなるように分割した場合について評価を行った。

並列ネットワークシミュレータとしては PDNS [13] バージョン 2.26-v2 を用い、6000 [s] のシミュレーションを実行した。パケット長、TCP のパラメータ、ルータのバッファサイズなどは、すべて PDNS バージョン 2.26-v2 のデフォルト値を用いた。実験では、以下のような同じ性能の計算機を 2 台使用した。

- CPU: Pentium III 1,266 MHz
- 搭載メモリ量: 1,024 Mbyte
- ハードディスク: 120 Gbyte
- ネットワーク: 1 Gbit/s Ethernet
- オペレーティングシステム: Linux バージョン 2.4.20

その結果、提案するネットワークモデル分割法を用いた場合、シミュレーションの実行時間は平均 236.4 秒、単純に部分ネットワークモデルのノード数が等しくなるように分割した場合、シミュレーションの実行時間は平均 280.7 秒であった。このことより、提案するネットワークモデル分割法を用いることにより、シミュレーションの実行時間が約 84 % に短縮されていることが分かる。

6 まとめと今後の課題

本稿では、大規模ネットワークのシミュレーションを分散計算機環境上で高速に実行するためのネットワークモデル分割法を提案した。提案したネットワークモデル分割法では、まず、シミュレーションを行いたいネットワークモデルに対して定常状態解析を適用することにより、リンク間トラヒック量を導出する。その後、定常状態におけるリンク間トラヒック量に応じて、グラフ理論の最小カットアルゴリズムを複数回適用し、ネットワークモデルを N 個に分割するという手法である。

本稿で提案した、並列シミュレーションの高速化のためのネットワークモデル分割法は、今後さまざまな拡張が考えられる。まず、本稿では、連続的にデータ転送を行う TCP トラヒックおよび UDP トラヒックを対象としていた。そこで、バースト的にデータ転送を行う TCP トラヒックを扱えるように、ネットワークモデル分割法を拡張することが考えられる。また、本稿ではユニキャストトラヒックのみを対象としていたが、現実の大規模ネットワークをシミュレーションするためには、マルチキャストトラヒックを扱えるように拡張する必要があると考えられる。

謝　　辞

提案するネットワークモデル分割法を設計するにあたり、有意義な議論をしていただいた、大阪大学サイバーメディアセンター教授の村田正幸氏に感謝いたします。

文　　献

[1] Large Scale Networking (LSN) Coordinating Group Of the Interagency Working Group (IWG) for Information Technology Research and Development (IT R&D), *Workshop on New Visions for Large-Scale Networks: Research and Applications*, Mar. 2001. available at `http://www.nitrd.gov/iwg/lsn/lsn-workshop-12mar01/index.html`.

[2] S. Floyd and V. Paxson, "Why we don't know how to simulate the Internet," Oct. 1999. available at `http://www.aciri.org/floyd/papers/wsc.ps`.

[3] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, New Jersey: Prentice-Hall, 1987.

[4] A. M. Law and M. G. McComas, "Simulation software for communications networks: the state of the art," *IEEE Communications Magazine*, vol. 32, pp. 44–50, Mar. 1994.

[5] V. S. Frost, W. W. Larue, and K. S. Shanmugan, "Efficient techniques for the simulation of computer communications networks," *IEEE Journal of Selected Areas in Comuunications*, vol. 6, pp. 146–157, Jan. 1988.

[6] H. T. Mouftah and R. P. Sturgeon, "Distributed discrete event simulation for communications networks," *IEEE Journal on Selected Areas in Communications*, vol. 8, pp. 1723–1734, Dec. 1990.

[7] G. F. Riley, R. M. Fujimoto, and M. H. Ammar, "A generic framework for parallelization of network simulations," in *Proceedings of the Seventh International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 128–135, Oct. 1999.

[8] S. Bhatt, R. Fujimoto, A. Ogielski, and K. Perumalla, "Paralell simulation techniques for large scale networks," *IEEE Communications Magazine*, vol. 38, pp. 42–47, Aug. 1998.

[9] D. Dutta, A. Goel, and J. Heidemann, "Faster network design with scenario pre-filtering," in *Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 237–246, Oct. 2002.

[10] M. Stoer and F. Wagner, "A simple min-cut algorithm," *Journal of the ACM*, vol. 44, pp. 585–591, July 1997.

[11] Scalable Network Technologies, Inc., "QualNet." `http://www.scalable-networks.com/`.

[12] Opnet Technologies, Inc., "OPNET." `http://www.opnet.com/`.

[13] PADS (Parallel and Distributed Simulation) Research Group, "PDNS - Parallel/Distributed NS." `http://www.cc.gatech.edu/computing/compass/pdns/`.

[14] "The network simulator – ns2." available at `http://www.isi.edu/nsnam/ns/`.