

# 可視化コンテンツ共有のための ネットワークスライス自動構築システム

遠藤 新<sup>1,a)</sup> 木戸 善之<sup>2</sup> 伊達 進<sup>2</sup> 下條 真司<sup>2</sup>

**概要:** 研究において生成される実験結果やシミュレーション結果を映像や画像に変換したものである可視化コンテンツを共有することは重要である。大容量かつ大規模な可視化コンテンツの共有には十分なネットワーク資源の確保が必要となる。可視化コンテンツの共有を行うユーザグループからネットワーク資源の確保を依頼されるネットワーク管理者は、独立したネットワーク資源を割り当てた論理ネットワークであるネットワークスライスを構築する。しかし、ユーザからの多様な要求を聞き入れる必要がある、かつ、複数のネットワークスライスが混在するため、可視化コンテンツ共有のためのネットワーク資源割り当てを最適化することは困難である。本研究では、可視化コンテンツ共有のためにネットワークスライスを自動で構築するシステムの実現の一環として、可視化コンテンツ共有の多様な要求に対応してネットワークスライスの構成を決定するスケジューリング手法を提案する。評価では、提案手法についてシミュレーションを行い、より多いユーザグループへのネットワークスライスの提供が可能であることを確認した。

**キーワード:** 可視化コンテンツ共有, 仮想ネットワーク自動生成, ネットワークスケジューリング

## Automatic Construction System of Network Slices for Sharing Visualization Contents

ARATA ENDO<sup>1,a)</sup> YOSHIYUKI KIDO<sup>2</sup> SUSUMU DATE<sup>2</sup> SHIJI SHIMOJO<sup>2</sup>

**Abstract:** In collaboration researches, it is important to share visualization contents, or, images or movies into which experimentation and simulation results of researches are transformed. An user group requires enough network resources to share large-scale and -capacity visualization contents so that it asks a network administrator to allocate the network resources to itself. The network administrator constructs a network slice, that is, a logical network with the network resources independent from other logical networks for the allocation. However, it is difficult for the network administrator to optimize the network resource allocation for sharing visualization contents because meeting diverse requirements of user groups and constructing multiple network slices are needed. In this research, we propose the scheduling method to meet the diverse requirements as part of the realizing the automatic construction system of network slices for sharing visualization contents. To evaluate the proposed method, we conducted the simulation and verified that more network slices can be provided to user groups with the proposed method.

**Keywords:** Visualization Content Sharing, Virtual Network Automatic Generation, Network Scheduling

### 1. はじめに

共同研究における研究者同士の意見交換をサポートする共同研究基盤の重要性が高まっている。共同研究では、研究者はそれぞれ地理的に離れた拠点に属しており、研究者

<sup>1</sup> 大阪大学大学院情報科学研究科  
Graduate School of Information Science and Technology,  
Osaka University, Suita, Osaka 565-0871, Japan

<sup>2</sup> 大阪大学サイバーメディアセンター  
Cybermedia Center, Osaka University

<sup>a)</sup> endou.arata@ais.cmc.osaka-u.ac.jp

は自身の拠点とは別の拠点に所属する研究者と共同作業することが共同研究活動の上で必要となる。遠隔地間での研究活動を支援する基盤、つまり共同研究基盤は、共同研究の件数は年々増加しているため、今後重要となることが予測される。

共同研究基盤の1つの要素として、高解像度な可視化コンテンツの表示が可能である大規模可視化装置がある。研究の議論において、実験やシミュレーションから得られる知見の共有は重要である。知見の共有には直感的な理解や分析が必要であり、実験結果である数値データを画像や映像に変換した可視化コンテンツは、直感的な理解や分析を容易にする。大規模可視化装置は、4K や 8K などの高解像度を持つ可視化コンテンツを高精細に表示可能な装置である。代表例としては、Tiled Display (TD) [7] がある。

しかし、各拠点の大規模可視化装置間で高解像度な可視化コンテンツを共有し、同時視聴や協調的な操作を行うことは困難である。可視化コンテンツ共有を広域ネットワーク上で実施するユーザグループが高精細、高品質な可視化コンテンツの視聴を要求する場合、遅延やパケットロスによる可視化コンテンツの品質劣化を回避するのに十分なネットワーク資源を確保することが必要となる。ネットワーク資源の確保には、論理ネットワークの利用とネットワークの占有の2つの方法があげられる。論理ネットワークとは、物理ネットワーク上に作成した L2 仮想オーバーレイネットワークを指す。論理ネットワークの利用には、ネットワーク管理者を頼ることが必要となる。ネットワーク管理者は、ネットワークのトポロジ、トラフィック、必要となる帯域幅、もしくはホップ数などを考慮して、手でユーザグループの要求する可視化コンテンツ共有のための論理ネットワークを設定する。一方で、ネットワークは他のユーザグループにとっても共通の基盤であるため、占有しての利用は難しい。しかし、論理ネットワークの作成においてユーザグループ間でのネットワーク資源の割り当てを最適化しなければならず、ネットワーク最適化問題は、最適解を導き出すことが難しい NP 困難な問題である。さらに、広域ネットワーク上の論理ネットワークの経路上にある全てのスイッチなどのネットワーク機器に対して、ユーザグループごとに設定を施すことはネットワーク管理者の負担となる。

論理ネットワークに対してネットワーク資源を割り当てるスケジューリング手法としては、Gu 氏らの提案する Virtual Overlay Network Scheduling がある [4]。しかし、この手法は可視化コンテンツ共有に応じて発生するユーザグループの多様な要求に応じてスケジューリングすることができない。また、この手法はヒューリスティック法を用いているため、最適なネットワーク資源の割り当てができない場合もある。論理ネットワークの設定の自動化については、Sherwood 氏らが提案する Flowvisor がある [8]。

この手法は、単一端末における操作のみで、独立した帯域幅やトポロジを割り当てた論理ネットワークであるネットワークスライスを OpenFlow ネットワーク上に生成する。このように、スケジューリング手法の問題によってネットワークスライス自動構築システムの実現が困難となっている。

本研究では、ネットワーク管理者の負担の解消のために、可視化コンテンツ共有を可能とするネットワークスライスを複数のユーザグループに対して提供するシステムの実現を目指す。そのために、可視化コンテンツ共有に応じて発生するユーザグループの多様な要求に対応し、最適なネットワーク資源の割り当てを可能とするスケジューリング手法を提案する。提案手法を用い、ネットワークスライス自動構築システムのプロトタイプを実現することを本研究の目的とする。

以下、2章では既存のスケジューリング手法における問題を説明する。3章では、可視化コンテンツ共有に対応するネットワークスライスのスケジューリング手法を提案する。4章では、提案手法を用いたネットワークスライス自動構築するプロトタイプシステムについて説明する。5章では、提案手法を評価するため、プロトタイプシステムを用いたシミュレーション実験を行う。6章では、本研究のまとめと将来課題について述べる。

## 2. 既存のスケジューリング手法の問題

ユーザグループのリクエストに基づいてネットワークスライスに割り当てる帯域幅とトポロジを決定するスケジューリング手法の研究としては、Virtual Overlay Network Scheduling [1], [2], [4] がある。これらの研究では、ランダムに発生する複数のユーザグループからのリクエストに対して、複数拠点を接続する仮想オーバーレイネットワーク（ネットワークスライス）に割り当てる帯域幅とトポロジをヒューリスティック法で逐次的に算出していく手法を提案している。具体的なアルゴリズムの例として、Algorithm 1 を示す。このアルゴリズムを採用するスケジューラは、管理する物理ネットワークのネットワークグラフとリンク帯域幅を把握しているものとする。各定数と変数は、表 1 で示す。まず、ユーザグループはリクエストをスケジューラに送信する。リクエストを受信したスケジューラは、物理ネットワークグラフの複製を生成し、そのリクエストが帯域幅の不足により利用できない物理リンクを複製したネットワークグラフから削除する。この複製したネットワークグラフ上で、3つのヒューリスティック法による経路探索手法のうちいづれかをを用いて各仮想リンクに割り当てる経路を算出する。どの手法を用いるかはスケジューラの実装時に決定される。全ての仮想リンクについて経路を算出できた場合は、リクエストを受理し、ネットワークスライスに割り当てる帯域幅とトポロジの情

**Algorithm 1** Virtual Overlay Network Scheduling [4]

```

1: Given incoming request  $r^n = (S^n, L^n, T^n, b^n)$ , generate temporary graph copy  $G'(V, E) = G(V, E)$ 
2: Remove non-feasible links in  $G'(V, E), i.e., c_e(t) < b^n$  in  $[t_s^n, t_e^n]$ 
3: for  $i = 1$  to  $|L^n|$  do
4:   if minimum hop count then
5:     Assign unity weight to links in  $G'(V, E)$ 
6:     Run Dijkstra's shortest-path for  $i$ -th virtual links
7:   else if minimum distance then
8:     Assign dynamic weight to links in  $G'(V, E)$ 
9:     Run Dijkstra's shortest-path for  $i$ -th virtual links
10:  else
11:    Run k-SP between end-points of  $i$ -th virtual link
12:    Select path with maximum bottleneck bandwidth
13:  end if
14: end for
15: if all virtual link connections in  $L^n$  routed then
16:   Setup successful, copy  $G'(V, E) \rightarrow G(V, E)$ 
17: end if
    
```

表 1 各定数と変数の定義

定数・変数	定義
$V$	ルータやスイッチなどの物理ノード集合
$E$	物理リンク集合
$G(V, E)$	物理ネットワークを表すネットワークグラフ
$n$	リクエストを送信したユーザグループ
$r^n$	$r^n = (S^n, L^n, t_s^n, t_e^n, b^n)$ で表されるユーザグループ $n$ のリクエスト
$S^n$	リクエスト $r^n$ が指定するノード集合
$L^n$	リクエスト $r^n$ が指定する仮想リンク集合
$t_s^n$	リクエスト $r^n$ が指定する可視化コンテンツ共有を開始する時間
$t_e^n$	リクエスト $r^n$ が指定する可視化コンテンツ共有を終了する時間
$b^n$	リクエスト $r^n$ が指定する帯域幅
$c_e(t)$	物理リンク $e$ のリンク帯域幅のうち、時刻 $t$ にネットワークスライス生成を予約したリクエストが利用していない帯域幅

報であるネットワークスライス構成を出力する。

上述した既存のスケジューリング手法では、ネットワークスライスに帯域幅とトポロジを効率的に割り当てる上で可視化コンテンツ共有におけるユーザグループの多様な要求に対応することは困難である。なぜなら既存のスケジューリング手法では、前提として全ての仮想リンクが同じ帯域幅を持ち、仮想リンク間で物理リンクを共有しない、つまり占有利用するネットワークスライスのみを生成することになっている。一方で、可視化コンテンツ共有では、共有する可視化コンテンツや共有を行うグループ、共有に用いる手段などのユーザグループの要求が変化すると、ネットワークスライスに必要な帯域幅やトポロジが変わる。例えば、解像度が異なる複数の可視化コンテンツの共有に提供するネットワークスライスでは、各仮想リン

クを利用する可視化コンテンツの解像度に合わせてその仮想リンクに要求される帯域幅が異なる。また、可視化コンテンツがマルチキャストとユニキャストのどちらの手法で配信されるかによって、ネットワークスライスの各仮想リンクに要求される帯域幅が異なる。このように多様なユーザグループの要求で必要とされるネットワークスライスに対して、既存のスケジューリング手法では、全ての仮想リンクが同じ帯域幅を持ち、仮想リンク間で物理リンクを共有しないネットワークスライスのみを生成する。この特徴により、各仮想リンクへの異なる帯域幅の割り当てや、マルチキャストな帯域幅の利用ができず、冗長な帯域幅の割り当てが生じてしまう。

また、既存のスケジューリング手法では、ヒューリスティック法を用いるために、帯域幅とトポロジの最適な割り当てができない場合がある。Algorithm 1 で示す手法では、リクエストを到着順に処理し、受理済みのリクエストに対する帯域幅とトポロジの再割り当てはしないため、十分な帯域があっても割り当てができない場合がある。これに対して、Bai 氏らの提案する手法では再割り当てをしない点を改良している [1], [2]。これらの手法では、Algorithm 1 においてある仮想リンクに対する物理経路の割り当てを失敗した時に、その仮想リンクに対して 1 つの候補経路を用意する。そして、用意した候補経路が割り当て可能となるように、受理済みのリクエストに割り当てられた帯域幅とトポロジを再割り当てする。しかし、用意される候補経路は 1 つだけであり、この経路が最適であるとは限らない。このことによって、スケジューリングが失敗するリクエストが増える場合がある。

帯域幅とトポロジの最適な割り当てを達成する手法として整数計画法があるが、スケジューリング手法に整数計画法を採用する場合、ユーザグループに対してネットワークスライスの生成が可能かどうかを判断するまでの時間が長くなるのが課題となる。一般的に、ユーザグループのスケジュールには可視化コンテンツ共有の実施だけでなく、他の多くの予定が決められている。ユーザグループが自身のスケジュールを管理する上で、可視化コンテンツ共有の実施可能な日をできるだけ早く知ることが可能であることが望ましい。そのためには、リクエスト受理の成否をできるだけ早くユーザグループに伝えることが必要となる。しかし、整数計画問題は NP 困難な問題であり、その計算時間は長く、リクエスト受理の成否を受け取るまでの時間が長くなる。上記であげた既存研究 [1], [2], [4] においても、整数計画問題が NP 困難な問題であることを理由に、整数計画法によるスケジューリングは困難であると述べられている。

### 3. 提案手法

本研究では、可視化コンテンツ共有における多様なユー

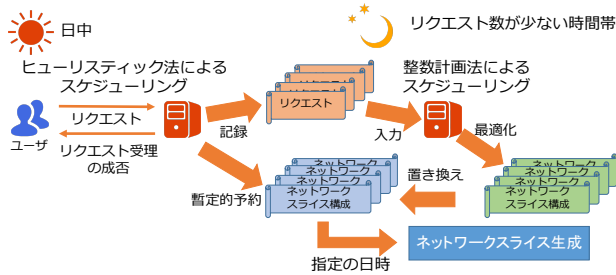


図 1 提案手法の概要

ザグループの要求に対応し、かつ整数計画法による最適な帯域幅とトポロジの割り当てを行うため、ユーザグループから受け付けたリクエストに対して 2 段階に分けたスケジューリングを実行する手法を提案する。提案手法の概要を図 1 で示す。各段階のスケジューリングにおいては、可視化コンテンツ共有における多様なユーザグループの要求に対応への一歩として、様々な解像度を持つ可視化コンテンツ、ストリーミングによる可視化コンテンツの配信、マルチキャストな帯域幅の利用を想定する。

リクエストの受け付け時である 1 段階目では、ユーザグループに対するリクエスト受理の成否を短い時間で返すために、ヒューリスティック法によるスケジューリングを行う。提案手法を実装するシステムは、複数のユーザグループからのリクエストを受け付ける。スケジューリングで必要とするリクエストは、2 章で述べた既存研究におけるリクエストと同様に  $r^n = (S^n, L^n, t_s^n, t_e^n, b^n)$  とする。システムは、複数のユーザグループからのリクエストに対して受け付けたリクエストから順にヒューリスティック法を用いたスケジューリングを実行する。スケジューリングに成功した場合、すなわち、リクエストの要求に応じてネットワークスライスに割り当てる帯域幅とトポロジを確保できた場合は、そのリクエストを受理する。受理したリクエストについて、リクエストの内容は記録して 2 段階目における整数計画法で利用し、ネットワークスライス構成はネットワークスライス生成の暫定的な予約に利用する。失敗した場合は、リクエストで指定した日時とは他の、ネットワークスライス生成が可能な日時を、ユーザグループに対して複数提示するリコmendを実行する。リコmendにおいては、 $T^n$  のみを別の日時に変更したリクエストの複製に対してヒューリスティック法によるスケジューリングを行う。リコmendで探索する別の日時については、本研究では簡易化のため、ユーザグループが指定した日付の前後 2 日間とした。

ヒューリスティック法によるスケジューリングは、2 章で述べた既存研究を基礎とし、マルチキャストな帯域幅の利用に対応する。本研究ではマルチキャストな帯域幅の利用を想定するスケジューリングの実現を目的の 1 つとしているため、ヒューリスティック法の基本的な部分は、実装の

容易な Algorithm 1 の minimum hop count の手法に沿ったアルゴリズムとする。ただし、全ての仮想リンクへの物理リンクの割り当てが成功した時、その物理リンク集合から重複する物理リンクを 1 つを残して取り除いたものを、ネットワークスライスに割り当てる。これにより、仮想リンク間で重複する物理リンクの本数分の帯域幅が 1 つのネットワークスライスに割り当てられていた既存手法を修正し、1 本分の帯域幅を割り当てるに止めることができる。

リクエストの少ない時間帯に行われる 2 段階目では、受理した全てのリクエストに対する帯域幅とトポロジの割り当てを最適化する目的で、整数計画法によるスケジューリングを行う。リクエストを出すユーザグループは勤務時間内に可視化コンテンツ共有を実施する日付を検討することが想定されるため、夜間などの勤務時間外にユーザグループがリクエストを出すことは少ない。こうした時間帯に、受理した全てのリクエストに対する整数計画法によるスケジューリングを長時間かけて実行する。算出された最適解で得られる帯域幅とトポロジの割り当てが、暫定的予約より帯域幅の利用率が低い場合、暫定的予約とその最適解から得た帯域幅とトポロジの割り当てによるネットワークスライス生成の予約を置き換える。時間内に最適解が得られなかった場合は、最適解の代わりに実行可能解を適用して、予約を置き換える。帯域幅の利用率が高い場合、もしくは、実行可能解もない場合は、暫定的な予約によるネットワークスライスの生成をそのまま指定の日時に行う。この段階は、リクエストの少ない時間帯が訪れるたびに行い、より低い帯域幅の利用率で予約を更新していく。

提案手法で用いられる整数計画法では、目的関数を式 1、制約式を式 2-9 とする整数計画問題を解く。式中の各変数の定義は、表 2 で示す。この整数計画問題は、2 章で述べた既存研究で言及されている整数計画問題を修正し、マルチキャストな帯域幅の利用に対応した問題である。目的関数の最小化は、各リクエストが利用する物理リンクの本数を、リクエストの指定する帯域幅が大きいものを優先して最小化することを意味している。各制約式について説明する。まず、式 2, 3 は  $u^{n,e,k} = \min \sum v_i^n \sum v_j^n p_{i,j}^{n,e,k}$  を表現する big-M 法 [10] を活用した制約式である。式 4-8 は各仮想リンクに物理リンクを割り当てるための制約式である。仮想リンクの両端である 2 つのノード間をつなぐ経路上にある各ノードの特徴として、経路の両端に位置するノードのリンク次数は 1、それ以外のノードのリンク次数は 2 となる。これらの式は、この特徴を定式化しており、仮想リンクの両端のノードをつなぐ経路の存在を保証する。式 9 は、ネットワークスライスを利用する間、仮想リンクには同じ物理リンクが割り当てられることを保証する制約式である。式 10 は、全てのリクエストに割り当てられた物理リンクの帯域幅の合計がリンク帯域幅を超えないことを保証するための制約式である。

表 2 各変数の定義

変数	説明
$R$	$r^n = (S^n, L^n, t_s^n, t_e^n, b^n)$ で表されるリクエスト集合
$E$	物理リンク集合
$C$	リンク帯域幅
$M$	big-M 法における十分に大きい定数
$T^n$	$t_s^n \leq T^n \leq t_e^n$ で表されるネットワークスライスを利用する時間
$p_{i,j}^{n,e,k}$	リクエスト $r^n$ の仮想リンク $(v_i^n, v_j^n)$ が時間 $k$ に物理リンク $e$ を利用するかどうかを示すバイナリフラグ (利用する場合 $p_{i,j}^{n,e,k} = 1$ )
$u^{n,e,k}$	リクエスト $r^n$ が時間 $k$ に物理リンク $e$ を利用するかどうかを示すバイナリフラグ (利用する場合 $u^{n,e,k} = 1$ )

$$\text{minimize} \quad \sum_{r^n \in R} \sum_{e \in E} \sum_{k \in T^n} b^n u^{n,e,k} \quad (1)$$

$$\sum_{v_i^n \in S^n} \sum_{v_j^n \in S^n} p_{i,j}^{n,e,k} \leq M u^{n,e,k} \quad r^n \in R, e \in E, k \in T^n \quad (2)$$

$$\sum_{v_i^n \in S^n} \sum_{v_j^n \in S^n} p_{i,j}^{n,e,k} \geq u^{n,e,k} \quad r^n \in R, e \in E, k \in T^n \quad (3)$$

$$\sum_{v_i^n \rightarrow e} p_{i,j}^{n,e,k} = 1 \quad r^n \in R, k \in T^n, v_i^n \in S^n, v_j^n \in S^n \quad (4)$$

$$\sum_{e \rightarrow v_i^n} p_{i,j}^{n,e,k} = 0 \quad r^n \in R, k \in T^n, v_i^n \in S^n, v_j^n \in S^n \quad (5)$$

$$\sum_{e \rightarrow v_j^n} p_{i,j}^{n,e,k} = 1 \quad r^n \in R, k \in T^n, v_i^n \in S^n, v_j^n \in S^n \quad (6)$$

$$\sum_{v_j^n \rightarrow e} p_{i,j}^{n,e,k} = 0 \quad r^n \in R, k \in T^n, v_i^n \in S^n, v_j^n \in S^n \quad (7)$$

$$\sum_{e \rightarrow v} p_{i,j}^{n,e,k} = \sum_{v \rightarrow e} p_{i,j}^{n,e,k} \quad r^n \in R, k \in T^n, v \notin v_i^n, v_j^n, v_i^n \in S^n, v_j^n \in S^n \quad (8)$$

$$p_{i,j}^{n,e,k} = p_{i,j}^{n,e,k+1} \quad r^n \in R, e \in E, k \in T^n, v_i^n \in S^n, v_j^n \in S^n \quad (9)$$

$$\sum_{r^n \in R} b^n u^{n,e,k} \leq C \quad e \in E, k \in T^n \quad (10)$$

#### 4. ネットワークスライス自動構築システム

本研究では、提案したスケジューリング手法を組み込んだネットワークスライス自動構築システムのプロトタイプを実装した。本システムの概要を図 2 で示す。本システ

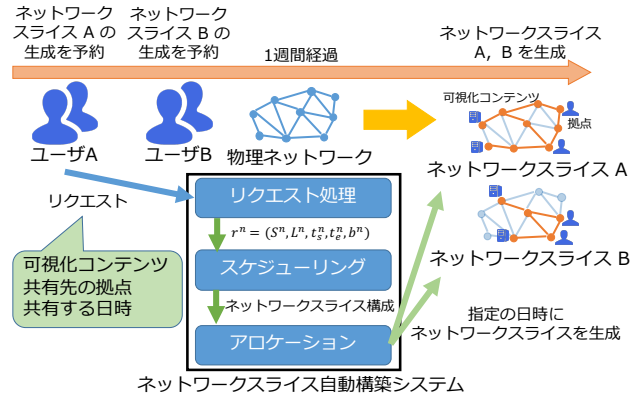


図 2 ネットワークスライス自動構築システムの概要

ムは、ユーザグループにとって簡易的なリクエスト形式の提供を実現するリクエスト処理モジュール、最適なネットワークスライス構成を算出するスケジューリングモジュール、ネットワークスライス生成を実行するアロケーションモジュールの3つのモジュールで構成される。まず、ユーザグループが本システムに対してリクエストを送信した場合、リクエスト処理モジュールがそのリクエストを受け付ける。受け付けたリクエストは、スケジューリングモジュールが要求するリクエスト形式に変換され、スケジューリングモジュールに渡される。スケジューリングモジュールは、提案手法の1段階目のスケジューリングを実行し、算出したネットワークスライス構成を記録する。ユーザグループからのリクエストが少ない時間帯に、提案手法の2段階目のスケジューリングを実行し、記録したネットワークスライス構成を上書きする。リクエストでユーザグループが指定した時間になった場合、アロケーションモジュールが記録したネットワークスライス構成を元にネットワークスライスをネットワーク上に生成する。指定した時間が経過した場合は、生成したネットワークスライスをネットワーク上から削除する。各モジュールの詳細については、以下で説明する。

本システムでは、以下の項目を入力するためのユーザインタフェースを想定する。

- 共有する可視化コンテンツ
- 共有先拠点
- 共有する日時

リクエスト処理モジュールでは、上述した値より、以下の項目を抽出する。

- 共有拠点の仮想ノード集合
- 共有拠点間の仮想リンク集合
- 共有する日時
- 要求帯域幅

仮想ノード集合は、共有する可視化コンテンツと共有先の拠点が位置するノードの集合とする。仮想リンク集合は、共有する可視化コンテンツが位置するノードと共有先の拠



点が位置するノードをそれぞれつなぐリンクの集合とする。共有する日時は、ユーザグループが指定した日時をそのまま利用する。帯域幅は、各可視化コンテンツが必要とする帯域幅の合計とする。各可視化コンテンツが必要とする帯域幅については、本研究では簡易化のために既知のものとした。

スケジューリングモジュールでは、リクエスト処理モジュールから受け取ったリクエストに対して、提案手法を実装したスケジューリングを実行する。スケジューリングの内容は、3章で述べた通りである。提案手法の2段階目で計算する必要のある整数計画問題は、オープンソースウェアであり、利用の簡単な最適化問題ソルバーである GLPK (GNU Linear Programming Kit) [6] を用いて解く。整数計画問題がリクエストの少ない時間帯だけで完了するとは限らないので、本システムではスケジューリングの計算時間の上限を6時間と設定し、それを越えた時点の実行可能解をスケジューリングの結果とする。スケジューリングが算出し、記録したネットワークスライス構成は、アロケーションモジュールと共有する。

アロケーションモジュールでは、スケジューリングモジュールで算出したネットワークスライス構成を元にネットワークスライスの生成を実行する。ネットワークスライス生成の手法としては1章でも紹介した Flowvisor の他にも、AutoSlice [3] や、OpenSlice [5]、AutoVFlow [9] など数多くの手法が存在する。本システムでは、ネットワークスライス生成手法の中でも、早期に開発され、単純な作りで扱いが容易である Flowvisor を用いる。Flowvisor を利用して、ネットワークスライス構成に指定される物理リンクが割り当てられたネットワークスライスを生成する。

## 5. 評価

本研究では、提案するスケジューリング手法に関して、より多いユーザグループに対するネットワークスライスの提供と、短い時間でのリクエスト受理の成否の伝達の2つの指標を評価するために、提案手法を組み込んだネットワークスライス自動構築システムのシミュレーション実験を行った。シミュレーションにおいては、2つの指標を評価するために、スケジューリングにおいて帯域幅の不足などによりネットワークスライス生成の予約ができなかった確率を指すブロック率と、ユーザグループがリクエストを出してからその成否を受け取るまでの時間であるシステム応答時間を測定した。ブロック率に関しては、比較のために、提案手法を用いた場合と、1段階目のヒューリスティック法のみを用いた場合のそれぞれについてシミュレーション実験を行った。提案手法のブロック率がより低いということは、それだけ多くのリクエストを受け付けたということであり、より多いユーザグループへのネットワークスライスの提供を達成したことを示す。

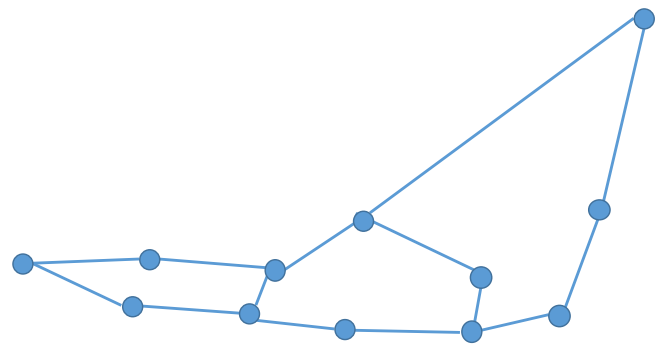


図3 想定するネットワーク

シミュレーション実験の想定について説明する。まず、本システムは図3で示すネットワーク上でネットワークスライスを提供すると想定する。このネットワークは、学術ネットワーク SINET [11] を模式化した図であり、各物理リンクのリンク帯域は10 Gbpsとする。これにより、実環境に近いシミュレーションが行える。ネットワーク上の各ノードは、それぞれが可視化コンテンツを持つ研究拠点であると想定し、リクエスト内では全てのノードが可視化コンテンツと共有先の拠点として指定可能であるとする。各拠点の持つ可視化コンテンツの解像度は、高解像度なものほどその数が少なくなるようにした上でランダムに設定する。ユーザグループがリクエスト内で指定する可視化コンテンツと共有先の拠点については、それぞれ指定する数をランダムに決定したのち、指定する内容もランダムに決定する。これらのランダムに条件を決定するものについては、様々な状況を想定するために各シミュレーションにつきそれぞれ異なる条件を用意する。ただし、提案手法とヒューリスティック法のみを正しく比較するために、それぞれの手法を適用するシミュレーションで同じ条件を使用したシミュレーションの結果同士のみを比較する。

シミュレーション実験の内容について説明する。シミュレーションでは、シミュレーション内の数日の間だけリクエストを受け付けるを行う。シミュレーション内の1日あたりにユーザグループが送信するリクエストの数については、ブロック率に影響を与えることが想定されるため、10, 15, 20, 25, 30にそれぞれ値を変えてシミュレーションを実施する。シミュレーション内の1日では、事前に決めた1日あたりに送信するリクエスト数だけ送られるリクエストを順に処理し、そのシミュレーションが提案手法を適用する場合は最後に整数計画法によるスケジューリングを行う。この処理の過程において、それぞれのリクエストを受け付けてからスケジューリングが終了するまでの間をシステム応答時間として測定する。また、各リクエストに対してスケジューリングが失敗し、かつ、リコメンドも失敗した回数を測定する。リコメンドが成功し複数の日時が提示された場合は、その中からランダムに1つ日時を選び、共有する日時をその日時に変更したリクエストを受理したも

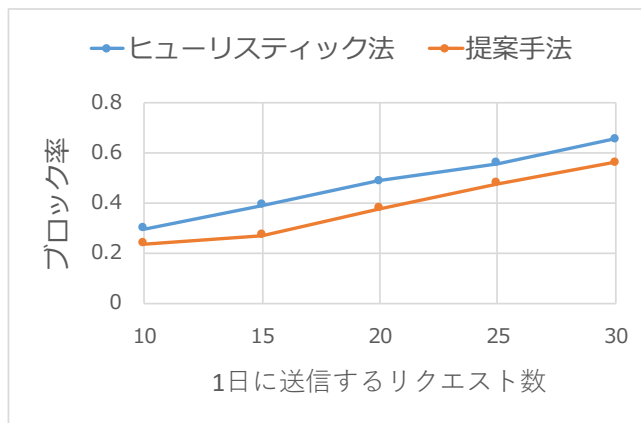


図 4 ブロック率の測定結果

のとして扱う。

シミュレーション実験の結果について説明する。各シミュレーションの試行回数は10回とした。システム応答時間の結果は、平均 25.3 ms となり、十分に短い時間と言える。提案手法とヒューリスティック法のみの方のブロック率については、図 4 の通りである。ブロック率は、スケジューリングに失敗した回数を送信された全リクエストの数で割ることで算出する。結果からは、1日に送信するリクエスト数が増えるに従ってどちらの手法も増加していることがわかる。この増加は、ネットワークスライスに割り当て可能なリンク帯域幅が有限なために発生する。また、結果からは、提案手法のブロック率がヒューリスティック法のみの方のブロック率を全体的に下回っていることがわかる。これは、提案手法で実行される整数計画法によってネットワークスライスに割り当てられる帯域幅やトポロジが最適化され、より多くのネットワークスライスが提供できたことを示す。

## 6. まとめ

本研究では、可視化コンテンツ共有を行うユーザからのリクエストに応じた、論理ネットワーク自動構築システムのプロトタイプを提案した。ネットワークはスケジューリングによって生成され、スケジューリングにはヒューリスティック手法と整数計画法の二段階の組み合わせの実装を行った。提案手法の評価では、シミュレーション実験を実施し、システム応答時間とブロック率をそれぞれ測定した。結果として、システム応答時間は平均 25.3 ms と十分に短い時間となり、ブロック率は提案手法がヒューリスティック法のみの方の手法を全体的に下回った。これは、より多くのユーザグループに対してネットワークスライスの提供が可能であることを示す。これらのことから、提案手法は、リクエスト受理の成否の伝達までの時間が短く、より多いユーザグループにネットワークスライスの提供が可能であると確認できた。

謝辞 本研究は JSPS 科研費 JP26540053 の助成を受け

たものです。

## 参考文献

- [1] Bai, H. et al.: Virtual Network Scheduling Design, *Proceedings of 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet 2014)*, pp. 362–367 (2014).
- [2] Bai, H. et al.: Overlay Network Scheduling Design, *Computer Communications*, Vol. 82, pp. 28–38 (2016).
- [3] Bozakov, Z. and Papadimitriou, P.: AutoSlice: Automated and Scalable Slicing for Software-Defined Networks, *Proceedings of the 2012 ACM Conference on CoNEXT Student Workshop*, pp. 3–4 (2012).
- [4] Gu, F., Xie, C., Peng, M., Cavdar, Ç., Khan, S. and Ghani, N.: Virtual Overlay Network Scheduling, *IEEE Communications Letters*, Vol. 15, No. 8, pp. 893–895 (2011).
- [5] Liu, L., Muñoz, R., Casellas, R., Tsuritani, T., Martínez, R. and Morita, I.: OpenSlice: an OpenFlow-Based Control Plane for Spectrum Sliced Elastic Optical Path Networks, *Optics Express*, Vol. 21, No. 4, p. 4194 (2013).
- [6] Makhorin, A.: GLPK - GNU Project - Free Software Foundation (FSF), [http://www.mext.go.jp/a\\_menu/shinkou/sangaku/1380184.htm](http://www.mext.go.jp/a_menu/shinkou/sangaku/1380184.htm).
- [7] Renambot, L., Marrinan, T., Aurisano, J., Nishimoto, A., Mateevitsi, V., Bharadwaj, K., Long, L., Johnson, A., Brown, M. and Leigh, J.: SAGE2: A Collaboration Portal For Scalable Resolution, *Future Generation Computer Systems*, Vol. 54, pp. 296–305 (2016).
- [8] Sherwood, R., Gibb, G., Yap, K.-K., Appenzeller, G., Casado, M., McKeown, N. and Parulkar, G.: Flowvisor: A Network Virtualization Layer, *OpenFlow Switch Consortium, Technical Report*, pp. 1–13 (2009).
- [9] Yamanaka, H., Kawai, E. and Shimojo, S.: AutoVFlow: Virtualization of Large-scale Wide-area OpenFlow Networks, *Computer Communications*, Vol. 102, pp. 28–46 (2016).
- [10] 宮代隆平, 松井知己: ここまで解ける整数計画, システム制御情報: システム制御情報学会誌, pp. 363–368 (2006).
- [11] 栗本崇, 漆谷重雄: 学術情報ネットワーク (SINET5) の概要 (ネットワークシステム), 電子情報通信学会技術研究報告, Vol. 116, No. 44, pp. 41–46 (2016).