

帯域予約型サービスにおける効率的な資源分配アルゴリズムの検証

川崎 考蔵

Kozo Kawasaki

1 はじめに

学術情報ネットワークなどの先進的なインターネットでは、増大するユーザからの需要 (遠隔会議, 遠隔医療などのリアルタイム性が要求される大規模データ転送等) により、高品質サービスとして柔軟な資源配分による QoS, Bandwidth-on-Demand (BoD) などが求められている^{5) 6)}。これらの先進的インターネットにおいては、資源を効率的に利用することが求められる。そこで、任意の通信拠点間のネットワークを専用線のように利用する事が可能なレイヤ1帯域オンデマンドサービス (Layer-1 BoD)^{1) 2)} が注目されている。Layer-1 BoD では、ユーザに任意の拠点間におけるネットワーク帯域を任意の時間帯で提供する³⁾。ユーザは、使用したい拠点間、帯域、日時等をリクエストとして指定することにより、その時間帯で、そのユーザ専用のネットワーク帯域を専用線と同じように利用できる。このようなサービスにおいては、ユーザの満足度や公平性を考慮し、ネットワーク資源を効率的に運用することが求められる。一方で、大規模ネットワークの packet 通信を対象ネットワーク資源を効率的に運用する研究が盛んに行われている^{7) 8) 9)}。しかしながら、Layer-1 BoD においては従来の packet トラフィックにおける資源割当ての最適化ではなく、各ユーザの帯域確保要求であるリクエストに対して、どのリンクを割り当てるのか、また全リクエストを受理できない状況において、どのユーザのリクエストを受理、または却下するのかを公平性、資源利用効率の両面において考慮する必要がある。本研究は Layer-1 BoD を対象とし、そのネットワークリソースを複数のユーザに効率的に分配する方法を検討する。

2 Layer-1 BoD

2.1 Layer-1 BoD の概要

L1 BOD では、ユーザに任意の拠点間におけるネットワーク帯域を任意の時間帯で提供する。ユーザは、使用したい拠点間、帯域、日時等を指定することにより、その時間帯で、そのユーザ専用のネットワーク帯域を専用線と同じように使うことができる。このサービスにより、ユーザはネットワークリソースを高品質で利用することが可能となる。これは、従来は高品質を得るために専用線を用いてきたユーザをインターネットバックボーンに収容することを可能にする技術であり、ネットワークの資源利用効率を向上させることも可能である。

2.2 Layer-1 BoD の構成

Layer-1 BoD の構成を述べる。Layer-1 BoD のサービスイメージを Fig. 1 に示す。Layer-1 BoD は大きく分けて「ユーザ」「バックボーンネットワーク」「オンデマンドサーバ」により構成される。

ユーザはユーザ装置を通してネットワークをどのように利用したいかを条件付けし、リクエストとしてオンデマンドサーバにサブミットする。

バックボーンネットワークはユーザが実際に通信を行う際利用する回線 (リンク) とそれらを繋ぎ、制御する拠点 (ノード) 群である。

オンデマンドサーバはユーザからのリクエストを受けて保持するキューと、それらが利用するバックボーンネットワークのリソースの制御機能からなる。もし各リクエストが実行可能であれば、それらが利用するバックボーンネットワークのリソースを割り当て、実行不可能なリクエストがあればその旨をユーザに通知しリジェクトする。

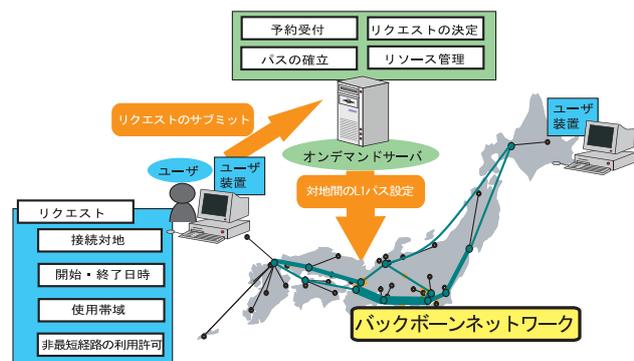


Fig.1 Layer-1 BoD の構成

2.3 Layer-1 BoD の処理プロセス

想定する Layer-1 BoD の処理プロセスを、あるユーザの希望パスのリクエストからパス確立において述べる。

- 予約開始
 - ユーザは Web ベースのインタフェースを介してオンデマンドサーバに帯域確保のリクエストを行う
- リクエストの作成
 - ユーザは以下のような項目を設定する
 - 通信発着ノード
 - パスを確立する開始拠点と終了拠点
 - 利用日時
 - パスを確立する開始時間と終了時間
 - 利用帯域

パスを確立する際、どの程度通信帯域を確保するか

– 経路条件

ネットワーク通信の遅延時間を最小にするネットワーク経路を希望するか、遅延時間が最小ではない経路で通信を行うことを許可するか

● リクエストの送信

ユーザはオンデマンドサーバにリクエストを送信する

● 要求受付完了通知

オンデマンドサーバは、上記の一連のプロセスを終了すると、ユーザの要求受付を完了し、そのユーザのリクエストを自身が持つキューにスケジューリング契機まで保持する。また、その旨をユーザに通知する

● スケジューリング

オンデマンドサーバは、サブミットされた各ユーザのリクエストをスケジューリング契機まで保持する。スケジューリング契機になると保持しているリクエストに関して、独自のアルゴリズムでネットワークリソースを分配する。そして、要求している時間、経路、帯域でパスの確保が可能かどうかを判断する。もし、パス確立が不可能なリクエストが出現した場合は、そのリクエストは却下する。

● 予約成立または不成立の通知

オンデマンドサーバは、あるリクエストについて予約の成立・不成立の旨をそのリクエストをサブミットしたユーザに通知する

● アベイラブルリソースの再計算

オンデマンドサーバは、ある予約が受理されると各リンクの使用予定状況を再計算する

● パスの確立

オンデマンドサーバは、受理した各リクエストのパス確立の希望使用開始時間となれば、そのリクエストが希望する接続対地間のリンクのパスを希望する帯域分だけ確立する

● パスの解放

オンデマンドサーバは、受理した各リクエストのパス確立の希望使用終了時間となれば、そのリクエストが利用している対地間のリンクのパスを解放する

3 スケジューリングアルゴリズムの検討

3.1 スケジューリングの要求

Layer-1 BoD は不特定多数のユーザを想定したシステムである。そのため、複数のユーザで帯域を確保したい拠点間の使用リンク、使用日時が重複すると、全ユーザの希望した通りにネットワークを確保することが不可能な状況が生じる場合がある。そのような状況が発生した場合、どのユーザのリクエストを受理し、またどのユーザのリクエストを却下するかを決定するリクエストスケジューリングが重要となる。具体的にはいかのようなことが求められる。

1. 全体のスループットを高く保つ

常にリクエストの処理数や、各リンクの使用率を高く保つことによって、サービスのスループットを高く保つことが望まれる。そのためには、どのユーザのリクエストを受理し、それぞれのリクエストに対し、どのようにネットワークリソースを提供するかということが非常に重要となる。

2. 全ユーザが平等にネットワークリソースを利用する

Layer-1 BoD では複数のユーザがネットワークリソースであるリンクの帯域を共有して利用する。そのため、ある特定のユーザによって常にリソースが占有され、他のユーザがリンクを利用することができないという状況は避けるべきである。

本報告においては、上記のことを考慮しリンクを効率的に利用し、各ユーザの満足度を高く保つことを目標にしたスケジューリングアルゴリズムを検討する。

3.2 評価値の設定

スケジューリングアルゴリズムを考慮する際、その評価を行うために、評価値を導入する。評価値は次のように設定する。

$$Fitness = \sum_{i=1}^n (W_i^{-1}(t) \times \rho_i)$$

$$W_i(t) = 0.5(dt/h) \times W_i(t-dt) + \{1 - 0.5(dt/h)\} \times \rho_i$$

$$\rho_i = D_i \times T_i$$

上式はユーザが n 人存在したときの評価値の計算方法である。ユーザ i が要求する帯域を D_i 、使用時間(使用終了時間 - 使用開始時間)を T_i としたとき、それらの積をユーザ i のネットワーク利用度 ρ_i と定義する。ネットワークを効率的に利用するという観点においては、各ユーザのネットワーク利用度の和を最大とするスケジューリングが求められると考えられる。しかしながら、各ユーザのネットワーク利用度を最大化するスケジューリングでは、ネットワークを利用できるユーザに偏りが生じ、平等性を保つという目的に反してしまう状況が発生すると思われる。そこで、過去のネットワーク利用度を考慮に入れた重み W_i をスケジューリングシステムである Condor¹⁰⁾ を参考にパラメータとして導入する。Condor は分散コンピューティング環境において、効率かつ平等に計算資源を利用することを目的としたジョブスケジューリングシステムである。Condor は各ユーザに平等にリソースを分配するため、過去のリソース利用を考慮した優先度決定メカニズム¹¹⁾ を用いている。このメカニズムは、Layer-1 BoD のユーザにリソースを平等に分配する上で非常に有用であると考えられる。 $W_i(t)$ は時刻 t におけるユーザ i の重みであり、この値は、前回のスケジューリング時刻におけるユーザ i の重み $W_i(t-dt)$ と、時刻 t において得たネットワーク利用度 ρ_i に大きく依存する値である。この値はユーザがネットワークを利用した際に上昇し、時刻を経るごとに減少する。またこ

の減少する割合を半減期 h により決定することが可能である。Fitness は、各ユーザのネットワーク利用度とその重みの積で得た値の総和によって決定し、本報告では、この Fitness の値を高くするリクエストスケジューリングを目指す。これにより、各ユーザがネットワークを効率的かつ平等に利用することが可能であると考えられる。

3.3 提案スケジューリングアルゴリズム

本報告では、上記の条件を満たすためのスケジューリングアルゴリズムとして、Figure2 に示すアルゴリズムを提案する。

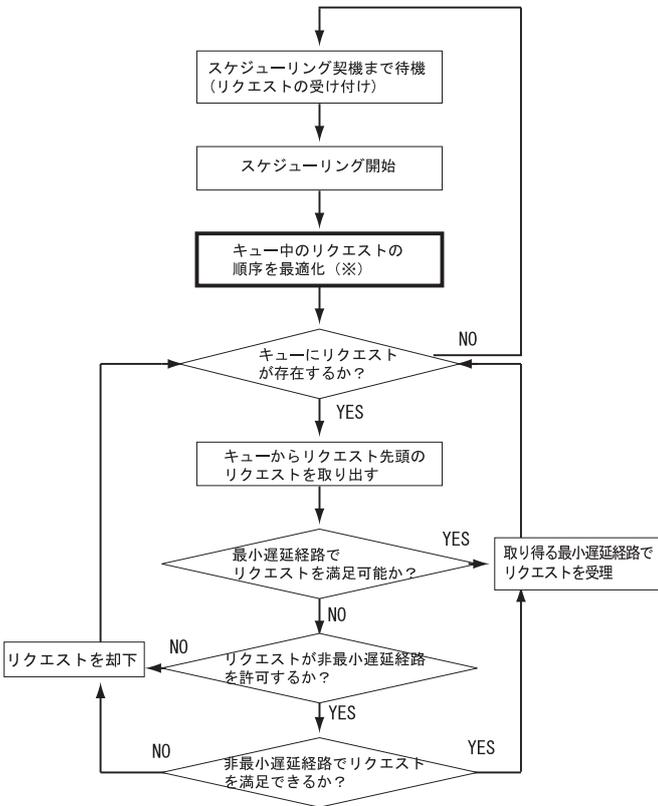


Fig.2 提案スケジューリングアルゴリズム

本スケジューリングアルゴリズムは、()のプロセスにおいて処理するリクエストの順序を決定し、決定した処理順序によって各リクエストを受理するかかを決定する。また、リクエストが両端拠点まで、どのリンクを利用するかを決定するルーティングルールとして、リクエストを処理する時点で取り得る最小遅延経路を選択することとする。このため、()のプロセスにおいて決定するリクエストの処理順序は非常に大きく評価に影響を与える。

よって、次節からは()のプロセスにおけるリクエスト処理順序の最適化の方法を述べる。

3.3.1 GA によるリクエスト順序最適化

Genetic Algorithm(GA)^{12) 13)} はデータ(解の候補)を遺伝子で表現した「個体」を複数用意し、適応度の高い個体を優先的に選択して交叉(組み換え)・突然変異などの操作を繰り返しながら解を探索する。適応度は適応度

関数によって与えられる。

本アルゴリズムにおいては、個体を「リクエスト処理順序」とし、交叉手法には巡回セールスマン問題などの離散問題を解く際に代表的に利用される「順序交叉¹⁴⁾」を利用した。また、適応度関数は全ユーザの満足度の和とする。

3.3.2 LS によるリクエスト順序最適化

Local Search(LS)¹⁷⁾ は近似アルゴリズムの中でも最も単純なアルゴリズムの枠組みの一つである。LSは現在の解の近傍の内一つをある条件で選び近傍解とする最適化手法である。本報告においては、次候補は現在の解の2-Change¹⁵⁾を行うことにより作成する。

3.3.3 GA によるリクエスト順序最適化

Greedy Algorithm(GR)^{15) 16)} とは問題の要素を複数の部分問題に分割し、それぞれを独立に評価を行い、評価値の高い順に取り込んでいくことで解を得るという方法である。GRでは一度選択した要素を再考する事は無い。

3.3.4 ランダムによるリクエスト順序決定

上記のアルゴリズムの比較を明確にするために、Fitnessを全く考慮せず、ランダムにキュー中のリクエストを並び替えるアルゴリズムを実装し、検証した。

4 数値実験

本報告では、自作のLayer-1 BoDを模したシミュレータにより、各アルゴリズムがどのような振る舞いを見せ、どのような問題が生じるのかを解析した。そしてスケジューリングする上で、どのような部分に焦点を当ててスケジューリングアルゴリズムの改良を行うべきかを検討する。

4.1 実験環境

数値実験として、Figure 3 に示すような、複数のノード、リンクを持つようなトポロジーにおいて、各アルゴリズムによるスケジューリングを解析する。この環境においては、1つのリンクをどのようなリクエスト群を受理すれば、ネットワークリンクを効率的に運用できるかを確認し、評価値がスケジューリング結果にどのように影響を与えるのかを確認する目的がある。本報告におけるGA及びLocal SearchのパラメータをTable1に示す。また、各リンクは共通して最大利用帯域が100Mbpsであり、レイテンシは3msecとした。

Table2 ユーザのサブミットパターン

Group	UserName	Source	Destination	Allowance Latency	Bandwidth (Mbps)	LinkUp Time(step)	Communication Time(step)	Submit Interval(step)
I	user0	node6	node9	FALSE	9	6	6	100
	user1	node9	node4	TRUE	8	8	7	100
	user2	node3	node8	TRUE	9	5	7	100
	user3	node1	node8	FALSE	10	8	2	100
	user4	node2	node1	FALSE	6	6	11	100
	user5	node6	node4	TRUE	6	10	6	100
	user6	node4	node1	TRUE	10	10	12	100
	user7	node2	node9	FALSE	7	10	2	100
user8	node7	node8	FALSE	9	10	7	100	
II	user9	node6	node8	FALSE	58	6	6	100
	user10	node5	node1	TRUE	60	9	9	100
	user11	node4	node7	FALSE	56	7	6	100
	user12	node2	node5	TRUE	51	7	9	100
	user13	node5	node7	TRUE	50	8	9	100
	user14	node6	node4	TRUE	57	6	5	100
	user15	node7	node6	FALSE	56	8	12	100
	user16	node9	node3	TRUE	55	6	6	100
user17	node9	node4	FALSE	60	9	12	100	
III	user18	node1	node5	FALSE	66	5	25	100
	user19	node4	node5	FALSE	80	7	24	100
	user20	node8	node4	TRUE	71	9	13	100
	user21	node5	node2	FALSE	76	9	15	100
	user22	node1	node6	FALSE	63	9	22	100
	user23	node9	node1	TRUE	69	9	12	100
	user24	node1	node7	FALSE	73	5	23	100
	user25	node6	node8	FALSE	80	6	24	100
user26	node7	node4	TRUE	65	7	19	100	
IV	user27	node1	node7	FALSE	28	9	13	100
	user28	node6	node3	TRUE	15	9	8	100
	user29	node3	node4	TRUE	27	7	13	100
	user30	node4	node2	FALSE	17	7	11	100
	user31	node3	node8	FALSE	29	9	9	100
	user32	node7	node6	FALSE	30	10	11	100
	user33	node6	node3	FALSE	20	5	17	100
	user34	node1	node5	FALSE	26	9	12	100
user35	node7	node3	TRUE	24	5	15	100	

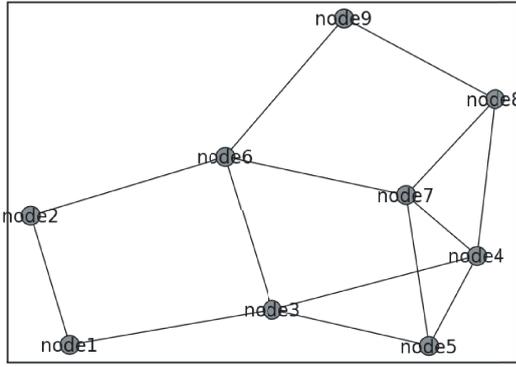


Fig.3 数値実験におけるトポロジー

Table1 パラメータ

GA	Population size	20
	Generation num	200
	Crossover rate	0.7
	Mutation rate	0.3
	Local Search	Trial num

次に、ユーザの行動規則を表したデータを Table 2 に示す。「開始ノード」、「終端ノード」により通信を行う拠点間を決定している、また、「レイテンシの許可」は通信遅延時間が非最短の場合でも許可するかを示すものである。「利用帯域」、「利用時間」はそれぞれ、通信を行う際に利用する帯域と時間を表す。「使用開始時間」はサブミットした時点から何 step 後に通信を開始したいかを表した数値である。「待機時間」はサブミットを行う時間間隔を表している。本報告では、スケジューリングが必ず発生するように「待機時間」は全ユーザ共通とする。また、より分析を行いやすいよう、ユーザを「使用帯域」と「使用時間」により大きく 4 つのグループに分類した。Table 2 における点線はそれらのグループを表す。上から「使用帯域」、「使用時間」共に小さいユーザ群 (Group I)、「使用帯域」は大きい「使用時間」が短いユーザ群 (Group II)、「使用帯域」、「使用時間」共に大きいユーザ群 (Group III)、「使用帯域」は小さい「使用時間」が大きいユーザ群 (Group IV) に分類した。

4.2 実験結果 1

上記の実験環境において、1 度のスケジューリングで、どのような評価が得られていたのかを確認する。各アルゴリズムにおいてスケジューリングをそれぞれ 5 回ずつ行い、その評価をとった結果を Figure 4 に示す。また、GA と LS の探索の結果を Figure 5 に示す。また、Table 3 に、各手法において、それぞれのリクエストがどのようにルーティングが割り当てられたかを示す。Table 3 は、各手法において、各ユーザのリクエストが、どのようなノードを経由し、情報を伝えるかを示したものであり、経由ノードが多い程多くのリンクを使用することを示す。また、空白表示の欄は、そのユーザのリクエストが却下された結果であることを示す。

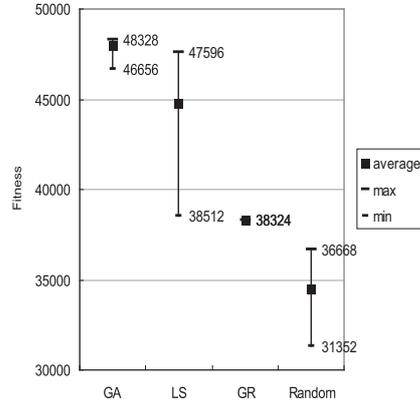


Fig.4 各手法の評価値

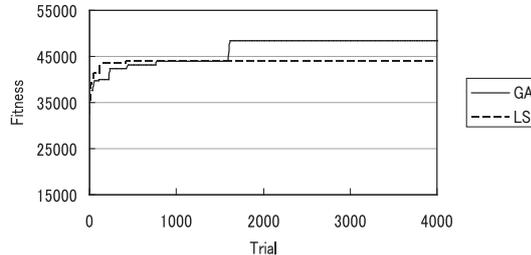


Fig.5 GA と LS の探索

Figure 4 から、GA, LS, GR, Random の順で評価が高いことが分かる。本実験においては、一度のみのスケジューリングであるので、各ユーザの過去の利用は考慮しない、そのため評価値は各ユーザのネットワーク利用度 ρ の総和に等しく、Figure 4 の結果は、いかにネットワークリソースを利用できているかを示した数値である。このことから、評価値が高いアルゴリズムはネットワークリソースを有効に利用していると言える。Fig4 により GA と LS の探索を比較すると、LS での解探索は 500Trial 付近で局所解に陥ってしまい、それ以降解交換が行われていない。また、GA は 800Trial 付近から LS

Table3 各手法のルーティング

Group	User	GA	LS	GR	Random
I	user0	node6, node9	node6, node9		node6, node9
	user1	node9, node8, node4	node9, node8, node4	node9, node8, node4	node9, node8, node4
	user2	node3, node4, node8	node3, node4, node8	node3, node5, node7, node8	node3, node1, node2, node6, node7, node8
	user3	node1, node3, node4			
	user4	node2, node1	node2, node1	node2, node1	
	user5	node6, node3, node4	node6, node3, node4	node6, node7, node5, node4	node6, node9, node8, node4
	user6	node4, node3, node1		node4, node5, node7, node6, node2, node1	node4, node7, node6, node2, node1
	user7	node2, node6, node9	node2, node6, node9	node2, node6, node9	node2, node6, node9
user8	node7, node8	node7, node8	node7, node8	node7, node8	
II	user9				
	user10				
	user11				
	user12				
	user13	node5, node7	node5, node7		node5, node7
	user14				node6, node3, node4
	user15				
user16	node9, node6, node3	node9, node6, node3			
user17					
III	user18	node1, node3, node5	node4, node5	node4, node5	
	user19	node4, node5	node8, node4		
	user20	node8, node4			
	user21		node5, node3, node1, node2		
	user22	node1, node2, node6		node1, node2, node6	node1, node2, node6
	user23				node9, node8, node4, node5, node3, node1
	user24			node1, node3, node4, node7	
	user25	node6, node7, node8	node6, node7, node8	node6, node7, node8	node6, node7, node8
user26	node7, node4	node7, node4	node7, node5, node3, node6, node9, node8, node4	node7, node4	
user27					
IV	user28	node6, node3	node6, node3		node6, node3
	user29	node3, node4	node3, node4	node3, node6, node9, node8, node4	node3, node5, node4
	user30	node4, node3, node1, node2	node4, node3, node1, node2		node4, node3, node1, node2
	user31				
	user32				
	user33	node6, node3	node6, node3		node6, node3
	user34			node1, node3, node5	
	user35	node7, node4, node3	node7, node4, node3	node7, node4, node3	node7, node4, node3

と同程度の評価値で局所解に陥っているが、1700Trial付近で解交換が行われている。しかし、それ以降は解交換は行われていない。これらのことから、本実験の環境は、局所解に陥りやすい問題であると考えられる。また、Table6 から、GA や LS は各ユーザが利用するリンク数が比較的少なく、GR, Random では比較的使用するリンクが多いことが分かる。GR は ρ の値が大きいユーザを優先するにも係らず GA よりも ρ が大きいグループの受理リクエスト数が少ない。これは、リクエストを受理した際のルーティングが大きく影響しているためと考えられる。特に user26 のリクエストは最小で1つのリンクで通信を行えるが、GR では6つのリンクを利用している。これは、非常に効率の悪い資源割り当てであることを意味する。Random においても同様に GroupII のような ρ が大きいリクエストを非効率的なルーティングで受理している。一方 GA は、GroupIII のような ρ の大きいリクエストをより少ない使用リンクで受理していることにより、効率的にリソースを運用している。このことから GA は LS, GR, Random よりも効率的なスケジューリングを行っているということが分かる。しかし、全てのアルゴリズムに関して GroupII のユーザは受理数が少ない。これは本実験の *Fitness* の設定では、GroupII のようなユーザが他のユーザと競合した際、受理されにくいことを示す。

4.3 実験結果 2

次に、各ユーザの平等性を保つという観点のもと、過去のネットワークリソースの使用状況を考慮したスケジューリングが行われるかを解析するために、一度のシミュレーションに複数のスケジューリングが発生させる状況において実験を行った。各ユーザはあるパターンによりリクエストをサブミットする。サブミットする時間間隔は Table 2 に示すように全ユーザで共通で 100step である。また、全体のシミュレーションは 1200step 行った。そのため、本報告でのスケジューリングは 12 回行った。

Table 4 に、1200step までの各ユーザのネットワーク利用度 ρ の累積が最も大きいユーザ、最も小さいユーザ、中央値のユーザ、全ユーザの総和を示した。また、Table 5 に全てのユーザのリクエストの受理回数とその総和、分散の値を示す。

Table4 ネットワーク利用度 (ρ) の累積

Algorithm	Total	Max	Min	Median
GA	107191	21120	120	1635
LS	99609	15360	168	1605
GR	114972	23040	0	0
Random	94936	17280	0	1545

Table 4 を見ると、Total の値は、GR, GA, LS, Random の順で高い。これは、スループットの観点において、この順序で性能が高いことを示している。しかしながら、Max, Min, Median の値を見ると、GR では、Min, Median とともに 0 である。これは一度も受理されないユーザが存在することを示し、リクエストが受理されるユーザに大きく偏りがあることを示している。同様に Random アルゴリズムにおいても一度も受理されていないユーザが存在している。一方で、GA, LS では、一度も受理され

Table5 ユーザのリクエスト受理回数

Group	User	GA	LS	GR	Random
I	user0	12	12	0	12
	user1	12	12	12	12
	user2	12	12	12	5
	user3	6	9	0	2
	user4	12	12	12	2
	user5	12	12	12	7
	user6	10	11	12	8
	user7	12	12	12	2
	user8	12	12	12	5
II	user9	3	3	0	3
	user10	3	4	0	1
	user11	5	4	0	5
	user12	3	3	0	12
	user13	10	9	0	3
	user14	4	4	0	0
	user15	4	4	0	5
	user16	5	5	0	4
	user17	6	6	0	2
III	user18	3	2	0	4
	user19	11	8	12	6
	user20	5	5	0	3
	user21	3	2	0	12
	user22	4	4	12	11
	user23	3	3	0	8
	user24	3	3	12	9
	user25	5	4	12	5
	user26	5	6	12	8
IV	user27	4	3	0	9
	user28	12	11	0	6
	user29	8	10	12	9
	user30	4	5	0	11
	user31	5	5	0	12
	user32	7	8	0	12
	user33	12	12	0	12
	user34	3	5	12	12
	user35	10	11	12	1
Total		250	253	180	240
variance		13.4	13.5	36.0	15.8

ていないユーザは存在せず、中央値 GR, Random よりも高い。これは、公平性を保ちながら他のアルゴリズムよりスループットを高く保っていることを意味する。

また、Table 5 を見ると、全ユーザのリクエストの受理回数は LS, GA, Random, GR の順で多い。GR では同じスケジューリングが繰り返され、受理されるユーザとそうでないユーザで二分されている。また、GA, LS, Random を比較すると Total は LS, GA, Random の順で多い、しかしながら Variance では GA, LS, Random の順で良くなっている。特に GA では最低回数が 3 回であるのに対し、Random では GroupII で一度も受理されないユーザが存在する。これらのことから、GA, LS では GroupII のように受理されづらいユーザでも複数回スケジューリングすると、公平性を考慮したメカニズムが作用し、受理されるようになっていくことが確認できる。

これらのことから、本実験における Fitness はスケジューリングする上でスループット、公平性共に効果的に作用している事が分かる。しかしながら、GR のように受理が偏ってしまう状況から、より環境にあった重み設定が必要であると考えられる。

5 まとめと今後の展望

本報告では、Layer-1 BoD におけるスケジューリングアルゴリズムの提案を行い、そのスケジューリングアルゴリズムの解探索手法として *Fitness* を導入し、GA, LS, GR において比較を行った。検証の結果、スループット、公平性に関して、*Fitness* の高いスケジューリング程、効率よくネットワークを運用できることが分かった。今後の展望として、より効果的な重み設定の検討、及びスループット、平等性において性能の良い解を探索することが可能なアルゴリズムの検討が必要となる。

参考文献

1) SINET3: <http://www.sinet.ad.jp/sinet3/>

- 2) Shigeo Urushidani, et al. "Resource Allocation and Provision for Bandwidth Networks on Demand in SINET3", IEEE BoD 2008 Workshop, Salvador da Bahia, Apr. 2008
- 3) Shigeo Urushidani et al. "Design of Versatile Academic Infrastructure for Multilayer Network Services", IEEE Journal on Selected Areas in Communications (JSAC), Apr. 2009
- 4) <http://www.sinet.ad.jp/service/network/11>
- 5) <http://www.internet2.edu/>
- 6) <http://akari-project.nict.go.jp/overview.htm>
- 7) Hidehiro Kobayashi et al. "Bandwidth Allocation using Genetic Algorithms", Special Issue on Multimedia Communication and Distributed Processing 1999, pp91-96
- 8) Q Ma, P Steenkiste, H Zhang, "Routing high-bandwidth traffic in max-min fair share networks", ACM SIGCOMM Computer Communication Review, Volume 26, Issue 4 (October 1996) pp206 - 217
- 9) L Zhu, RL Wainwright, DA Schoenefeld - Proc. "A genetic algorithm for the point to multipoint routing problem with varying number of requests", IEEE Intl. Conf. on Evolutionary Computation, 1998
- 10) A Bricker, M Litzkow, M Livny - Computer Sciences Department, University of Wisconsin, 1992 - eprints.kfupm.edu.sa
- 11) D Thain, T Tannenbaum, M Livny - Grid Computing: Making the Global Infrastructure a Reality, 2002
- 12) Holland, J.H.: Adaptation in Natural and Artificial Systems, University of Michigan Press (1975)
- 13) Morgan Kaufman, "A Genetic Algorithm for the Point to Multipoint Routing Problem with Varying Number of Request of Genetic Algorithms 4", 1993
- 14) CL Mumford, "An order based evolutionary approach to dual objective examination timetabling", IEEE Symposium on Computational Intelligence in Scheduling, 2007, pp179-186
- 15) P Preux, EG Talbi, "Towards hybrid evolutionary algorithms", International Transactions in Operational Research, 1999
- 16) Cormen, Leiserson, and Rivest, "Introduction to Algorithms", 1990, Chapter 16 "Greedy Algorithms" p. 329
- 17) J Gu, X Huang, "Efficient local search with search space smoothing a case study of the traveling salesman problem (TSP)", IEEE Transactions on Systems, Man and Cybernetics, 1994, pp728-735