特別研究報告

題目

IP over WDM ネットワークにおける統合経路制御手法の実装と評価

指導教員
村田 正幸 教授

報告者
筒井　宣充

平成 20 年 2 月 19 日

大阪大学 基礎工学部 情報科学科

平成 19 年度 特別研究報告

IP over WDM ネットワークにおける統合経路制御手法の実装と評価

筒井　宣充

内容梗概

WDM（波長分割多重: Wavelength Division Multiplexing）ネットワークにおいて、ノード間に光パスを設定することにより論理トポロジを構築し、その上で IP (Internet Protocol) を用いる方式 (IP over WDM) が考えられている。しかし、IP と WDM はそれぞれ独立して経路制御を行うため、WDM ネットワークで設定した光パスが必ずしも IP によって利用されるとは限らず、WDM ネットワークの波長資源を有効に利用することができない。我々の研究グループでは、IP over WDM ネットワークにおける統合経路制御手法を提案し、計算機シミュレーションにより統合経路制御手法の有効性を示している。本報告では、統合経路制御手法を計算機上に実装し、提案された手法の実現性を示す。8 ノードからなる実験ネットワークを構築し、実装したプログラムを動作させ、統合経路制御手法が正しく動作することを確認する。また、実装したプログラムにおける実行時間がどの程度あるのかを計測し、実用上問題にならないことを示す。さらに、統合経路制御手法が動作する IP over WDM ネットワークにおいてデータ転送実験を行い、統合経路制御手法によりノード処理負荷が低減されることも示す。

主な用語

WDM、GMPLS、IP over WDM、波長ルーティング、仮想リンク、OSPF

目 次

図 目 次

表 目 次

1 はじめに

WDM（波長分割多重: Wavelength Division Multiplexing）技術は、一本のファイバ上で複数の波長を多重して伝送することにより大容量通信を実現する光通信技術である。WDM 技術を利用すれば、各ノードにおいて波長レベルでの交換を行い（波長ルーティング）、波長によってのみ構成される光パスがエッジノード間に設定される。ネットワーク内では、光信号を電気信号に変換することなくパケットを転送することが可能になり、ルータでの電気処理による負荷を軽減することができる。

インターネットトラヒックの大部分は IP トラヒックで構成されているため、IP トラヒックを直接 WDM ネットワークに転送する方式（IP over WDM）が考えられている [1, 2]。IP over WDM ネットワークには、オーバーレイモデル、ピアモデルモデルがある。オーバーレイモデルは、IP ネットワークおよび WDM ネットワークのそれぞれにおいて、経路制御を独立して行う方式である。ピアモデルは、IP 層と WDM 層がネットワーク情報を共有し、統一された経路制御手法を用いるのことで、IP over WDM ネットワークの資源の有効活用を図るものである。しかし、IP ネットワークおよび WDM ネットワークのすべての資源利用状況に関する情報を取得する必要がある。したがって、制御オーバーヘッドが大きく拡張性が損なわれる。オーバーレイモデルでは、WDM ネットワークにおいてノード間に光パスを設定することで論理トポロジを構成し、IP ネットワークでは論理トポロジ上で従来の経路制御などのプロトコル処理を行う [3]。この場合、IP プロトコルの変更が必要ないという利点は残されているが、波長ルーティングによって光パスを設定したとしても、IP パケットの転送に利用されない可能性があり、WDM ネットワークで光パスを設定する際に、IP ネットワークの振る舞いを考慮する必要がある。これは、IP パケットの経路は IP の経路制御機構によって決定され、WDM ネットワークで提供される光パスを使用するとは限らないためである。

その一方で、IP/MPLS ネットワークと WDM ネットワーク間でトポロジ情報やリンクステート情報を共有し、単一の経路制御によって、ラベル情報でパケットが転送される経路である LSP（Label Switched Path）を WDM ネットワークに収容する方式も考えられている [1, 2]。MPLS (Multi-Protocol Label Switching) とは、ラベルスイッチング方式を用いたパケット転送技術である。文献 [2] では、IP/MPLS over WDM ネットワークを対象とし、次々と離散的に到着するトラヒックフローに対して、その LSP の経路および光パスの経路を求める MIRA（Minimum Interference Routing Algorithm）を提案している。MIRA では、ある必要とする帯域が明示的に定められたトラヒックフローに対して、その LSP に対する残余帯域に着目し、その残余帯域の最大化を目指して経路を選択する。適切な経路がない場合は LSP 設定要求は棄却される。文献 [1] では、IP/MPLS over GMPLS ネットワー

クにおいて、新たに到着するトラヒックフローの LSP 経路を求める際に、GMPLS ネットワークのトポロジ情報を用いることで、LSP 設定要求の棄却率を改善することが示されている。これらの研究を含め、IP ルーティングと WDM の波長ルーティングの統合化に関する研究では、IP トラヒックは MPLS の LSP にマッピングされ、その LSP と WDM の光パスの統合制御が提案されてきた。この場合、IP トラヒックの変動に対しては、LSP を新たに設定・開放することにより対応することができる。しかし、そのためにはフローの検出およびフロー量の計測が IP ルータにおいて必要となる。

そこで、我々の研究グループでは、IP over WDM ネットワークにおいて、設定した光パス上に IP パケットが確実に転送される統合経路制御手法を提案している [4]。統合経路制御では、光パスが設定されていないノード間に対して仮想リンクを定義し、IP ネットワークでは仮想リンクを含むトポロジにおいて最適な経路を計算する。経路計算によって仮想リンクが選択された場合に、その仮想リンクのノード間に光パスを設定する。これにより、光パスは IP パケットの転送に確実に利用され、WDM ネットワークの波長資源を有効活用することができる。計算機シミュレーションによる評価によって、統合経路制御手法は 50%程度のスループットを改善することを示している。しかし、[4] は計算機シミュレーションによる評価であり、ネットワーク運用上重要となる制御オーバーヘッド、処理遅延については議論されていない。

本報告では、IP over WDM ネットワークにおける統合経路制御手法を計算機上に実装し、文献 [4] で提案された手法の実現性を示す。8 ノードからなる実験ネットワークを構築し、実装したプログラムを動作させ、統合経路制御手法が正しく動作することを確認するとともに、実装したプログラムにおける実行時間がどの程度あるのかを計測し、実用上問題にならないことを示す。さらに、統合経路制御手法が動作する IP over WDM ネットワークにおいてデータ転送実験を行い、統合経路制御手法によりノード処理負荷が低減されることも示す。

本報告の内容は以下の通りである。まず、2 章において WDM ネットワークにおける光パス設定の方法について述べる。3 章において、統合経路制御手法の詳しい説明を行う。4 章では、PC に統合経路制御手法を組み込むためのプロトコルの設計方法やネットワークモデルについて述べる。5 章では、実証実験を行いデータの取得、考察について述べる。最後に 6 章では、本報告のまとめと今後の課題について述べる。

2 IP over WDM ネットワーク

2.1 GMPLS

GMPLS (Generalized MPLS) の基となる MPLS (Multi-Protocol Label Switching) について先に説明する。MPLS は IP 網にラベルスイッチの概念を導入することでパスによる網の運用を可能にしたラベルスイッチング方式を用いたパケット転送技術である。現在インターネットで主流となっているルータを用いたバケツリレー式のデータ転送を、より高速・大容量化する技術である。本来、ルータが他のルータから受け取ったパケットを別のルータに転送する際には、ルーティング情報として IP ヘッダを利用するが、MPLS ではこれの代わりに「ラベル」と呼ばれる短い固定長の識別標識を利用する。MPLS 対応ルータ (Label Switching Router : LSR) によって構成されたネットワーク内では、パケットの行き先に応じて次にどのルータに転送するかという情報を各ルータが保持しており、それぞれの経路はラベルによって識別される。このネットワークの入口にあるエッジルータにパケットが届くと、パケット内の経路情報にラベルを付加して、次のルータに転送する。次のルータは、パケットについているラベルを認識し、どのルータに転送すべきかを判断し、適切な転送先にパケットを送る。外部ネットワークへの出口にあるエッジルータは、到着したパケットからラベルを取り除き、外のルータへ転送する。LSR 同士は LDP (Label Distribution Protocol) というプロトコルを用いて経路情報の交換を行ない、経路が変更されるとラベルの再割り当てが行なわれる。このようにラベルをもとにした転送を行なうことにより、転送処理と経路計算処理の分離が可能となり、個々のルータの負担が軽減され、処理の高速化が実現される。GMPLS は、MPLS を一般化し IP 網だけでなく、SDH (Synchronous Digital Hierarchy) のような TDM (Time Division Multiplexing) 網、波長スイッチ網などをはじめとするパス網の運用を自律分散的に行う技術である。GMPLS では、波長に対してラベルをつけ、ラベルによってスイッチングを行うことにより光信号を電気信号に変換することなくスイッチングを可能にする。データチャネルとは別に制御チャネルを設け、制御チャネルを用いて制御メッセージの通信を行い、光パス設定および開放を制御している。この際、制御チャネルの確立および維持を行う LMP (Link Management Protocol) と波長の予約および開放と資源の管理を行う RSVP-TE (ReSerVation Protocol extended for Traffic Engineering) 、ルーティングを行う OSPF-TE (Open Shortest Path First extended for Traffic Engineering) が動作している。

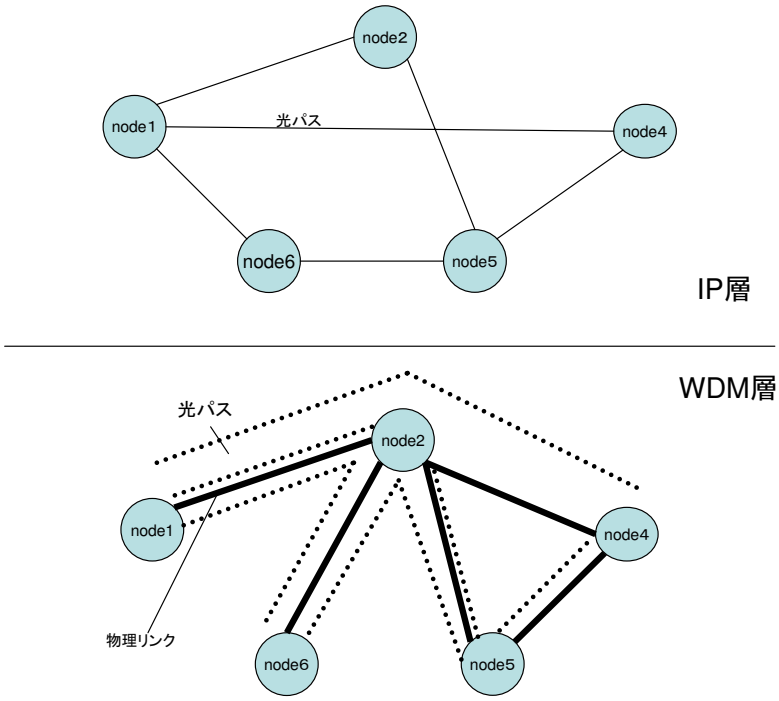

図 1: IP over WDM ネットワーク

2.2 LMP

LMP は主に制御チャネルの管理とリンク属性の相関の確認を行う。制御チャネルの管理とは隣接ノード間の制御チャネルの設定と維持および状態の管理であり、ノード間で一本以上の制御チャネルを必要とする GMPLS では必要不可欠である。制御チャネルの設定は Config メッセージの交換により行われ、Hello メッセージを定期的に交換することにより制御チャネルの維持と状態の管理を行う。一定時間 Hello パケットが受信出来ない場合は、制御チャネルに異常があると判断する。リンク属性の相関の確認は、TE リンクやデータリンクのローカルとリモートの相関を取得することである。MPLS では同一物理インターフェース内部の論理パスを扱っていたのでこの機能は必要なかったが、GMPLS では複数の物理インターフェースを扱うので、隣接ノード双方の各物理インターフェースの相関を確認する必要がある。相関の確認は LinkSummary メッセージの交換と、それに対する LinkSummaryAck/LinkSummaryNack メッセージの返信によって行われる。また取得した相関の確認として Test メッセージの送信とそれに対する TestStatus メッセージの返信が

行われる。また、オプションとしてエラー管理機能が備わっており、エラーの検出が可能である。

2.3 OSPF-TE

現在の IP/MPLS ネットワークの代表的なルーチングプロトコルとして、OSPF (Open Shortest Path First) や IS-IS (Intermediate System - Intermediate System) がある。IS-IS は、大規模ネットワーク向きの内部ゲートウェイ・プロトコルであり、AS（自律システム）間でも使用され、IP 以外のネットワークプロトコルにも対応する。GMPLS ネットワークでの OSPF-TE は、IP ネットワークで用いられた OSPF を拡張している。OSPF-TE はリンクステート型のルーチングプロトコルであり、各ノードがネットワークトポロジに関するデータベースを持ち、そのデータベースに基づいて宛先ノードへの最短経路を計算する。最短経路を算出するために、ダイクストラのアルゴリズムを用いる。１つのノードが同一領域内のネットワークトポロジに関する全ての情報をもつために、各ノードは隣接リンクのリンク状態を調べ、隣接ノードに対してその状態を広告する。各ノードは隣接リンクの状態の変更や、広告された同一領域内のネットワークトポロジに関する情報を受信によりデータベースを更新し、動的に最短経路を導出することができる。図 1 は、WDM 層と IP 層における光パスの概念を示す。図 1 の WDM 層のネットワークにおける実線が光ファイバを使用した WDM 層での物理リンクである。図 1 の WDM 層のネットワークにおける破線は、ノード間の光パスである。図 1 では、合計 6 本の光パスが設定されている。図 1 の WDM 層のネットワークを利用し、光パスによる論理的なネットワークが IP 層に作られる。ノード 1-2 間、1-6 間、2-5 間、5-6 間、1-4 間、4-5 間に光パスが設定されている。例えば、ノード 1 がノード 5 に通信を行う場合は、先述の光パス情報を用いて、経路選択を行うので、ノード 1-2-5 間、1-6-5 間などの経路が考えられる。この節では、GMPLS における OSPF の役割を簡単に述べた。本報告では、統合経路制御手法を実装するために主に OSPF-TE について拡張を施したので、４章で詳しく述べる。

2.4 RSVP-TE

光パスを設定するには、波長変換器などの特殊な装置を用いない限り送受信ノード間の全リンクで同一の波長を用いなければならないという波長連続制約が存在する [5]。また、光パスに利用することができる波長は限られており、何らかの制御を行わなければ複数の光パス設定要求による資源の競合が発生してしまう。そこで、RSVP-TE により送受信ノード間の全リンクで同一波長を予約し、資源の競合を回避する。光パス設定方式には波長予約を送

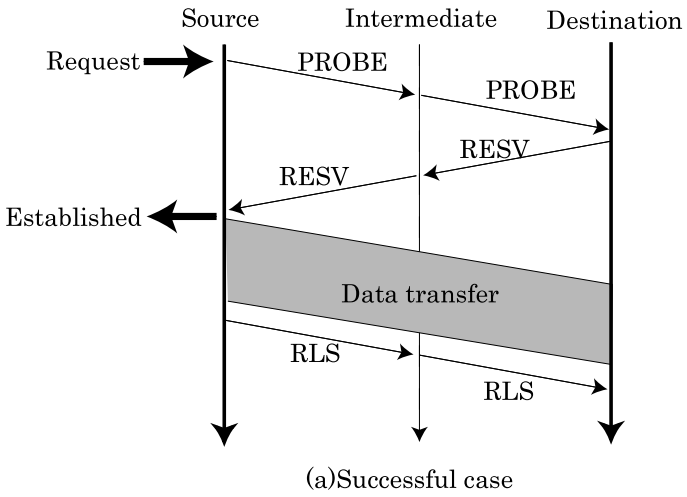

図 2: バックワード型光パス設定方式（パス設定に成功する場合）

信ノードから受信ノードに向けて行うフォワード型光パス設定方式と、受信ノードから送信ノードに向けて行うバックワード型光パス設定方式がある [6]。本報告では、波長の予約時間を短くでき、最新の利用状況に基づいた波長予約が可能であるため、バックワード型光パス設定方式を用いる。

バッグワード型光パス設定方式の制御メッセージのタイミングチャートを図 2、図 3 に示す。バックワード型光パス方式では、データ転送要求の発生時に送信ノードは受信ノードまでの経路を選択する。送信ノードに接続されている経路上のリンクにおいて利用可能な波長を、送信ノードは PROBE 信号として登録する。そして経路にそって PROBE 信号を受信ノードへ送信する。この時、PROBE 信号にはその経路で利用可能な波長情報が格納される。そして、この経路上に存在する中継ノードがこの PROBE 信号を受信する。この時、中継ノードはさらに中継ノード自身のリンク中で利用可能な波長情報を PROBE 信号に反映させる。もし利用不可能な波長が PROBE 信号に登録されていれば削除する。受信ノードが PROBE 信号を受け取ると、PROBE 信号中に格納された利用可能な波長集合の中から１波長を選択する。この選択した波長を用いて送信ノードから受信ノードまで PROBE 信号の経由した経路にそってその波長を予約する。波長予約は RESERVE 信号を用いて行い、受信ノードから送信ノードに向けて行われる。バックワード型光パス設定方式における光パス設定要求の棄却には２種類ある。１つは PROBE 信号により利用可能な波長を調べた結果、利用可能な波長が存在しないと判明した時であり、判明した中間ノードは NACK 信号を送信ノードに対して送信する。もう一つは受信ノードから送信ノードに向けて波長を予約する際に、選択した波長についてすでに他の光パスによって波長予約が行われていた場合であり、その中間ノードは送信ノードに NACK 信号を送ると同時に、受信ノードに対して

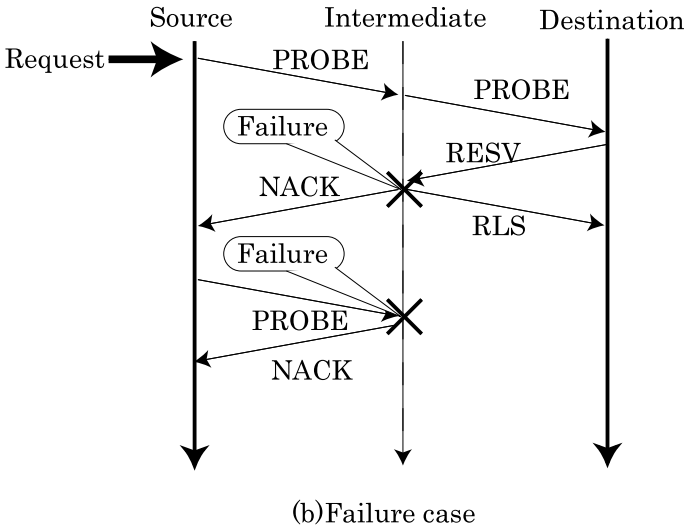

図 3: バックワード型光パス設定方式（パス設定に失敗する場合）

RELEASE 信号を送り、それまで予約した波長を開放する。バックワード型光パス設定方式では、PROBE 信号による利用状況の調査と、実際に他の光パス設定によって波長が使用されている可能性がある。しかし、バックワード型光パス設定方式はフォワード型光パス設定方式に比べて波長の予約時間を短くすることができるため性能が向上し、また、波長の利用状況を動的に収集することから、最新の利用状況に基づいた波長選択が可能になる [7]。

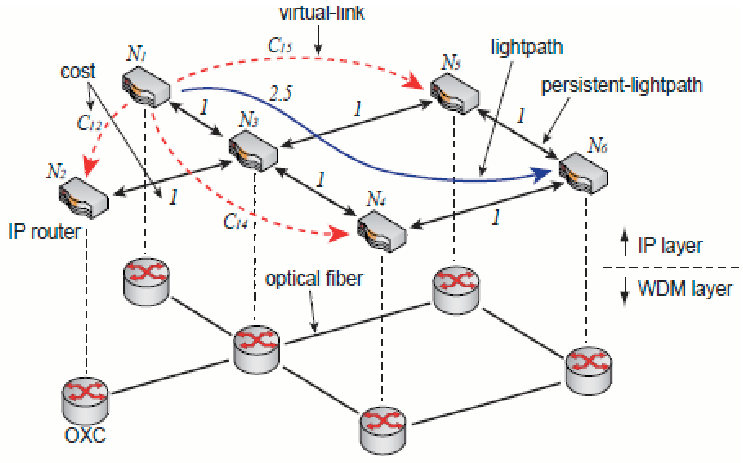

図 4: 仮想リンクを用いた統合経路制御

3　統合経路制御手法

3.1　ネットワークモデル

本報告で想定するネットワークは、光ファイバとノードから構成される。各ノードは、WDM インターフェースを持った IP ルータと、波長ルーティング機能を提供する光クロスコネクト（OXC）で構成される。エンド・エンド間の到達可能性を保証するために、隣接ノード間には一波長利用して静的に光パスを設定する。残りの波長資源は、要求に応じて動的に設定する光パスに利用する。

3.2　仮想リンクの概念

IP の経路制御と WDM の経路制御を統合するために仮想リンクの概念を導入する。仮想リンクは、光パスとして構成されていないが、要求に応じて波長資源を利用することで光パスとして構成できる論理的なリンクである。仮想リンクおよび既存の光パスにはコストを設定し、仮想リンクと IP ネットワークから構成されるトポロジ上で最小コスト経路を探索する。経路計算の結果、仮想リンクが経路として選択されると、波長資源を利用して仮想リンクに光パスを設定する。ここで設定した光パスは、IP の経路制御アルゴリズムによって選

択された経路に含まれるため、IP パケットは確実に光パス上を転送される。仮想リンクを用いた簡単なネットワークの例を図 4 に示す。図 4 は、上層が IP 層、下層が WDM 層となる。ノードは六台使用しており、各ノードは IP ルータと OXC を持つ。WDM 層では、光ファイバを使用して隣接 OXC 間を接続する。IP 層では、隣接ノード間に波長資源を利用し、光パスを設定し、persistent-lightpath と表記している。ルータ R_1 とルータ R_6 間に動的に光パスが設定されており、ルータ R_1 からルータ R_2、R_4、R_5 に三本の仮想リンク (virtual-link) が設定されている。

3.3 経路制御アルゴリズム

仮想リンクを用いた統合経路制御手法のアルゴリズムを以下に示す。

3.3.1 トポロジデータベースの作成

1. 物理トポロジ上の各リンクにおいて、残余波長資源のない物理リンクを削除する。
2. 物理トポロジから最小ホップの経路を選択し、仮想リンクの経路とする。
3. ノードから、ほかの全てのノードに対して仮想リンクを設定する。利用可能な波長資源がなく、光パスを設定できない場合は、仮想リンクは設定しない。
4. 仮想リンクにコストを設定する。

3.3.2 経路選択

上述の手順で作成したトポロジデータベースを用いて経路制御を行う。

1. 仮想リンクと既存の光パスを含めた論理トポロジから、最小コストの経路を選択する。
2. 選択された経路に仮想リンクが含まれていれば、その仮想リンクに対して波長予約を行い、光パスを設定する。
3. 経路選択の結果、IP の経路の一部として利用されなくなった光パスは開放する。

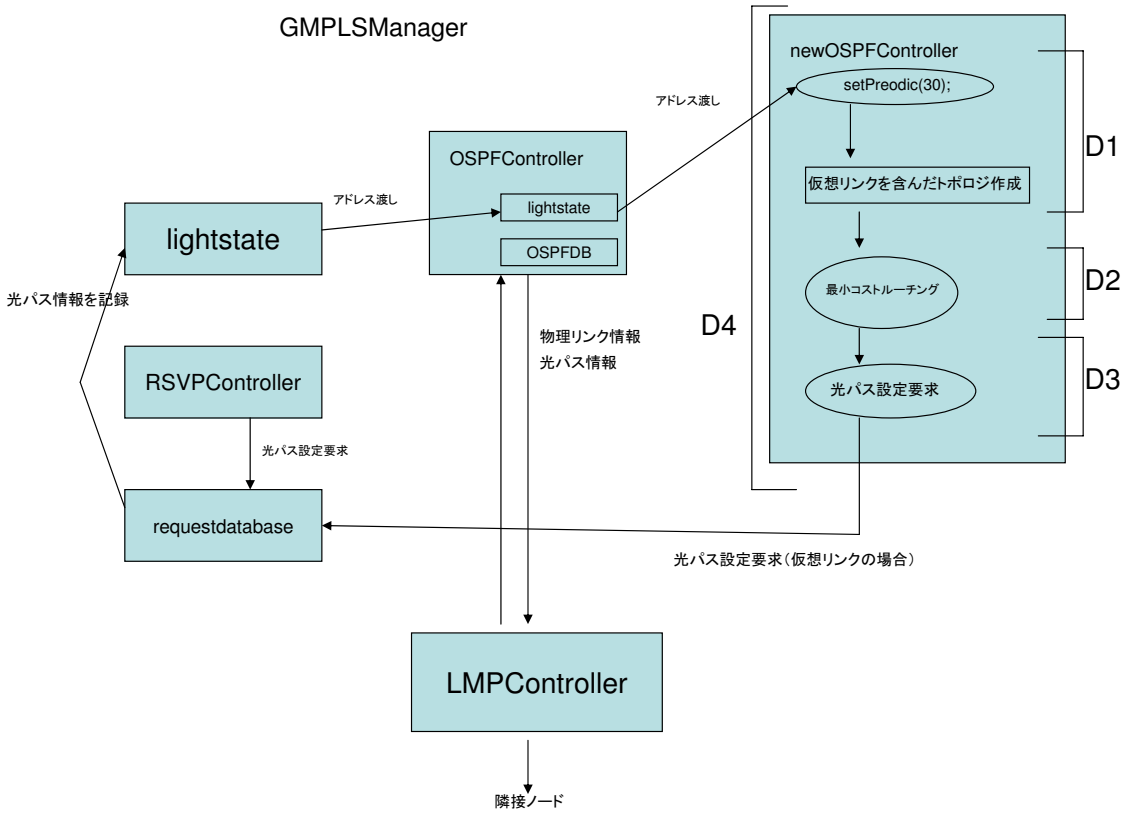

図 5: GMPLS プログラムの全体構成

3.3.3 仮想リンクのコスト

リンクのコスト関数として、主な変数を着ノードの負荷とし、以下に定義する。

$$C_{ij} = v_j^2 + \beta \tag{1}$$

ノード iが発ノード、ノード jが着ノードである。v_j は、ノード jの負荷とし、β は、定数とする。

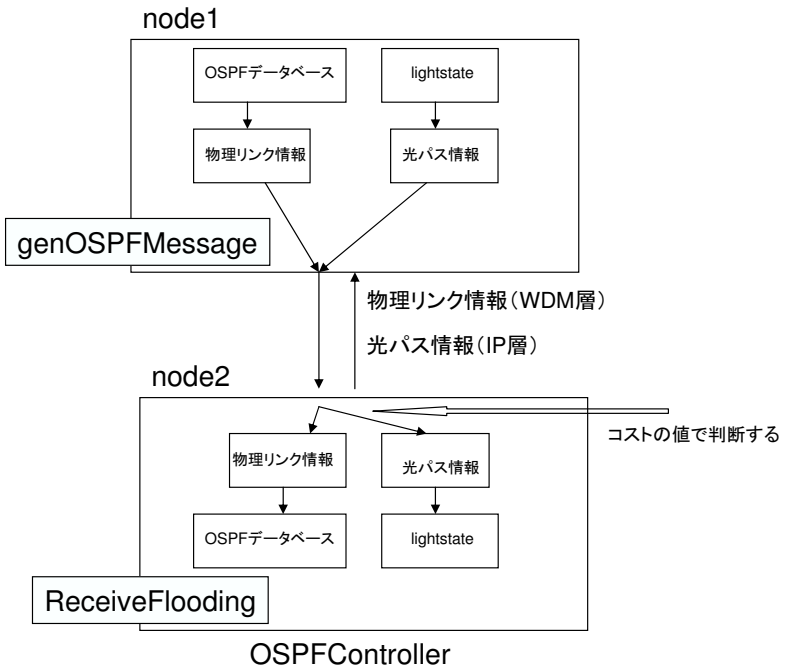

図 6: OSPF-TE の機能拡張：光パス設定情報の配布

4　統合経路制御手法の実装

4.1　プロトコル設計

本報告では、主に GMPLS プログラムの OSPF-TE に拡張を施す。実際の拡張内容については、4.2 節で詳細を記すので、ここでは概略を述べる。本報告の統合経路制御手法を実装するために GMPLS プログラムとは別に新しく導入したモジュールである newOSPFController がある。newOSPFController は、必要な光パス情報を RSVP-TE から取得する。隣接ノード間との情報交換により、全体のネットワークトポロジを把握することが可能である OSPFController から隣接ノードに光パス情報を広告、収集し、IP 層のトポロジを作成する。作成した IP 層のトポロジに仮想リンクを加える。送信ノードから受信ノードまでのコストが最小となる経路を仮想リンクを加えたトポロジを用いて決定する。決定した経路が仮想リンクである場合、RSVP-TE に光パス設定要求を行う。決定した経路が仮想リンクでない場合、何も行わない。

4.2 OSPF-TE の機能拡張

図 5 に統合経路制御手法の実装における拡張内容の概要を示す。IP 層のネットワークトポロジ情報のデータベースにあたる lightstate モジュールを以下のように作成する。

- IP 層の光パス情報は、Lightpath モジュールとして動作する。
- Lightpath モジュールには、送信ノード、受信ノード 、コスト、仮想リンクであるかどうか 、の四つの光パス情報を保持する。
- 送信ノードは、Lightpath モジュールのメンバ変数 srcnodeid として格納する。
- 受信ノードは、Lightpath モジュールのメンバ変数 dstnodeid として格納する。
- コストは、Lightpath モジュールのメンバ変数 metric として格納する。
- 仮想リンクであるかどうかは、Lightpath モジュールのメンバ変数 virtuallink が 1 の場合、仮想リンクであり、 0 の場合、仮想リンクでないとする。
- 二次元配列を用いて、lightstate モジュールに光パス情報である Lightpath モジュールを格納する。

IP 層のネットワークトポロジにおける光パス情報を取得するために、光パス設定を行う RSVP-TE において送信ノード、受信ノード間の光パス情報を lightstate に加える。具体的には、setlightstate 関数を RSVP モジュール内の波長予約を行う establishpath 関数内で使用し、自ノードからの光パス設定情報を lightstate に加える。この段階では、自ノードのみの光パス情報しか認識できないので、IP 層全体のネットワークトポロジを把握出来ない。そこで、GMPLS プログラムの OSPF-TE モジュールを利用し、光パス情報の配布を行う。

OSPF-TE は、WDM 層のネットワークトポロジを構成する　物理リンクの情報を隣接ノード間で交換し、ネットワークトポロジを把握することが出来るので、物理リンクの配布情報の中に光パス情報を組み込むことにより、光パス情報を広告、収集できる。OSPF-TE モジュールにおいて、物理リンクは、Linkinformation モジュールとして動作する。Linkinformation には以下の情報を保持する。

- srcnodeid、dstnodeid　物理リンクの両側のノード番号を格納する。
- metric　物理リンクのコストを格納する。
- lambdas　物理リンクの波長数を格納する。

- availables　物理リンクに使用されていない波長帯があれば１、全ての波長が使用されていて光パス設定のために物理リンクが利用できない場合は０を格納する。

光パス情報を OSPF-TE モジュールを利用し、物理リンクと同様の処理を行い、配布・広告することにより IP 層のネットワークトポロジを把握する。広告する際、光パス情報を物理リンクの Linkinformation に組み込む。また、広告を受信する際、物理リンクと光パス情報を区別する。そのために、genOSPFMessage 関数、ReceiveFlooding 関数に拡張を施す。図 6 に概要を示す。

genOSPFMessage 関数では、自ノードのもつ物理リンク情報を OSPF メッセージに格納する。その際に自ノードの光パス情報のコストを特定の値（本報告では 777）にして、OSPF メッセージに物理リンク情報と共に格納する。

ReceiveFlooding 関数では、隣接ノードから広告された情報を受信する。受信する際に、コスト値が 777 以外の場合は物理リンクとして処理をする。コスト値が 777 の場合は光パス情報として処理を行い、lightstate モジュールに格納する。この際、新たにコスト値を決定する必要があるが、本報告の統合経路制御手法では、実験の都合上 10 として格納する。また、拡張を加えた OSPF-TE 内の genOSPFMessgae 関数、ReceiveFlooding 関数においては、5.2 節で実行時間を計測し考察を行う。

以上で収集した光パス情報が、IP 層の光パスで形成されたネットワークトポロジである。

newOSPFController モジュールは上述した lightstate モジュールを利用し、以下の実装を行う。newOSPFController では、IP 層のネットワークトポロジを管理し、仮想リンクを含んだトポロジを作成する。lightstate モジュールの setlightstate 関数を用いて仮想リンクを lightstate モジュールに組み込むことが出来る。setlightstate 関数の引数は、送信ノード、受信ノード、コスト、仮想リンクであるかどうかの 4 つの情報である。

仮想リンクを含んだトポロジを用いて、光パスを設定する発信ノード、受信ノード間のコストが最小である経路をダイクストラ法を用いて算出する。ダイクストラのアルゴリズムに与えるデータとして、各ノード間が隣接している場合は、光パスのコスト 10 とし、隣接していない場合を 100000 として最小コストを計算する。算出した経路が、2 ホップ以上の場合は仮想リンクでないので、光パス設定要求を送信しない。算出した経路が 1 ホップである場合は仮想リンクであるので、RSVP モジュールに光パス設定要求を送信する。ただし、1 ホップの仮想リンクであったとしても、仮想リンクの送信ノード、受信ノード間に WDM 層の波長資源が必要である。WDM 層での最小ホップ数である経路を計算し、その経路上に光パス設定要求を行う。

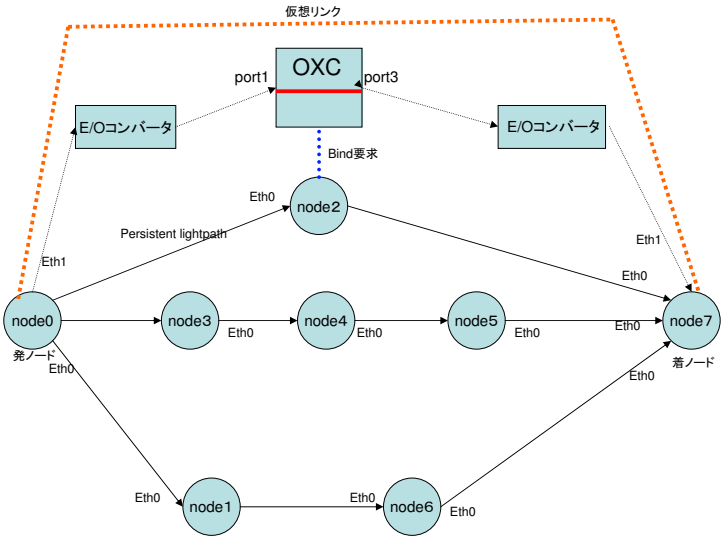

図 7: ネットワークトポロジ（8 ノード）

5　実証実験

5.1　実験環境

実験で用いるネットワーク構成は、図 7 に示す。図 8、図 9、図 10 は、実験環境の写真である。図 7 の OXC (Optical Cross Connect) は、光スイッチング機能を動作させる PC である。E/O コンバータは、電気を光に変換する機能を持つスイッチングハブである。

図 11 は、スイッチングハブを用いてノード 8 が 8 台の PC を統括している図である。それぞれの PC の eth0 のインターフェースをスイッチングハブと接続する。ノード 8 が、ssh プログラムを使用して、全ノードにログインする。ノード 8 の端末上で全ての GMPLS プログラムを動作させる。表 1 は、ネットワーク全体のノードと OXC の IP アドレス対応表を示す。

実験の目的は、以下に示す。

1. OXC による光パス設定の確認
2. 仮想リンクを設定し状況に応じて仮想リンクを光パスとして動作することの確認
   - 発ノード、着ノード、コスト、仮想リンクであるか、の 4 つの情報を持つ光パス情報を発ノードが収集し、仮想リンクを設定し状況に応じて仮想リンクが光パス

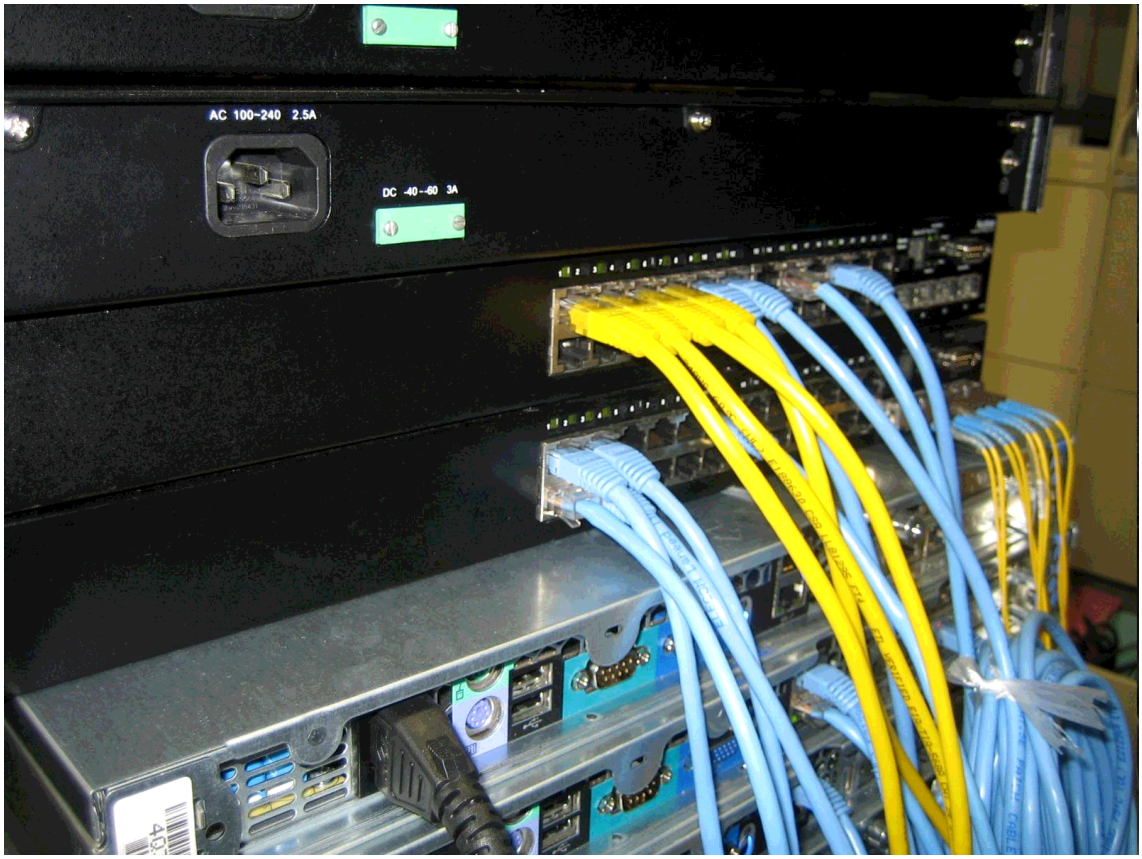

図 8: 実験環境: E/O コンバータ

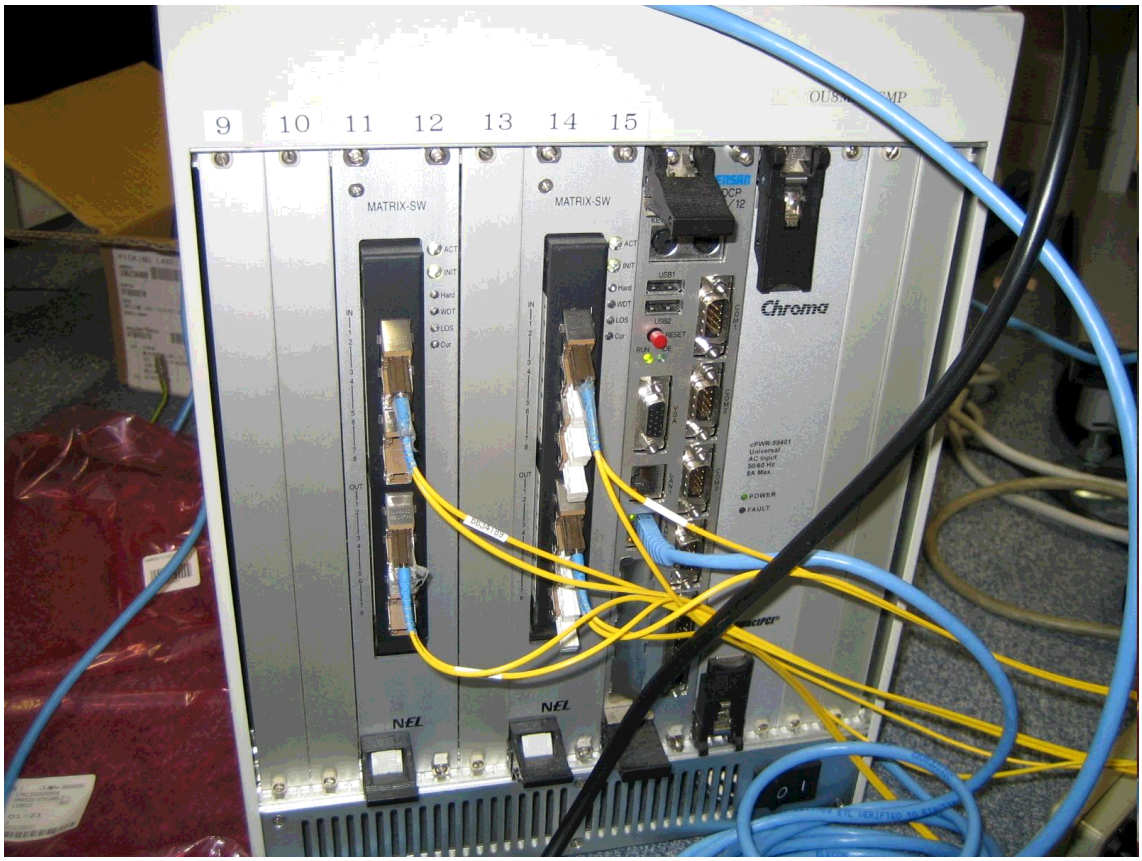

図 9: 実験環境: OXC の接続

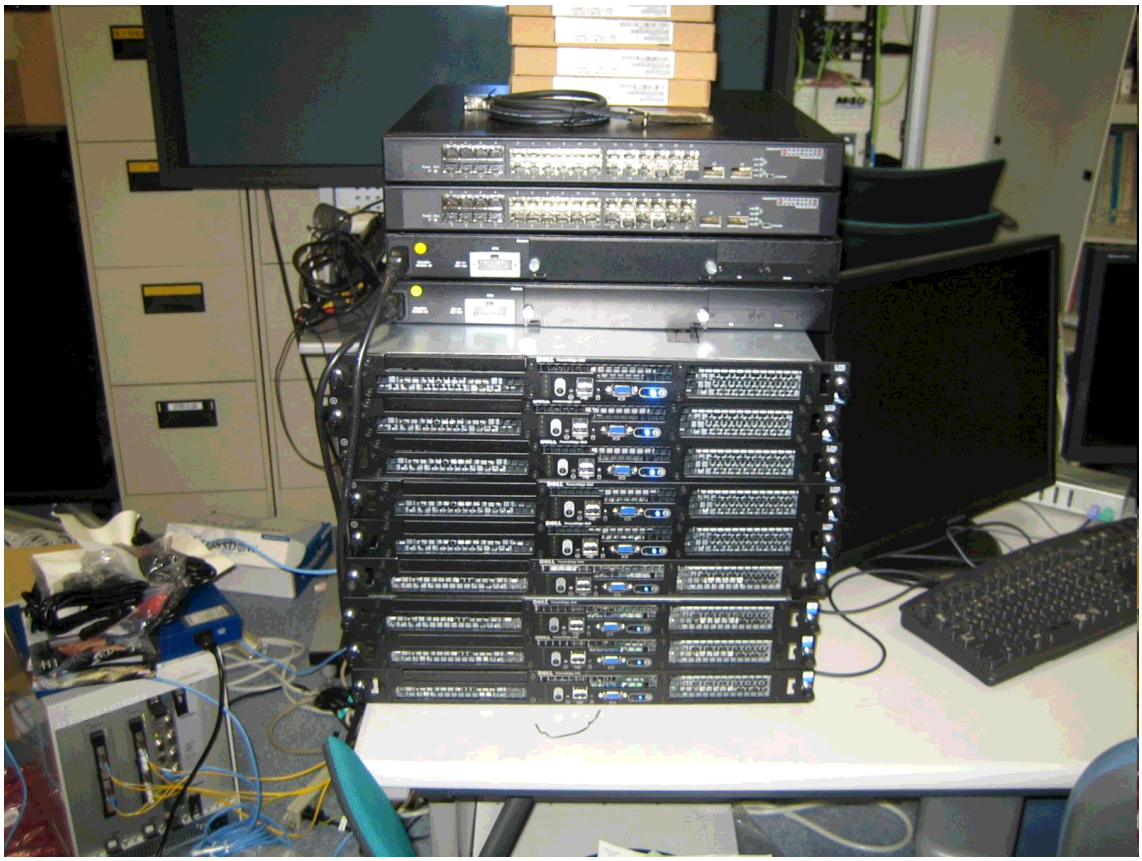

図 10: 実験環境: 制御コントローラ用 PC

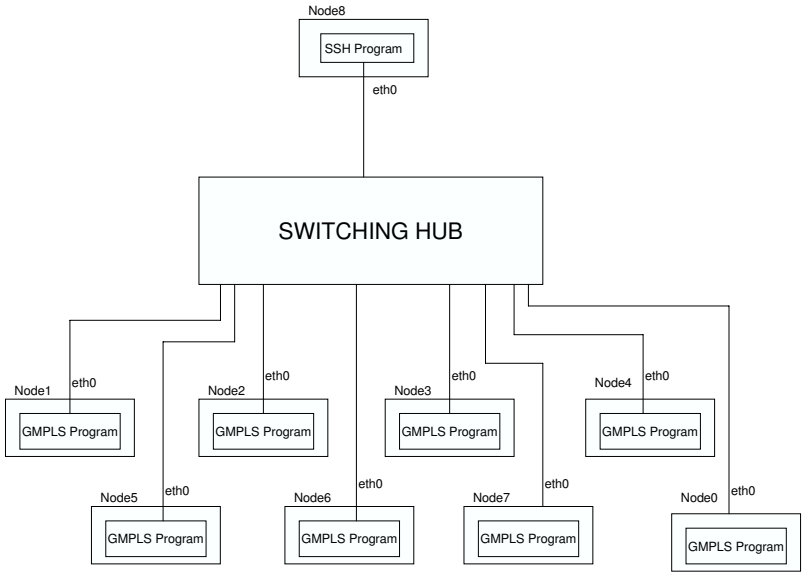

図 11: 実験ログ収集機構

として動作することを確認する。

3. 統合経路制御手法の実装部の実行時間の計測

4. ネットワークスループットの計測

GMPLS 制御部の遅延の計測は、5.2 節に記す。ネットワークスループットの計測は 5.3 節に記す。発ノードから着ノードまでの経路情報を仮想リンクも含めた有向グラフとして表す。発ノードから着ノードまでの経路を最小コストルーティング（dijkstra 法）を用いて計算する。そして、仮想リンクが経路となる場合、ノード 2 が OXC に光パス設定要求を送信し、仮想リンクが光パスとなるので、IP 層の光パスによるネットワークトポロジが変化する。仮想リンクが経路とならない場合、IP 層のネットワークにおいて光パスは新たに設定されず、ネットワークトポロジに変化は無い。

光パス設定の確認を以下のように行う。OXC では、入力ポート、出力ポートの bind,unbind 設定が可能である。図 7 では、PORT1 と PORT3 が bind されている状態である。ノード 0 からノード 7 に ping をノード 0 の eth1 を通して送信する。ping が通過する経路は、ノード 0、E/O コンバータ、OXC、E/O コンバータ、ノード 7 の順である。OXC は、ノード 2 から光パス設定要求を受けると、入力 PORT、出力 PORT を bind し、ping がノード 0 の eth1 からノード 7 の eth1 に到達する。そして、ノード 7 からの ACK が同じ経路を逆に通

表 1: IP アドレス対応表

ノード	IP アドレス
node0	192.168.1.100
node1	192.168.1.101
node2	192.168.1.102
node3	192.168.1.103
node4	192.168.1.104
node5	192.168.1.105
node6	192.168.1.106
node7	192.168.1.107
node8	192.168.1.108
OXC	192.168.1.50

表 2: node0 の経路情報

インタフェース	IP アドレス
eth0	192.168.1.100
eth0	192.168.1.101
eth0	192.168.1.102
eth0	192.168.1.103
eth0	192.168.1.108
eth1	192.168.1.70

過しノード 0 の eth1 に返り、光パスが設定されたと確認出来る。インタフェース指定することで、ノード 0、ノード 1、ノード 6, ノード 7 のような他の経路から ping が通過することを防いでいる。簡単のため、図 7 では、片方向しか接続していないが、実際は双方向に接続している。発ノードがネットワーク全体の光パス情報を収集しているかどうかを以下のように確認する。図 7 と同じネットワークトポロジを実際に作成するために、各ノードが光パス設定機能を持つ RSVP を用いて光パス設定を行う。各ノードがネットワークトポロジを把握するためのリンクステート型プロトコルである OSPF-TE を利用し、情報交換を行う。今回は、最大ホップ数が 4 であるので、4 回の情報交換を行い、ネットワーク全体の光パス情報を収集する。その後、発ノードの光パス情報を管理する lightstate モジュールが光パス情報を収集できているかを確認する。最小コストルーティングが正しく行われているかの確認は、図 7 のネットワークトポロジで、ノード 0 からノード 7 までの経路の内、最小

表 3: node2 の経路情報

インタフェース	IP アドレス
eth0	192.168.1.100
eth0	192.168.1.102
eth0	192.168.1.107
eth0	192.168.1.108
eth2	192.168.1.50 (OXC)

表 4: node7 の経路情報

インタフェース	IP アドレス
eth0	192.168.1.102
eth0	192.168.1.105
eth0	192.168.1.106
eth0	192.168.1.108
eth1	192.168.1.60

表 5: node8 の経路情報

インタフェース	IP アドレス
eth0	192.168.1.100
eth0	192.168.1.101
eth0	192.168.1.102
eth0	192.168.1.103
eth0	192.168.1.104
eth0	192.168.1.105
eth0	192.168.1.106
eth0	192.168.1.107
eth0	192.168.1.108

表 6: newOSPF モジュールの処理遅延： 8 ノード、仮想リンクを選択する場合

試行回数	D1(us)	D2(us)	D3(us)	D4(us)
1	2	20	35	79
2	12	21	29	80
3	2	208	95	514
4	2	212	97	520
5	2	18	35	76
6	2	223	101	542
7	2	33	37	99
8	2	225	96	551
9	3	32	30	84
10	2	27	32	80

のコストを持つ経路が算出されることを端末で表示して行う。仮想リンクが光パスとして動作することを確認するために以下を行う。仮想リンクのコストを 10 とした場合、ノード 0 →ノード 7 がコスト 10 となり仮想リンクを通るコストが最小の経路となる。この場合、仮想リンクを通るコストが最小の経路となるので ping の返答があり、光パスが設定されたと確認できる。仮想リンクのコストを 50 とした場合、ノード 0 →ノード 2 →ノード 7 がコスト 20 となりコストが最小の経路となる。この場合、仮想リンクがコストが最小の経路にならないので ping の返答はなく、光パスが設定されないと確認できる。

図 11 は、ノード 8 が他のすべてのノードをスイッチングハブを介して管理する。図 11 の状態であると、eth0 を通じて全てのノードと接続することが出来るので、図 7 に示すネットワークトポロジを構築しているとはいえない。なぜなら、ノード 0 からノード 4 に eth0 を通して通信が出来る。そのために、route コマンドを用いて必要でない経路を遮断する。表 2、表 3、表 4、表 5 にノードの経路情報を記した。なお、他のノードについても同様の経路情報を保持させている。

このようにすることで、図 7 のノード 0 からノード 5 に対して eth0 で直接接続出来ない。ノード 0 からノード 5 に ping を送信することで直接接続出来ないことを確認した。

5.2 制御モジュールにおける処理遅延

実験におけるネットワークは、ノード 8 台の場合、図 7 のトポロジを使用する。ノード 3 台の場合は、図 12 のトポロジを使用する。また、表 6,7,8,9 の各モジュールの実行時間は、ノード 0 で動作する newOSPFController モジュールを組み込んだ GMPLS プログラムにお

表 7: newOSPF モジュールの処理遅延： 8 ノード、仮想リンクを選択しない場合

試行回数	D1(us)	D2(us)	D3(us)	D4(us)
1	2	281	0	384
2	1	239	0	386
3	1	229	0	337
4	1	232	0	386
5	1	218	0	366
6	1	34	0	51
7	1	230	0	390
8	1	224	0	370
9	1	239	0	386
10	1	23	0	39

表 8: newOSPF モジュールの処理遅延： 3 ノード、仮想リンクを選択する場合

試行回数	D1(us)	D2(us)	D3(us)	D4(us)
1	1	251	129	641
2	1	19	48	88
3	1	19	53	92
4	1	17	54	91
5	0	18	52	89
6	0	17	53	92
7	1	20	50	91
8	1	247	124	622
9	1	249	125	632
10	1	252	137	647

表 9: newOSPF モジュールの処理遅延： 3 ノード、仮想リンクを選択しない場合

試行回数	D1(us)	D2(us)	D3(us)	D4(us)
1	0	248	0	416
2	1	22	0	39
3	1	248	0	417
4	1	19	0	34
5	0	19	0	34
6	0	250	0	419
7	1	18	0	33
8	0	18	0	32
9	1	17	0	32
10	0	249	0	422

表 10: ノード 8 台による OSPF モジュールの実行時間

試行回数	genOSPFMessage(us)	ReceiveFlooding(us)
1	116	13
2	17	14
3	104	14
4	58	14
5	147	13
6	78	15
7	154	11
8	17	14
9	72	14
10	18	13

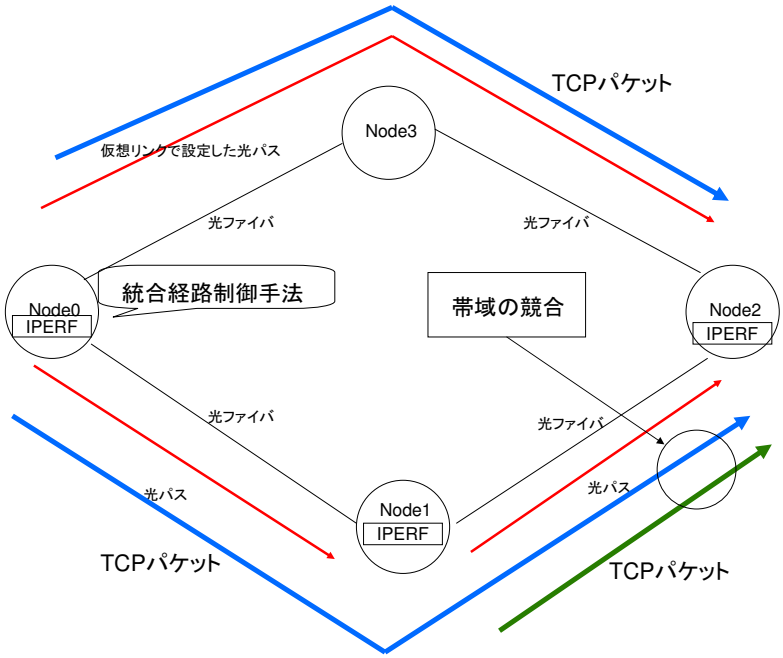

図 12: ネットワークトポロジ (3 ノード)

ける実行時間である。実行時間の計測には、マイクロ秒の精度を持ち、I/O 等も含めたプログラム全体の測定を行うので、gettimeofday を利用した。

表 6、表 7 は、8 ノードのネットワークトポロジにおいて統合経路制御手法における実装部である newOSPF モジュールの実行時間の計測結果を示す。D1 は、図 5 に示すように、newOSPFController が光パス情報を確認して、仮想リンクを含んだトポロジを作成するまでの時間である。D2 は、コストが最小の経路を決めるまでの実行時間である。D3 は、D2 の間の制御において、コストが最小の経路を決定し、その経路が仮想リンクである場合、光パス設定要求を送信するまでの実行時間である。D4 は、newOSPF モジュールが呼び出されて、実行開始から光パス設定を RSVP モジュールに要求するまでの実行時間である。試行回数を 10 回とした。

表 6 の D1、D2、D3、D4 は試行する毎に多少ばらつきがあるが、全て 550us 以下の値を示しており、大きな制御遅延は無く、仮想リンクが選ばれた場合の統合経路制御手法が正しく動作していると確認できる。

表 7 の D1、D2、D4 は、表 6 とほとんど差異は無いが、D3 は 0 である。理由として、経路選択時に仮想リンクが選ばれない場合は、光パス設定要求を送信しないことが挙げられる。このことから光パス設定要求を送信しない場合の制御が正しく動作していることが表 7 から確認できる。

newOSPFController において、図 5 に示すように、

$$D4 = D1 + D2 + D3 \tag{2}$$

となるはずであるが、IF 文の実行時間、CPU のメモリ割り当 てなどからマイクロ秒単位の誤差が生じているので、多少 D4 が大きな値となる。

表 8、表 9 は、3 ノードのネットワークトポロジにおいて統合経路制御手法における実装部である newOSPF モジュールの実行時間の計測結果を示す。表 8、表 9 は、D2 の部分で多少 8 ノードの場合より低い値が得られると予測できる。理由として、D2 の部分はコストが最小である経路を算出するダイクストラ法の実装部において、計算量がノードの二乗に比例するからである。しかし、実測値にはあまり違いがみられないので、8 ノードと 3 ノード程度の差では、大きな影響がないことが確認できる。

表 10 は、GMPLS プログラムにおける OSPF-TE モジュールの genOSPFMessage、ReceiveFlooding の実行時間である。genOSPFMessage、ReceiveFlooding 関数は、統合経路制御手法を実装する上で光パス情報を広告するために拡張を加えた部分であるので実行時間を計測した。genOSPFMessagae 関数では、OSPFController が保持する物理リンクのデータベースを隣接ノードに配布するためのメッセージを作成する。その際に、光パス情報もメッセージに加える。ReceiveFlooding 関数は、隣接ノードから送信された情報を受信し、リンク情報ならば、リンク情報のデータベースに登録し、光パス情報ならば、光パスのデータベース (lightstate) に登録する制御を行う。

genOSPFMessage 関数の実行は全て 200us 以下で行われている。ReceiveFlooding の実行は全て 20us 以下で行われている。表 10 では、試行を繰り返してもほとんどばらつきが無い。以上から統合経路制御手法のために拡張を加えた OSPF-TE モジュールにおける genOSPFMessage、ReceiveFlooding の実装部は正しく動作することが確認できる。

5.3 ネットワークスループットの計測

図 13 に示すのは、スループット計測のためのデータプレーンである。3 ノードのネットワークにおいてデータを通信させる。ノード間のスループットを計測するために、node0、node1 において IPERF プログラムを CLIENT として動作させる [8]。node2 において IPERF プログラムを SERVER として動作させる。初期設定で OXC は、node0 と node1、node1 と node2 側のポートを接続する。node0 と node2 間の通信は、node1 を経由して 2 ホップで行う。経路は、node0 の eth1、node1 の eth1、node1 の eth2、node2 の eth1 である。また、node1 と node2 間の通信は、１ホップで行う。経路は、node1 の eth2、node2 の eth1

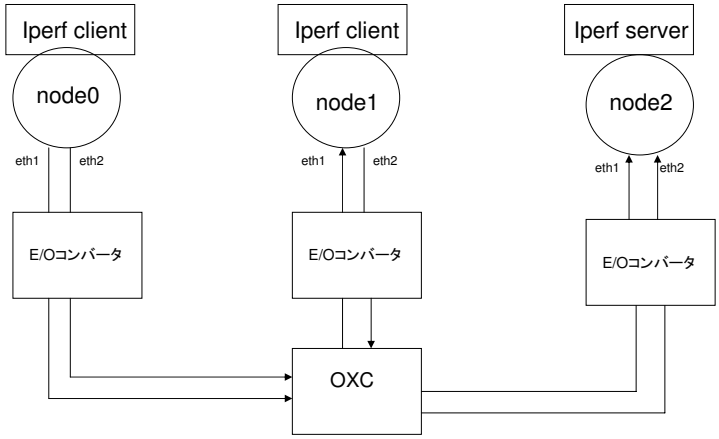

図 13: スループット計測時のデータプレーン

である。この場合、どちらの通信も node1 の eth2 と node2 の eth1 間で同じ帯域を使用する。node0 と node2 間に光パスが設定されると、node0 の eth2、node2 の eth2 という経路で通信が行われる。

実験のシナリオを以下に示す。

1. node0 が IPERF を実行し TCP パケットを node2 にむけて送信する。
2. node1 が IPERF を実行し TCP パケットを node2 にむけて送信する。
3. 統合経路制御手法を用いて、node0 と node2 間における 1 ホップの仮想リンクを光パスとして設定する。
4. 仮想リンクを選択すると、node2 宛てのパケットを node0 の eth1 から node1 の eth1 に送信するという route 情報を node0 の eth2 から node2 の eth2 に変更することで node0 と node2 間の 1 ホップの通信を行う。

図 14 が、スループット計測結果である。シナリオ 1 (0 秒〜6 秒) は、node0 と node1 と node2 間において、1Gbps/sec に近いスループットを計測している。シナリオ 2 (6 秒〜16 秒) は、TCP の輻輳制御が動作し、node0 と node1 と node2 間及び node1 と node2 間を流れるトラヒックは 500Mbps/sec となる。これは、node1 の eth2 と node2 の eth1 間で帯域の競合が

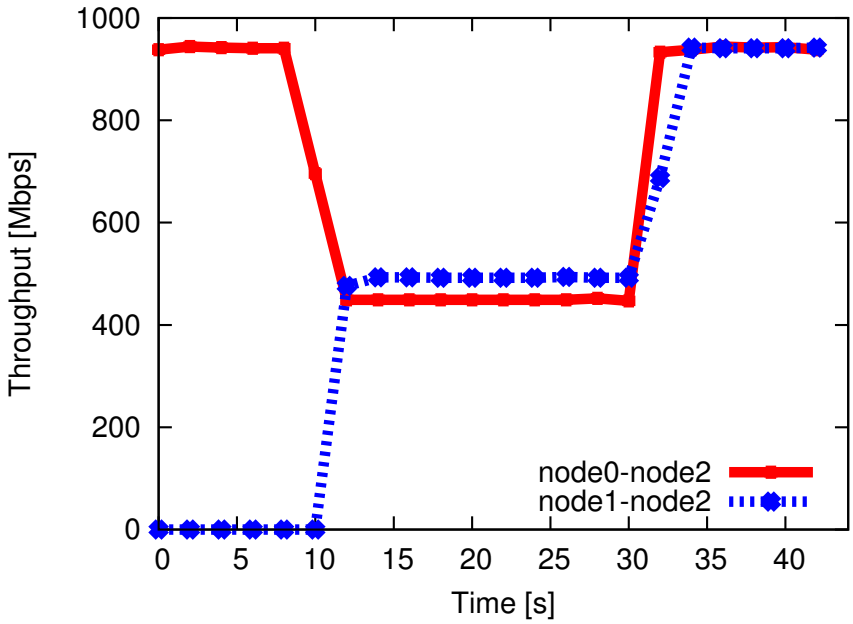

図 14: スループット計測

起きているからである。シナリオ 3 では、統合経路制御手法を用いて、仮想リンクを選択している。ここでは、簡単のため、仮想リンクに非常に低いコストを設定し、確実に仮想リンクがコストが最小の経路となり、光パスになるようにした。シナリオ 3 は (17 秒〜)、node0 と node1 と node2 間に流れていた TCP パケットは node0 と node2 間に流れるので、node1 と node2 の間を流れるトラヒックとの競合は無くなり、どちらも 1Gbps/sec に近いスループットを計測している。このことから、統合経路制御手法によるスループットの改善の一例を実環境において示すことができた。

6　おわりに

本報告では、IP over WDM における統合経路制御手法の実装と評価を行った。実験の結果から統合経路制御手法が実環境においても正しく動作することを確認した。実行時間においても非常に短い時間で統合経路制御手法が動作していることが確認できた。データ転送実験において、統合経路制御手法によりネットワークスループット改善の一例を実環境において示すことが出来た。今後は、動的に仮想リンクを設定し、仮想リンクが選択されない場合も含めて統合経路制御手法の有効性を示す予定である。

謝辞

本報告を終えるにあたりまして、ご指導、ご教授を賜わりました大阪大学大学院情報科学研究科　村田正幸教授に厚く御礼申し上げます。

また本報告の作成に熱心にご指導、御助言を賜りました大阪大学大学院情報科学研究科　荒川伸一助教に深く感謝致します。

並びに適切な御助言を賜りました大阪大学大学院情報科学研究科　若宮直紀准教授に心より感謝致します。

最後に日頃から本報告の作成にあたり様々な質問に答えて頂きました村田研究室の小泉佑揮氏、大橋正稔氏をはじめとする村田研究室の皆様方に心より御礼申し上げます。

参考文献

[1] M. Kodialanm and T. V. Lakshman, "Intergrated dynamic IP and wavelength routing in IP over WDM networks," in *Proceedings of IEEE INFOCOM 2001*, pp. 358–366, Apr 2001.

[2] J. Comellas, R. Martinez, J. Prat, V. Sales, and G. Junyent, "Intergrated IP/WDM routing in GMPLS-based optical networks," *IEEE Network Magazine*, vol. 17, pp. 22–27, Mar/Apr 2003.

[3] R. Dutta and G. N. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks," *Optical Network Magazine*, vol. 1, pp. 73–89, Jan 2000.

[4] Y. Koizumi, S. Arakawa, and M. Murata, "An intergrated routing mechanism for class-layer traffic engineering in IP over WDM networks," *IEICE TRANSACTIONS on Communications*, vol. 90, pp. 1142–1151, May 2007.

[5] H. Zhang, J. P. Jue, and B. Mukherjee, "A review of routing and wavelength assignment approaches for wavelength routed optical WDM networks," *Optical Networks*, vol. 1, no. 1, pp. 47–59, 2000.

[6] C. S. R. Murthy and M. Gurusamy, *WDM OPTICAL NETWORKS*. PRENTICE HALL, 2002.

[7] M. Ohashi, S. Arakawa, and M. Murata, "Implementation and evaluation of fast light-path setup method in wavelength-routed WDM networks," in *Proceedings of SPIE APOC 2006*, vol. 6354, pp. 63541V–1 – 63541V–9, Sep 2006.

[8] "Iperf – TCP/UDP bandwidth measurement tool," availables at `http://dast.nlanr.net/Projects/Iperf/`.