

# 複数の迂回パスを考慮したトラフィック推定に基づく 帯域設計法

M2011MM056 大野祐史

指導教員：奥村康行

## 1 はじめに

近年、ネットワーク資源を効率的に利用することができるような、柔軟性と信頼性を両立させるネットワークが求められている。

通信網における発着ノード（以下 Source-Destination[SD]）間を流れる交流トラフィックは、網設計において有益な情報である。しかし、交流トラフィックの取得は、装置に対する負荷を要するものであるため、どのような形の通信網においても実施されない。

先行研究 [1] では、交流トラフィックが取得できない状況下で、リンクにおいて取得するトラフィック情報から現用パス 1 本に対し、迂回パス 1 本の必要帯域を導出する方法を線形計画法によるアプローチを採っている。

また、問題を定式化し、小サイズの単純なメッシュ網に対して提案法を適用し、実施した適用性評価の結果、考察を示している。本研究では、柔軟性と信頼性をより高めるために現用パス 1 本に対して複数本の迂回パスを考慮したノード数サイズの大きい、より現実的なメッシュ網に対して、提案法を適用、実施した適用性評価の結果、考察を示す。

## 2 複数の迂回パスの網設計と課題

光パス網の設計と課題とその解決策について説明する。

### 2.1 複数の迂回パスの特性 [2]

本研究ではリンク障害が発生した際、複数の迂回パスを用いる。これらのもたらすメリットは、まず初めに帯域コストの削減がある。迂回パスを通るリンク毎の帯域を分散させ効率よくパケットを送ることができる。また、仮に迂回パス中のリンクに障害が発生した時も、複数本迂回パスがある為、接続が途切れることなくある程度の帯域を確保できる。しかし、リンク間の長さを全て 1 としてパスの設定を最短経路としているので、ルートの探索作業時間が増えてしまう。そのような課題はあるが、本研究ではあくまで帯域に注視し帯域コストのみを考える。

### 2.2 光パス・光パス網の設計と課題

ネットワークは、ノード間をつなぐリンクで構成されている。このリンク内には、始点、終点となる光パスと呼ばれる仮想的なパケットの通り道が設定され、各パケットはそれに沿って転送されることになる。数多くの光パスの中で、さまざまな制約条件を考慮しつつ、特定のリンク障害を考慮した上で、ネットワークリソースを設計することを光パス設計と呼ぶ。

光パス網を設計するにあたり、ノード負荷を考慮し交流トラフィックの情報がまったく取得できない状況で、リンク

において必要帯域を求める必要がある。障害が発生せず現用パスが選択されている際に所得したリンクでのトラフィックにより、現用パスに対し、必要な帯域を設計することは可能である。

しかし、完全な交流トラフィック情報を持たない場合、迂回パスに対し、必要なリンクでの帯域を求めることができない。

### 2.3 解決方針

この課題の解決策として、本研究では、線形計画法による定式化を用いる [1]。先行研究 [7] では、リンクで取得するトラフィックから、現用パス、及び、迂回パス容量を合わせた必要帯域の上限値を求める問題を線形計画法として定式化し、この問題を解くことで必要帯域を求める方法が提案された。そして、構成するノード数の異なる小サイズのメッシュ網に対して帯域設計法を適用し、必要帯域の平均値、及び、その 95% の信頼区間を求めていた。それらを評価する事で、小サイズなノードから構成されるメッシュ網における帯域設計においては効果が期待できることを示している。しかしながら、障害にも対応する柔軟性のあるネットワークで、提案法を用い帯域を改善する点を示していない。

そこで、本研究では、4. で示した帯域設計法を更に複数本の迂回パスを考慮した上で定式化し、より現実的なメッシュ網に対し適用する方法を提案する。メッシュ網では、構成する全ノード間を SD とするトラフィックが流れるものとした。ここで、交流トラフィックを除く全ての情報は帯域設計の際に既知として扱うこととする。なお、メッシュ網においては任意に決めた現用パスに対し、複数本の迂回パスは現用パスとは全て異なるパスを通るように決める。

また、他の解決策として、Network Tomography[3][5] を用いて、外部からパケットを送り、内部のトポロジーを推定する。その結果から得た交流トラフィックの推定結果を用いて必要帯域を導出するという方法が考えられる。しかし、設計においては、帯域不足を発生する危険を除外することが指摘されることから、確実な必要帯域を求める方法としては、前者の方法が望ましいと考える。

## 3 帯域設計法の前提条件

前提条件として、以下の項目を規定する。

- ・全 SD 間の現用パスとその迂回パスの経路は予め決定されている。
- ・トラフィック取得時においては、現用パスが利用されている。
- ・障害発生時には、障害リンクを有する現用パスは迂回パスに切り替えられる。
- ・単一リンクの障害（複数リンク同時障害は想定しない）

に対して、迂回パスによる復旧が可能なようにする。

・各リンクの各方向に対して、流れるトラヒックの和、各ノード上の流入トラヒックの和と流出トラヒックの和が、トラヒック情報として取得されている。

#### 4 線形計画法による定式化 [6]

前提条件のもとに、現用パスと迂回パスを疎通するのに必要な帯域を算出する帯域設計法について説明する。

なお、この帯域設計法は、迂回パスで利用する帯域が切り替え要因となる障害リンクを異にする迂回パス間で共有することを前提とする Shared Protection 方式を例に定式化について記述する。

基本的な考え方は、次のようなものである。まず、未知のSD間の交流トラヒック量を変数とし、リンク上を流れるトラヒックとノード上を出入りするトラヒックに着目する。変数である未知のSD間の交流トラヒックの各リンクにおける和、ないし各ノードにおける和が、観測されるリンク上のトラヒックやノード上のトラヒックに等しくなるという制約条件を置く。この線形制約条件を満たすように、変数である未知のSD間の交流トラヒックを定めると、迂回パスを疎通するトラヒックは、障害リンクを指定したとき一意に定まり、両トラヒックを疎通できる、各リンク上の帯域容量も一意に定まる。変数を、この線形制約条件の下で自由に動かした時、各リンク上の帯域容量のとりうる最大値が求める解となる。

想定する障害リンク  $k$  に対して、帯域容量を算出するリンク  $h$  を疎通しうる最大トラヒックを求める最大化問題として定式化する。  $k$  を障害が起り得るすべてのリンクについて変化させたとき、  $P[h,k]$  をリンク  $k$  に障害が起きることを想定したときのリンク  $h$  の必要帯域とすると、  $P[h,k]$  が取り得る最大値は、リンク  $h$  における帯域容量となる。さらにこのリンク  $h$  を変化させて取り得る最大の帯域容量を各リンク共通の帯域とすればよい。

最大化問題  $P[h,k]$  の線形計画法としての定式化を以下に示す。また、帯域削減を目標としているのに最大化する理由は、パスを必ず疎通しパケットを必ず送ることができる帯域を確保する前提がある為である。

[表記]

$V, E$ : それぞれ、ノード、(有向)リンクの集合。

$i, j$ : ノードに対する index.

$h, k, l$ : リンクに対する index.

$n$ : 迂回パス本数.

[パラメータ]

$$W_{ij} : i, j \in V, W_{ij} \subset E \quad (1)$$

ノード  $i$  からノード  $j$  に至る現用パスを構成するリンクの集合。

$$B_{ij}^n : i, j \in V, B_{ij}^n \subset E \quad (2)$$

ノード  $i$  からノード  $j$  に至る迂回パスを構成するリンクの集合。

$$G_k : k \in E, G_k \subset V \times V \quad (3)$$

SD ペア  $(i,j)$  の現用パスが障害リンク  $k$  を含むような  $(i,j)$  の集合。このSD ペア  $(i,j)$  間は、現用パスを疎通させることができないので迂回パスを疎通させることになる。

$$D_i^{in} : i \in V, D_i^{in} \geq 0 \quad (4)$$

ノード  $i$  から通信網に流入する観測トラヒック量。

$$D_i^{out} : i \in V, D_i^{out} \geq 0 \quad (5)$$

ノード  $i$  で通信網から流出する観測トラヒック量。

$$T_l : l \in E, T_l \geq 0 \quad (6)$$

通信網上のリンク  $l$  における観測トラヒック量。

[変数]

$$X(i, j) : i, j \in V, X(i, j) \geq 0, X(i, j) = \sum_{k=1}^n X_n(i, j) \quad (7)$$

SD ペア  $(i,j)$  を流れるトラヒック量。通常は現用パスを流れる。現用パスが障害リンク  $k$  により疎通できなくなったときは迂回パスを流れる。

[目的関数]

$$\text{maximize} \quad \sum_{h \in W_{ij} \& (i,j) \notin G_k} X(i, j) + \sum_{h \in B_{ij}^n \& (i,j) \in G_k} X_n(i, j) \quad (8)$$

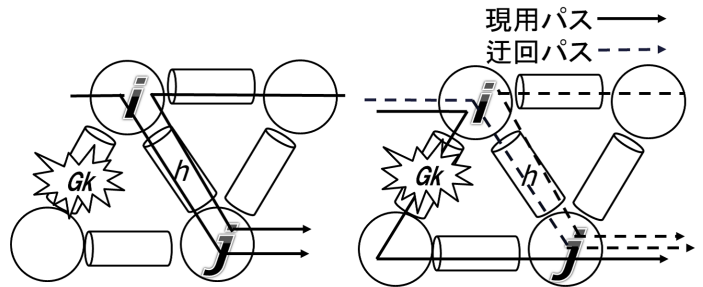


図1 式(8)の第1項  $\sum$  と第2項  $\sum$

現用パスと迂回パスを実際に流れるトラヒック量。障害リンク  $k$  一か所のみで障害が起っていると、そのときに実際に流れるトラヒックを最大化する。トラヒックの観測は障害リンク  $k$  とは異なる通信網上の1リンク  $h$  にて行う。目的関数の概要をノード数4のメッシュ網を例に図1に示す。

図1の左側のメッシュ網はSD ペア  $(i,j)$  間を現用パスが流れ、その現用パスが障害リンク  $k$  と異なるリンク  $h$  を流れるトラヒック量の集合で式(8)の第1項を表す。

図1の右側のメッシュ網はSDペア(i,j)間を現用パス、迂回パスが流れ、その現用パスが障害リンクkを通り、かつ、迂回パスがリンクhを流れるトラヒック量の集合で式(8)の第2項を表す。

[制約条件]

$$\sum_{i \in V-i} X(i, j) = D_i^{in}, i \in V \quad (9)$$

$$\sum_{i \in V-j} X(i, j) = D_j^{out}, j \in V \quad (10)$$

通信網の各ノードに着目し、そのノードから流入するトラヒックの総和と流出するトラヒックの総和が、それぞれ観測された流入トラヒック量と流出トラヒック量に等しいとする。

$$\sum_{i \in V} \sum_{j \in V-i} \sum_{l \in W_{ij}} X(i, j) = Tl, l \in E \quad (11)$$

通信網の各リンクに着目し、その(有効)リンクをその向きに流れる現用パスの実トラヒック量の総和が、観測された実トラヒック量に等しいとする。

## 5 複数の迂回パスを考慮したメッシュ網への適用

前章で述べた線形計画法を、複数の迂回パスを考慮したメッシュ網に適用する。それらのプログラム、方法、動作結果について説明する。

### 5.1 ILOG CPLEX[4]によるプログラム

本研究においては、メッシュネットワークモデルにおける制約条件を明確にし、ILOG CPLEXの環境を用いて実際にリンクに必要な帯域を求めることを実施した。プログラム内部においては、 $X(i, j)$ を現用パス、 $X_n(i, j)$ を迂回パスのトラヒック量とした。

### 5.2 プログラムの前提条件

実際には、ノード数は4から6までとし、全ノード間でリンクを繋ぐフルメッシュネットワークを用いる。例えばノード数4ではノード間を繋ぐリンク構造を図2に示すように定義する。

現用パス、迂回パスを流れるトラヒック量は1 [Mbps]から100[Mbps]までの中でランダムな自然数を取るように定義し、ノードiからノードjへの現用パス、迂回パスのルートは図2で示すように定義する。また、迂回パスの帯域を現用パスの帯域の半分を迂回パス帯域として分割する。迂回パス本数が増える毎に帯域の分割方法を決定する。例えば、迂回パス3本の場合は、それぞれ3分の1毎に帯域を分割する。

本研究の題目では迂回パスは複数本としているが、今回

の実験では2本から3本とする。同様にノード数5, 6, 7のメッシュ網のリンク構造、現用パス、迂回パスのルート、トラヒック量を定義し、また着目リンクhと障害リンクkをリンク構造から1本ずつ選ぶよう設定し、制約条件が適用されるように、プログラムを作成する。

ノード数毎の制約条件と目的関数は、C言語のプログラムで記述し、その値をILOG CPLEXで読み込み、リンク毎の帯域を導出した。

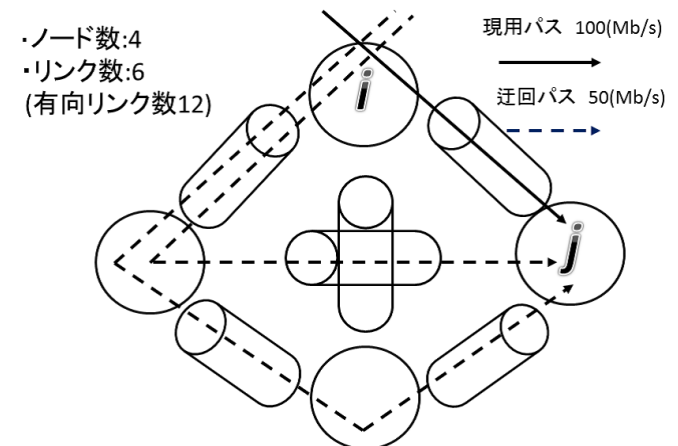


図2 ノード数4のメッシュ網のリンク構造とパス例

### 実験の条件

1. パスの設定として現用パスを設定した上でその現用パスと完全に一致しないように迂回パスを決定する。
2. パスの再配置はしない。
3. トラヒック量は発着ノード間で流れ、トラヒック量1~100のランダム値 [Mbps] の自然数の量が流れる。1つのメッシュ網に対してトラヒック量は10パターン用いる。
4. 着目リンクと障害リンクは、必ず同じリンクではないものにし、メッシュ網1つに対しそれぞれ1本設定する。
5. フルメッシュではあるがパスはランダムで決めるため、すべてのリンクを使わなくてもよい
6. 現用パスのルート設定は最短経路とする。
7. 迂回パスのルート設定は現用パスに次ぐ最短経路とする。
8. 複数本の迂回パスに対しては迂回パス1本毎に帯域を分割して流す。その分割パターンは1~3パターンとする。
9. 1つのメッシュ網1つに対してノード数は4~7個とする。

### 5.3 プログラムの動作結果

あらかじめ、メッシュ網のノード間を流れるトラヒック量と現用パス、迂回パスのルートを設定することにより、リンクに必要な帯域を自明解とした。さらに、トラヒック量、パスの設定をかえて着目リンク  $h$  と障害リンク  $k$  の組み合わせの全ケースによる線形計画で帯域容量を求め、自明解とその求めた解とを比較し正規化する。求めた値から必要帯域の平均値、及び、その 95% の信頼区間を求めた値を図 3 に示す。

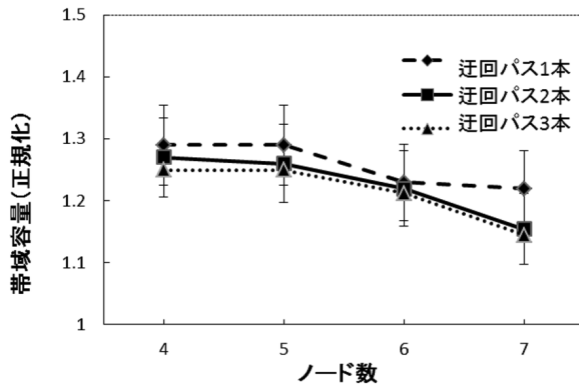


図 3 メッシュ網への適用性評価

### 5.4 考察

全試行において、図 3 より 4~7 ノードのフルメッシュネットワークにおいて 1.3 倍以下の帯域容量を確保すればよいと考えられる。そしてノード数増加につれ、帯域容量の正規化した値が減少している為、小ノード間でのシミュレーションであるが、ノード数増加にも対応しており、網のコストを削減する事が可能であると現段階では想定される。小ノードから構成されるメッシュ網では、複数の迂回パスの方が帯域容量が小さい。しかしながら迂回パス 2 本のととき、迂回パス 3 本のとときを比較すると帯域容量があまり減っていない為、小ノードにおいてはコスト的にも迂回パスの本数 2 本が最適であると考えられる。このことより、ノード数の大きなネットワークでは迂回パス本数を更に増やすことで、より帯域を改善できるのでないかと考えられる。

先行研究の迂回パス 1 本の場合よりも帯域に改善が見られていることより、トラヒック量のパターン数とノード数、試行回数を更に増やした場合にも提案法は同様以上の帯域改善の効果があると考えられる。

ノード数を増加するとともに、帯域容量も徐々に減少することが分かる。このことより大規模なフルメッシュネットワークになればなるほど、帯域容量において削減できると想定される。しかしながら、最適な迂回パス本数はノード数やフルメッシュであるかパーシャルメッシュであるかによって変わると考えられる。

また、ノード数が増加とともにリンクも増加する事から、

現用パスが障害リンクを通り、迂回パスが着目リンクを通らない場合が多く起こる。この場合は正規化した値は 1 となると考える。その 1 となる数がノード数増加とともに増加していく。よって平均値の減少、95% の信頼区間の減少と幅の減少が起こると考える。

## 6 まとめと今後の課題

今後の課題として以下のことが挙げられる。

ノードサイズをより大きくしたもので効果を確認する課題が挙げられる。本研究では経路探索アルゴリズムを用いず、フルメッシュで最短経路を求め迂回パスを設定するようにしました。その為、より現実的なネットワークを想定して、フルメッシュでない部分的にリンクが繋がるパーシャルメッシュ・ネットワークで適用する課題が挙げられる。

また、トラヒック量のパターン数を 10 パターンより増やすことで正確なシミュレーションを行うことができ、より結果の信頼性を上げることができると考える。

交流トラヒックが取得できない状況において、取得可能なトラヒック情報から複数の迂回経路用帯域を含めた必要帯域を線形計画法により求める方法をメッシュ網に適用した結果、4~7 のノード数から構成されるメッシュ網において帯域容量を削減する効果が期待できることを示した。

## 参考文献

- [1] 松村龍太郎, 辻野雅之, 岩下基, “不完全なトラヒック情報下での帯域設計法 線形計画法による定式化と定量評価”, 信学技報, ICM2008-78, pp.117-120, 2009.
- [2] 徳久正樹, 能上慎也, 阿部威郎, “フォトニック MPLS 網における光パス設計ソフトウェア”, NTT 技術ジャーナル, vol.15, no.7, pp.53-55, 2003.
- [3] 田中良明, ザニケエフ・マラット, “品質確保のためのネットワーク運用管理技術”, 電気通信普及財団研究調査報告書, no.23, pp.422-426, 2008.
- [4] ILOG CPLEX User's Manual, 2003.
- [5] 立花大祐, 田中善明, 内田真人, 鶴正人, 尾家祐二, “ネットワークトモグラフィを用いたトラヒック行列推定手法の実験”, 信学技報, CS2008-66, pp.55-60, 2009.
- [6] 杉野隆, “企業通信網における Hybrid 網構成決定のための数理モデル”, オペレーション・リサーチ, pp.210-215, 1997.
- [7] 大野祐史, 野原謙太, 大崎敬三, “不完全なトラヒック情報に基づくメッシュ網の帯域設計法”, 南山大学数理情報学部卒業論文, 2011.