

MPLS における複数パスの動的選択技術に関する研究

2005MT120 常川 勝広 2006MI120 西尾 広甫 2006MI134 岡田 千明

指導教員 奥村 康行

1. はじめに

近年ネットワークが拡大化に伴い、大容量データの FTP 転送、映像配信やテレビ会議等といった広帯域転送を必要とするアプリケーションが登場し、トラフィック量が増加している。トラフィック量の増加に伴いデータ転送の同時性、即時性が強く求められるようになった。また、個人ユーザとビジネスユーザのトラフィックが同じネットワークを利用する状況も増えてきているが、ビジネスユーザに対しては帯域最大遅延保証といった高品質サービスが求められ、個人ユーザに対しては低価格の帯域サービスが求められている。

以上のような問題を解決する技術として、トラフィックエンジニアリングや QoS(Quality of Service)が挙げられる。トラフィックエンジニアリングとは、複数の通信経路から負荷が分散するように経路を選択することであり、QoS とは優先度により適切な処理順序、信頼性などといった品質を管理することである。本研究ではフロー間の公平性を第一の目標とし、トラフィックエンジニアリングの観点から TCP トラフィックの転送性能を通常の IP ルーティングを行う場合と本アルゴリズムを適用する場合とでシミュレーション比較し、本アルゴリズムの特徴に関する議論を行う。[1]

2. IP ルーティングの課題

一般的な IP 網では OSPF(Open Shortest Path First)等のダイナミックルーティングプロトコルが用いられている。このような制御プロトコルでは SPF(Shortest Path First)アルゴリズムを用いた最短経路を計算することで IP パケットを宛先ネットワークへと転送するため、最短経路上のリンクノードにトラフィックが集中しやすい傾向があり、転送処理による遅延の増加や輻輳によるパケット消失が発生し易い。

その結果、TCP のフローの制御機能並びに輻輳制御機能の効果により高負荷な経路上を流れる転送機能の極度の劣化や負荷状態の異なる経路間における転送性能の不平等化、さらには、網全体にわたって考慮した場合における網資源の利用効率の低下といった課題が存在する。

3. MPLS (Multi-Protocol Label Switching)

トラフィックエンジニアリングの代表的なものとして MPLS

が挙げられる。MPLS では従来のルータの動作とは異なり、宛先 IP アドレスを見る代わりにラベルと呼ばれるパケットに付けられた識別子のみを見て転送を行う。それと同時に、どの IP アドレスがどのラベルを使うかの情報を配布された各ルータはルーティング・テーブルと共にラベル・テーブルを学習する。この処理を行うため MPLS ネットワークでは従来のルータの機能に加え、新たにラベル情報を配布、学習する機能を加える必要がある。

このようにラベルによって中継先を指定しつつ転送を行うことでルータは経路を選択、決定する処理から解放され、転送処理のみ行えば良いこととなり、個々のルータは負担が軽減され、転送処理は高速化される。しかし、現在ではハードウェアの発達によって高速化は十分に達成されているため、現在 MPLS に期待されている役割は自由に経路選択が可能である特徴を活かしたトラフィックエンジニアリングとなっている。

MPLS ネットワークではルーティングプロトコルとして CSPF(Constrained Shortest Path First)が適用される。CSPF は制限条件内でネットワーク上に K 本の経路を構築し、専用のメトリックや実際の帯域幅などの情報に基づき 1 本の最短経路を決定する。この制限条件やメトリックを操作することによりネットワーク上のトラフィックを管理者の望むように最適化することができる。このように、MPLS は IP では難しかった経路選択を実現する。例えば、帯域が空いている経路を動的に選択させることなどが可能となる。

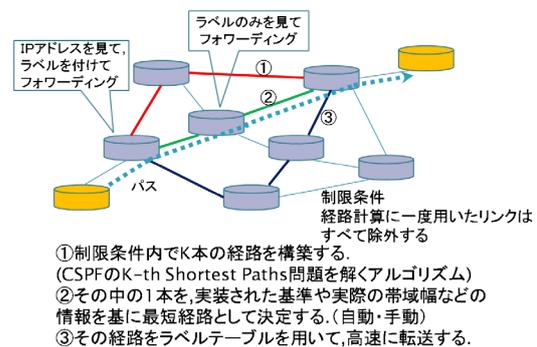


図 1 MPLS の概要

課題はトポロジーが大規模かつ複雑化した場合に運用が現実的に不可能である点が挙げられる。[2]

4. 動的負荷分散アルゴリズム

動的負荷分散を実現するために、以下の3つの機能に分類した。これらの機能が連携することにより、全体として一つの負荷分散機能が可能となる。

1. 複数の経路を用意する機能
2. 負荷情報を周期的に把握する機能
3. 負荷情報から経路を選択する機能

以降、各機能を実現する手段について述べる。

4.1. 複数の経路を用意する機能

負荷分散を実行するためには、同一宛先に対して2つ以上の経路が用意されていなければならない。すなわち最短経路を含め、第2経路、第3経路、…第K経路までを計算しなければならないが、このようなK-th Shortest Paths問題を新たに検討し、実装することは非常に困難である。そのため、本研究ではすでに確立された複数経路を前提とする。

4.2. 負荷情報を周期的に把握する機能

経路間の負荷のバランスを調整するためには、全ての経路の負荷情報を正確に把握しなければならない。

先行研究1 [3]では、負荷分散ノードが各経路に対して1本ずつ計測用のTCPフローを流し、それらのスループットを測定することにより各経路の負荷状態を測る。その理由はTCPフロー制御機構により同一経路を通る複数のTCPフローに対して、経路の物理帯域がほぼ均等に割り当てられるため、各経路に属する計測用フロー毎のスループットが経路の平均的な転送性能を反映していると言えるためである。ただし、計測用フローはリンク変更を行わないように設定する必要がある。

先行研究2 [4]では、各経路を通過するパケット数を負荷情報として用いる。

4.3. 負荷情報から経路を選択する機能

負荷情報を測定した後、その情報を基にして最適な経路へTCPフローを振り分けなければならない。

先行研究1では計測用フロー以外のある1本のフローを選択し経路制御を行う。この選択されたフローを各経路の負荷が均等になるように移動させる。この制御方法はフロー毎のリンク変更回数を均等にできるメリットがある。しかし、計測用フローが占有する帯域が大きな負荷となることが課題である。

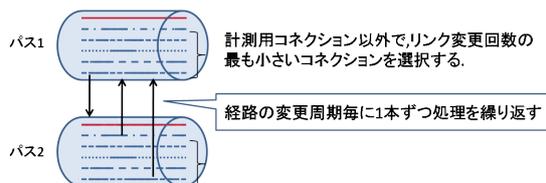


図2 先行研究1の経路選択機能

先行研究2では、負荷分散ノードで各パケットに対するハッシュ値を計算し、経路制御を行う。この付加されたハッシュ値とハッシュテーブルを分割する閾値と呼ばれる基準の値を比較することにより経路を選択する。

経路の使用効率に偏りが生じている場合、この閾値を移動させることで各経路に振り分けるフローの割合を変え、各経路の使用効率が向上するように調整する。

1. 全ハッシュ値を要素とするハッシュテーブルを生成
2. 1のテーブル内の要素を分類するハッシュ閾値を設定
3. 全経路の負荷が釣り合う方向へ2の閾値を移動
4. 負荷分散の周期毎に閾値操作を反復実行

通過したパケット数をカウントする変数であるパケットカウンタが設定した上限値に達したごとに閾値を1ずつ移動させる。

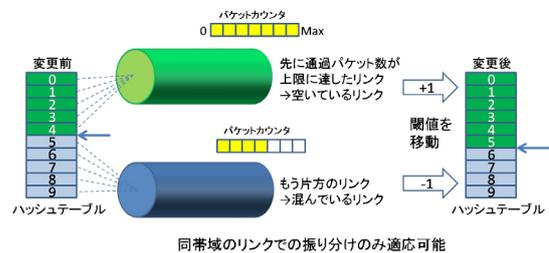


図3 先行研究2の経路選択機能

図に示す通り、閾値を変更するタイミングを「リンクを通過したパケット数をカウントする変数、パケットカウンタを用意し、あるリンクのパケットカウンタが設定した上限に達した時」としている。ここで、先に上限値に達しているリンクは他のリンクよりも多くのパケットを転送していることになるので、空いていると判断する。それに対し、他のリンクは混んでいるリンクと判断し、閾値の移動量は1ずつとしている。

課題として、先行研究2の閾値制御では、異なる帯域のリンクへの振り分けや3本以上の経路の振り分けについて考慮していない点が挙げられる。異なる帯域のリンクへの振り分けにおいて、TCPフローの平等性の観点から、ハッシュテーブルが帯域比で分割されていることが望ましい。しかし、先行研究2の閾値操作では常に広帯域のリンクのパケットカウンタの方が早く上限値に到達するため、最も高帯域のリンクにフローが集中してしまう問題がある。

5. 提案するアルゴリズム

本研究では、先行研究2の課題を受け、閾値を変更するタイミングと閾値の移動量を変更した次項の2つのアルゴリズムを提案し、その性能を比較する。

5.1. 提案1

本提案は、閾値を変更するタイミングとして周期を用いる。周期におけるフローあたりの通過パケット数の比でテーブルを分割するように閾値を動的に制御する。

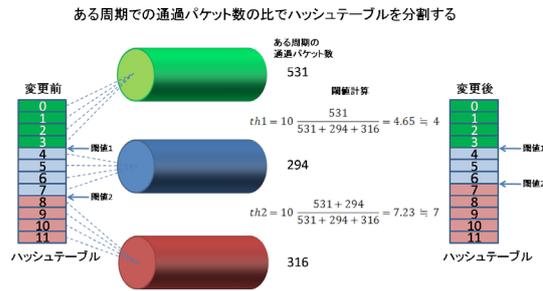


図 4 提案 1 の方法

5.2. 提案 2

本提案は、閾値を変更するタイミングを先行研究 2 と同様、ある経路のパケットカウンタが上限値に達した時に行う。

変更点として、閾値の移動量をパケットカウンタが上限値に達した経路へ割り当てるテーブル要素を 1 増やし、その時点でパケットカウンタが最小の経路へ割り当てるテーブル要素を 1 減らす。このようにして経路が 3 本以上の場合も同様に制御できる。

6. 実験

上述の提案 1, 提案 2 という動的な閾値制御と比較を行うために静的 1, 静的 2 という動的に閾値制御を行わないアルゴリズムを用意した。静的 1 は閾値を各パスに均等に振り分け、静的 2 はパスの帯域比で振り分ける。以上の 4 つの閾値制御を用意する。

様々な状況下で動的アルゴリズムの適用によって、経路間におけるフローあたりのスループットが均等に近づくことを示すため以下の 2 つの実験を行う。また、実験は NS-2 (Network Simulator 2) [1] を用いて行う。

実験の評価として、各経路のフローあたりのスループットの平均値と各経路の平均スループットのばらつきとして標準偏差を用い、範囲として平均スループットのプロット毎に折れ線でグラフ内に示す。つまり範囲が大きい程、各経路の平均スループットにばらつきがあり、平等性が失われていることとなる。

6.1. 実験 1

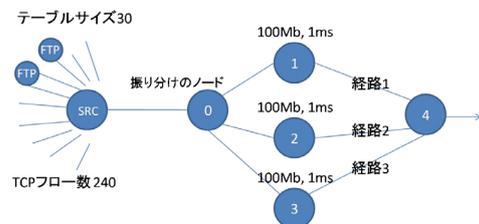


図 5 実験 1 のトポロジー

本実験のネットワークトポロジーは図に示す通り

100Mbps 3 本の経路、各経路の遅延 1ms、フロー数 240、シミュレーション時間 50sec のうち 18sec から 25sec を取り出して図に示す。本実験は、現実的には実現しにくい状況であるが、振り分ける帯域がすべて同等かつ TCP フローのみが発生し、輻輳や障害が全く起こらない状況を想定している。つまり、シミュレーション時間内において常に各経路に対して均等にフローが発生し、各経路のスループットが 100Mbps を保っている状況である。

その結果、動的な制御である各提案 1, 2 の標準偏差と比較して静的 1, 2 の標準偏差が小さくなった。これより本実験のような状況で提案した動的アルゴリズムを使う利点はなく、むしろ悪影響を与える結果となった。

6.2. 実験 2

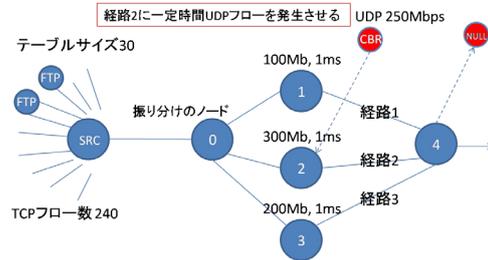


図 6 実験 2 のトポロジー

トポロジーは図に示す通り、100Mbps、200Mbps、300Mbps の 3 本の経路、各経路の遅延 1ms、フロー数 240、シミュレーション時間 50sec のうち 18sec から 25sec を取り出して図に示す。また、20sec から 40sec の間、において経路 2 に UDP フロー 250Mbps を発生させる。

本実験は以下 2 つの状況を想定した実験である。1 つ目は特定の経路に障害が発生し、実測値が低下した場合を想定する。つまり、シミュレーション開始時に障害が発生し 300Mbps 3 本から 100Mbps、200Mbps、300Mbps に低下したと仮定し、これらの異なる帯域の経路に対して障害が起こっていない同帯域のままの経路とみなして、静的に均等にフローを割り当てる制御 (静的 1) と動的アルゴリズムを比較することによって想定した状況の考察を行う。

図の見やすさを重視して静的 1 と提案 1, 2 を比較したものを図 8、静的 2 と提案 1, 2 を比較したものを図 9 とした。

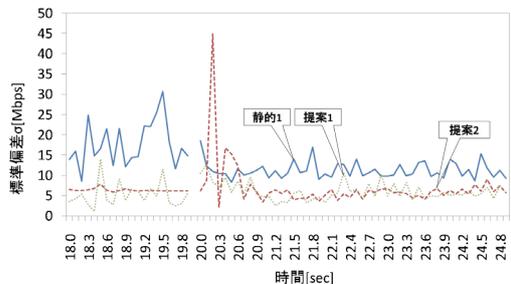


図 7 フロー当りのスループットの標準偏差 (実験 2)

図8の前半部のTCPフローのみの部分で静的1と比較して、提案1, 2の標準偏差が小さく、特定の経路の障害に対応して動的に制御されたことが示されている。

2つ目の想定する状況は特定の経路がUDPフローにより輻輳した場合である。図9の後半部のUDPフローが発生している間において帯域比でハッシュテーブルを静的に分割する制御(静的2)と各動的アルゴリズムを比較することによってUDPにより輻輳した状態での動的アルゴリズムの有効性について考察する。

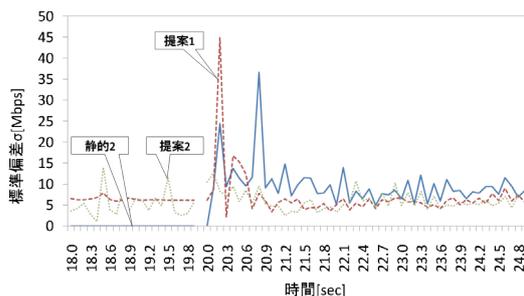


図8 フロー当りのスループットの標準偏差(実験2)

上図より、前半部のTCPのみの部分では静的2が最も標準偏差が小さい。これは実験1で示された結果と同様、TCPフローのみの場合、帯域比に応じてフローの振り分けを行った方が有効であるためである。しかしながら、UDPが発生する後半部では静的2の標準偏差の値が大きくなり、動的アルゴリズムの提案1, 2が標準偏差の範囲で優れるという逆転が生じる。これは静的な制御ではUDPによる輻輳を回避できないことを示している。UDPが発生した経路ではUDPフローの使用する帯域分だけTCPフローの使用できる帯域が減少する。そのため、フローの数が不変である静的2の制御ではその経路の平均スループットが低下し、各経路間の平均スループットの平等性が失われる。これに対し、動的1, 2の制御はTCPフローの使用帯域の減少を通過パケット数の減少から察知し、割り当てるフロー数を減らし平等性を保つため、UDPが発生した後半部で増加した標準偏差の範囲を徐々に小さくしていることが図から読み取れる。

7. 考察

実験1では動的な提案アルゴリズムよりも静的アルゴリズムの方が優れた結果となった。これは閾値の変更が頻繁に行われる動的制御はパケットの微小な変動に対して閾値を変動させるため、次にフロー変更が行われるまで偏りがあるように扱う。これにより、フローあたりのスループットの平等性が静的な制御に比べて失われたと考えられる。

実験2では静的なアルゴリズムと比べ提案アルゴリズム

の有効性が示された。また実験2において提案2よりも提案1の方が標準偏差の範囲を小さくする速度が上がる。これは上述したように提案1はパケット比を直接反映させるため、より高速な制御となったと考えられる。

実際のネットワークではTCP, UDPが存在し、障害による輻輳が考えられるため、現実のネットワークに近い状況において提案アルゴリズムの有効性を示すことができたと判断される。

なお本要旨では本研究のテーマである各経路のフローあたりの平均スループットの平等性についてのみ考察した。パケット損失量やそれに伴う平均スループットの変動については本稿に掲載する。

8. 課題

ただし本研究で負荷情報に用いた通過パケット数は変動が大きいため不安定である。よって経路の負荷情報を把握するための指標として単一では不十分であると考えられる。このため提案した2つの動的アルゴリズムにおいて大きな差を示すことができなかつたと思われる。ゆえにパケット数とは別の負荷情報を組み合わせることが望まれる。

加えて本研究で提案したアルゴリズムではハッシュ値とフローとの対応が固定されている。このためそのハッシュ値に対応する経路が変更され易いハッシュ値に対応づけられているフローは、変更が他のフローに比べ多くなる。それによりフローあたりのスループットが公平になっていてもユーザ視点では不平等が発生することが予想される。この問題の解決にはフロー毎に変更回数を記憶する機能といった、変更フローが平等になるような機能の追加が必要になってくる。

また本研究は経路が確立した状況下での振り分けアルゴリズムに絞って研究を行った。今後、実際にMPLSモジュールに本アルゴリズムを適応させシミュレーションを行うことによって、ルーティングアルゴリズムとしてより現実に即したものとされる。

参考文献

- [1] 銭飛, “NS2によるネットワークシミュレーション”, 森北出版, 2006.11.
- [2] Eric W.Gray, “マスタリング TCP/IP MPLS 編”, オーム社, 2002.2.
- [3] 福村 卓哉, 小川 功, 高橋 幸裕, “複数パスを用いたトラフィックの動的振り分け方式の研究”, 南山大学数理情報学部情報通信学科 2008 年度卒業論文, 2009.3.
- [4] 宮本 正和, 家永 憲人, “動的負荷分散型 IP 網の構成法と経路制御アルゴリズムの性能に関する研究”, 信学技報 CS2002-68, pp.25-30, 2002.9.