目 次

1 序論 **1**
1.1 研究の背景と目的 1
1.2 関連研究と問題点 1
1.3 本文の構成 2

2 消費電力の削減について **3**
2.1 はじめに 3
2.2 Energy Efficient Ethernet(EEE) について 3
2.2.1 EEE と Low Power Idle(LPI) 3
2.2.2 EEE の消費電力 4
2.2.3 Coalescing と EEE 5
2.3 OSPF と ECO-RP 7
2.3.1 コアルータとエッジルータ 7
2.3.2 Open Shortest Path First(OSPF) とネットワーク構造の最適化 7
2.3.3 ECO-RP について 9
2.3.4 OSPF コストの動的変更 10
2.3.5 コアルータの電源切り替え 11
2.4 問題点と解決手法 12
2.4.1 従来手法の問題点 12
2.4.2 経路選択に機器の消費電力が考慮されない 12
2.4.3 経路選択に機器の性能や情報が考慮されない 13
2.4.4 OSPF コストにより，不適切な経路選択が発生する可能性 13
2.5 おわりに 14

3 電力情報のモデル化 **15**
3.1 はじめに 15
3.2 実験環境と実験手法 15
3.2.1 ネットワーク機器の概要 15
3.2.2 実験装置の概要 16
3.3 実験結果と考察 17
3.3.1 Link up ポート数と消費電力の関係 17
3.3.2 トラフィックと消費電力の関係 18
3.4 結論と消費電力のモデル化 19
3.4.1 別の機器の結果と結論 19
3.4.2 消費電力のモデル化 20
3.5 おわりに 20

4 消費電力の最適化手法の提案 **21**
4.1 はじめに . . . 21
4.2 提案手法の位置づけ . . . 21
4.3 DCP(Define cost with Performance) . . . 22
4.3.1 DCP の定義式 . . . 22
4.3.2 DCP によるネットワークの最適化 . . . 23
4.3.3 DCP の問題点 . . . 24
4.4 DCN(Define cost with Neighbors) . . . 25
4.4.1 DCN の定義 . . . 25
4.4.2 基準値の定義 . . . 26
4.4.3 DCN のアルゴリズム . . . 27
4.4.4 DCN によるネットワークの最適化 . . . 28
4.4.5 係数 α,β の決定 . . . 29
4.4.6 DCN の問題点 . . . 31
4.5 DCNP(Define cost with Neighbors and Performance) . . . 32
4.5.1 DCNP の定義 . . . 32
4.5.2 基準値の定義 . . . 34
4.5.3 DCNP のアルゴリズム . . . 35
4.5.4 DCNP によるネットワークの最適化 . . . 36
4.5.5 係数 α,β の決定 . . . 37
4.5.6 DCNP の概念を再定義 . . . 39
4.6 おわりに . . . 41

5 比較実験と結果 **42**
5.1 はじめに . . . 42
5.2 実験環境，条件と手法 . . . 42
5.2.1 シミュレータの構造 . . . 42
5.2.2 実験条件と各種パラメータ . . . 45
5.2.3 初期 OSPF コストの設定 . . . 47
5.2.4 消費電力削減率 . . . 47
5.2.5 平均ホップ数 . . . 48
5.2.6 トラフィック溢れ状態の定義 . . . 48
5.2.7 ネットワーク電力の定義範囲 . . . 49
5.3 提案手法同士の比較 (正規分布トラフィック) . . . 50
5.3.1 消費電力削減率の評価 . . . 50
5.3.2 ホップ数の評価 . . . 52
5.3.3 トラフィック溢れ発生率の評価 . . . 54
5.4 提案手法と ECO-RP の協調動作 (正規分布トラフィック) . . . 57
5.4.1 消費電力削減率の評価 . . . 57

5.4.2 ホップ数の評価 59
5.4.3 トラフィック溢れ発生率の評価 61
5.5 提案手法 DCNP と従来手法 (正規分布トラフィック) 63
5.5.1 消費電力削減率の評価 63
5.5.2 ホップ数の評価 65
5.5.3 トラフィック溢れ発生率の評価 67
5.6 正規分布とコサイン波形トラフィックの比較 69
5.6.1 平均消費電力パーセンテージの比較 69
5.6.2 平均ホップ数の比較 71
5.6.3 トラフィック溢れ発生率の比較 73
5.7 まとめと考察 75
5.7.1 まとめ 75
5.7.2 考察 76
5.8 おわりに 78

6 結論 **79**

図 目 次

2.2.1 Maestro らが提案した Tramsmitter の LPI 状態遷移 [7] 3
2.2.2 Reviriego らが使用した EEE の消費電力表 [8] 4
2.2.3 Christensen らが定義した Coalescing の概念 [7] 5
2.2.4 Christensen らが行った消費電力と遅延の測定実験 [7] 6
2.3.1 ネットワーク例図 (エッジルータとコアルータ) 7
2.3.2 Dijkstra 法によるルーティング (黒太線) 8
2.3.3 ECO-RP の動作フローチャート 9
2.3.4 荒井らが提案した ECO-RP の動的 OSPF コスト更新 [12] 10
2.3.5 コアルータの停止条件 11
2.4.1 Dijkstra 法で経路選択されたネットワーク例 12
2.4.2 エッジルータ同士が並ぶネットワーク例 13
2.4.3 不適切な経路選択例 14
3.2.1 実験装置の概要 16
3.3.1 DGS-3426 の消費電力 (Link up ポート数) 17
3.3.2 DGS-3426 の消費電力 (トラフィック) 18
3.4.1 ネットワーク機器の消費電力 (Link up ポート数) 19
4.2.1 従来手法と提案手法 21
4.3.1 DCP による初期コスト更新と最適化 23
4.4.1 DCN の概念 25
4.4.2 DCN の概念 (フローチャート) 26
4.4.3 DCN による初期コスト更新と最適化 28
4.4.4 係数とそれぞれの消費電力平均 30
4.4.5 係数とそれぞれの平均ホップ数 31
4.5.1 DCNP の概念 33
4.5.2 DCNP の概念 (フローチャート) 34
4.5.3 DCNP による初期コスト更新と最適化 36
4.5.4 係数とそれぞれの消費電力平均 38
4.5.5 係数とそれぞれの平均ホップ数 39
4.5.6 DCNP の概念 (フローチャート) 40
5.2.1 シミュレータの構造 (フローチャート) 43
5.2.2 トラフィックモデル 44
5.2.3 NSFNET T1(1989 年)[13] 44
5.3.1 NSFNET T1 の最適化消費電力 50
5.3.2 ランダム生成ネットワークの最適化消費電力 51
5.3.3 NSFNET T1 の平均ホップ数 52
5.3.4 ランダム生成ネットワークの平均ホップ数 53

5.3.5 NSFNET T1 のトラフィック溢れ発生率 54
5.3.6 ランダム生成ネットワークのトラフィック溢れ発生率 55
5.4.1 NSFNET T1 の消費電力 57
5.4.2 ランダム生成ネットワークの消費電力 58
5.4.3 NSFNET T1 の平均ホップ数 59
5.4.4 ランダム生成ネットワークの平均ホップ数 60
5.4.5 NSFNET T1 のトラフィック溢れ発生率 61
5.4.6 ランダム生成ネットワークのトラフィック溢れ発生率 62
5.5.1 NSFNET T1 の消費電力 63
5.5.2 ランダム生成ネットワークの消費電力 64
5.5.3 NSFNET T1 の平均ホップ数 65
5.5.4 ランダム生成ネットワークの平均ホップ数 66
5.5.5 NSFNET T1 のトラフィック溢れ発生率 67
5.5.6 ランダム生成ネットワークのトラフィック溢れ発生率 68
5.6.1 NSFNET T1(均一コスト使用時) の平均消費電力 69
5.6.2 NSFNET T1(乱数コスト使用時) の平均消費電力 70
5.6.3 NSFNET T1(均一コスト使用時) の平均ホップ数 71
5.6.4 NSFNET T1(乱数コスト使用時) の平均ホップ数 72
5.6.5 NSFNET T1(均一コスト使用時) のトラフィック溢れ発生率 73
5.6.6 NSFNET T1(乱数コスト使用時) のトラフィック溢れ発生率 74

第1章　序論

1.1　研究の背景と目的

近年，ブロードバンドや光通信技術の普及が進み，大量のデータ転送が可能になった．同時にP2Pファイル共有ソフトや，ストリーミングによる動画配信が一般化したことで通信トラフィックの増加に拍車がかかり，クラウドコンピューティングの普及も相まって，近い将来に情報爆発が起こることが予見されている．なかでも通信を処理するためのスイッチ，コアルータといったネットワーク機器の不足は深刻な問題となりかねず，大規模データセンターの設置による消費電力の増加も深刻な問題として認知されている．2007年12月，日本で行われたグリーンITイニシアチブ会議における経済産業省の報告では，2025年までに通信トラフィックは190倍，ネットワーク機器の増加による電力増加が，現在の5.2倍になると報告された[1][2]．この問題を解決するため，電力の削減を目標とした研究が数多く行われてきた．中でも，ネットワークの構造を消費電力的に最適化し，不要なノードの電源を停止することで削減を行う手法は，比較的新しい最適化手法であり，規模の小さいネットワークでも高い削減率が期待できる手法として知られている．しかしながらこれらの従来研究では単体の装置のみで構成されたネットワークが対象となっており，異なる機器が偏在する環境での動作は保証されておらず，実際の通信トラフィックを考慮していないこともあり，その性能も未知数である．そこで本研究では，性能の異なるネットワーク機器が偏在する環境において消費電力が最適となるネットワーク構造を導出する手法を提案する．電力情報をモデル化し，シミュレーションを用いてネットワーク構造を電力的に最適化することで，実際のネットワークトポロジを変更することなく解の導出を行う，機器の消費電力を考慮したコアルータの電源切り替えの手法をプログラム実装し，シミュレーションによる評価を行う．

1.2　関連研究と問題点

増え続けるネットワーク機器の電力を削減する試みとして，様々な手法が提案されてきた．機器を構成するVLSIの消費電力を減らしネットワーク機器そのものの消費電力を減らす手法もその一つである[3]．またネットワーク機器側の消費電力削減として，送信パケット数に応じて通信帯域を変動させることで，ポートの消費電力を減らすAdaptive Link Rate(ALR)がある[4]．ALRはネットワークインターフェースカード(NIC)内にバッファ領域を用意し，一定量のパケットが溜まってから送信することで，転送量が少ない場合はネットワークの転送帯域を低く保つことができるようになっている．このほかEnergy Efficient Ethernet(EEE)はNIC内に存在するイーサネットトランスレータの電力削減としてLow Power Idle(LPI)という省電力モードを実装しており，パケットが流れてこない間はLPI状態にすることで省電力化を測っている[5]．このEEEはIEEE802.3azで規格化されており，より高性能な手法へと改良が急がれている[6][7][8]．搭載されたチップやボードを物理的に省電力化させる手法は，トラフィックの増大やデータセンターの消費電力問題が叫ばれるようになる以前から多数行われてきた．これらの手法は個々の機器における省電力化を実現することができ，国家レベルを対象とした大規模なネットワークにおいて有効な手法であるが，使用していないポートやNICを削減できず，規模が小さくなると削減率も小さくなるという問題を抱えていた．これらの手法を鑑みて誕生したの

が電力を最適化する式をもちいて，不要なネットワークインターフェースカードやポートを削減するGAMS-Based-Optimization である [9]．ある時点でのトラフィック情報を用いて計算式を定義することにより，電力的に最適となるネットワーク機器や NIC，ポートを使用し，不要となった機器や NIC を停止することで最適化を行い，消費電力の削減を図ろうとしたものである．同様の構想で，経路選択を光通信技術 (WDM) に適用したのが MILP-Optimization Models であり，こちらは光交換機の消費電力をも計算の視野にいれている [10]．これらの手法はトラフィックの経路を集約することができ，使用されていないポートや NIC を効率的に削減することができたが，ある時点のトラフィック情報を基準として最適化を行うため，時間や状態が変化すると最適性が失われてしまうという問題があった．また，削減対象がポートや NIC 単位であるため，シャーシ本体の電源を切ることができず，削減率が低めであるという特徴もある．この問題を解決するために登場したのが，ネットワーク機器単位で電力を動的に切り替える手法である．不要な L2 スイッチの削減を目的とし，イーサネット LAN の構造を電力的に最適化する手法もこの方式をとっている [11]．また GAMS-Based-Optimization と，トラフィックに応じたルーティングの動的更新を組み合わせ，コアルータの削減を図った ECO-RP は，従来から問題となっていた時間や状態の変化による最適性の喪失に対応している [12]．しかしながら，これらの研究はすべて均一な性能のネットワーク機器が存在していることが前提条件であり，現実のように異なる機器が偏在している環境での動作は考慮されておらず，こうした環境に適応した結果，性能が著しく低下するものも含まれている．そこで本研究では，機器の性能を考慮した消費電力最適化ネットワーク構造の導出を用い，性能や消費電力が異なるネットワーク機器が偏在する環境において，ネットワーク消費電力の最適化を行う手法を提案する．

1.3　本文の構成

本稿ではネットワーク機器の性能を考慮したネットワークの構造最適化，および消費電力最適化手法を提案する．2 章ではネットワーク消費電力の削減問題と，従来手法である Energy Efficient Ethernet，ECO-RP の解説を，3 章ではシミュレーションに使用するネットワーク機器の費電力情報のモデル化を，4 章では消費電力の最適化手法として，ネットワーク構造を導出する提案手法の解説を行う．5 章では従来研究との比較・考察を行い，6 章で本研究のまとめ，結論，統括を行う．

第2章　消費電力の削減について

2.1　はじめに

消費電力を削減する手法は，大きくわけて二つの部類に分けられる．個々のネットワーク機器が各々で削減を行う手法と，ネットワーク全体を考慮し，必要なトラフィック制御をおこなった後にポートなどを停止する手法である．前者は EEE，後者は ECO-RP である．従来では，チップや NIC の消費電力を削減することで機器を省電力化することに力が注がれており，ネットワークを考慮することは行われてこなかった．このネットワーク構造を最適化して消費電力を削減しようという試みは，近年，ネットワークが拡大し，複雑化が進んだことで新しく生まれた発想である．ネットワークの最適化による削減とは電力的に不要，または無駄であるポートやカード，シャーシそのものの電源を落として電力を減らすことである．電力を削減するためには，電源を切るポートや NIC を選択しなければならない．このとき，それぞれの条件に応じて適切な機器を選択する手法がネットワークの最適化である．どのように選択するのかは様々であるが，ECO-RP では Open Shortest Path First(OSPF) のコストを指針に用い，トラフィック変動に応じて動的変更することで不要な機器の選択を行っている．本章では EEE および ECO-RP を従来手法として例に挙げ，その動作原理を解説していく．

2.2　Energy Efficient Ethernet(EEE) について

2.2.1　EEE と Low Power Idle(LPI)

Energy Efficient Ethernet(EEE) は省電力ネットワークのために生み出された新しい規格であり，IEEE802.3az とも呼ばれている [4][5]．NIC 内に存在するイーサネットトランシーバの転送器 (Transmitter) は，送るパケットがない場合には常時起動させておく必要はなく，この時間を省電力モードにすることで消費電力の削減を図ることができる．この概念を物理層レベル (PHY) で実装したのが EEE である．Transmitter にデータを受け渡す機能をもつ受信機 (Receiver) を用い，連携して働くことできわめて効果的な動作を実現している．

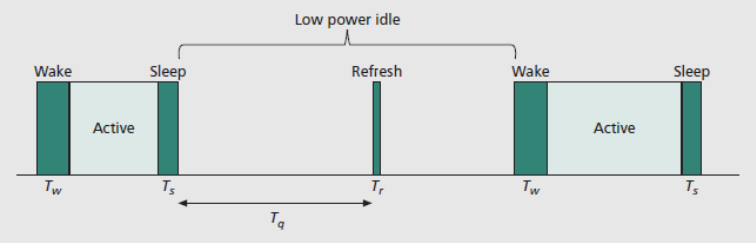


図 2.2.1: Maestro らが提案した Tramsmitter の LPI 状態遷移 [7]

図 2.2.1 は Transmitter の休止と再起動を示したものである．Transmitter はパケットが流れてこない間，Active 状態から省電力モードへと移行することで消費電力の削減を図っている．このとき必要な

Sleep 処理にかかる時間は T_s であり，終了後，Transmitter は省電力モードに移行する．この状態は Low Power Idle(LPI) と呼ばれ，時間の長さは T_q で表されている．パケットが到着した場合は Receiver からの信号によって Wake 処理がなされる．このとき，Active 状態に復帰するまでの時間が T_w である．またLPI 中に T_r 時間のリフレッシュ動作が存在する理由は，送信データの到着を知らせる Receiver との同期を取るためである．

2.2.2　EEE の消費電力

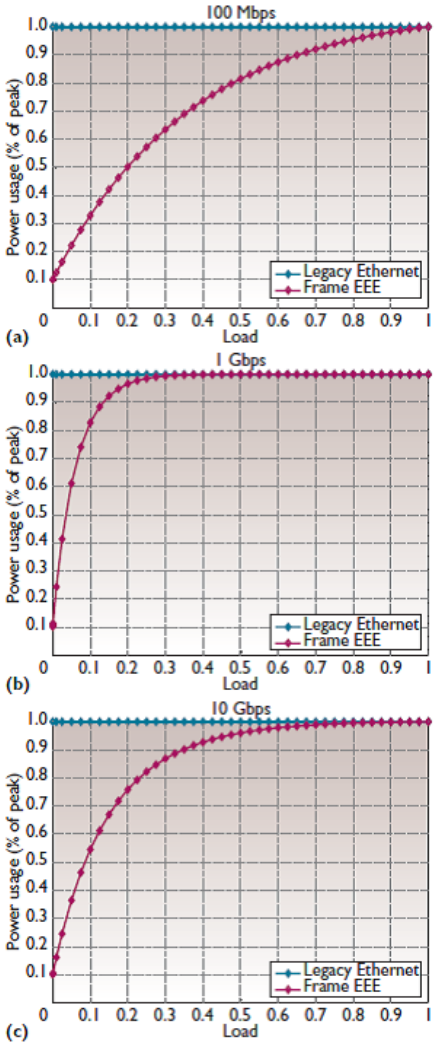


図 2.2.2: Reviriego らが使用した EEE の消費電力表 [8]

図 2.2.2 は (a)100BASE-T，(b)1000BASE-T，(c)10G-BASE-T に，それぞれ EEE を実装した際の消費電力の割合をしめしている．図中 Frame EEE が EEE の消費電力割合を示しており，どのグラフに

おいても，ある一定の load(回線使用率) までは一般的なイーサネットである Legacy Ethernet の電力を下回っている．この図より，使用率が低い場合は削減効果が高いが，高くなってくると悪くなるという特徴がみられる．これは EEE の行う Transmitter の LPI が，頻繁にパケットの到着する環境では効果的に運用できていないことを意味している．通常の EEE は，パケットが流れてこない間にしか LPI に移行できないため，省電力化の比率が低い．この問題を解決する手法がいくつか提案されてきたが，もっとも新しい手法として Coalescing の概念を持ち込もうという動きが広がっている．Coalescing とは Receiver 側にバッファを用意してパケットを蓄積し，制限時間が過ぎるか，一定数溜まったところで Transmitter へと転送．一斉に送信する，という考え方である．

2.2.3 Coalescing と EEE

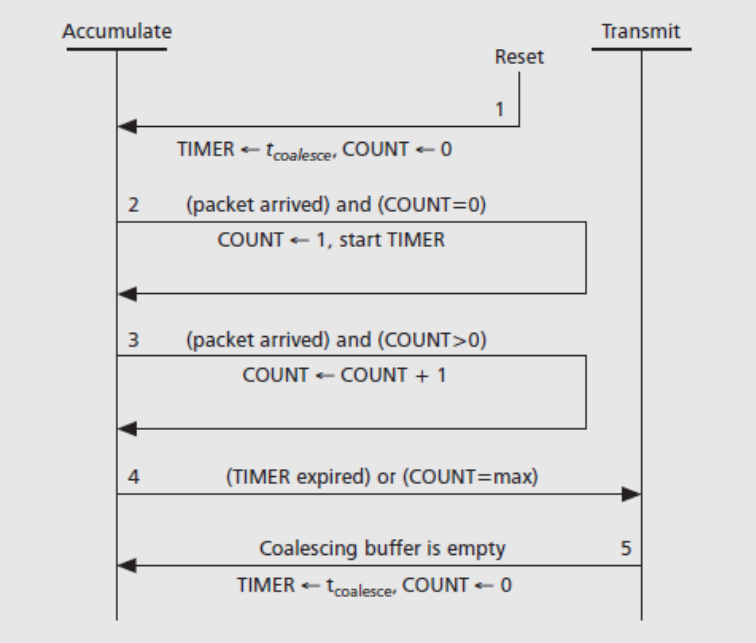


図 2.2.3: Christensen らが定義した Coalescing の概念 [7]

図 2.2.3 は Coalescing の概念を有限状態オートマトンで示したものである．図にある Accumulate はデータの蓄積を意味し，Transmit は溜まったデータを一斉に転送する動作になっている．制限時間を意味する TIMER とパケットの最大数を示す COUNT であり，COUNT が 0 のときにパケットが到着するとカウントが始まる．$T_{coalesce}$ が制限時間をすぎるか，COUNT が最大になった時点で転送が行われる．EEE にとっては省電力モードをいかに効率よく，長い時間続けられるかが最大の課題である．このバッファリングは関連研究である ALR にも使われていた技術であり，到着するパケットを Receiver 側に用意したバッファに一定数貯めおくことで逐次送信の必要性がなくなり，Transmitter をより長い時間，省電力モードにすることができる．

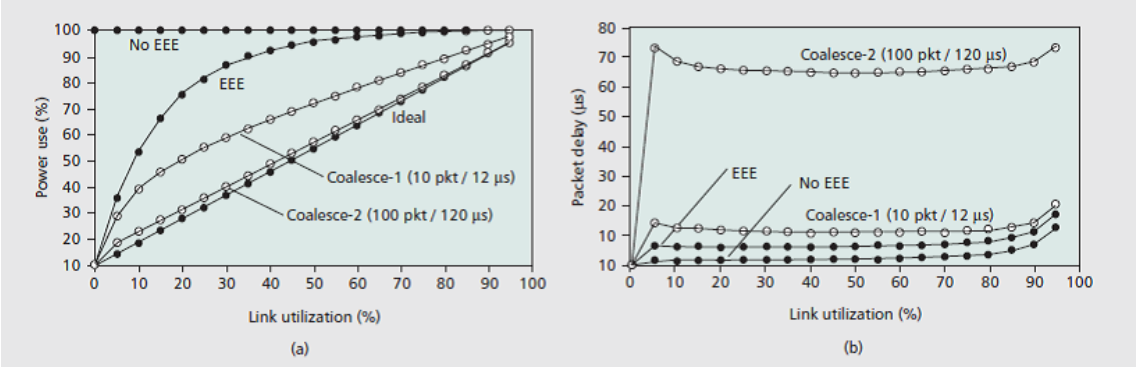


図 2.2.4: Christensen らが行った消費電力と遅延の測定実験 [7]

EEE は NIC 上で動作するため，ポート毎における消費電力の削減が主な効力になる．図 2.2.4(a) は 10GBASE-T 回線において，トラフィック流量と，ポート消費電力の関係を示したものである．Coalesce-1，Coalesce-2 は先に記述したバッファリングを実装したものであり，TIMER の制限はそれぞれ 12μs，120μs，蓄積パケットの最大値はそれぞれ 10pkt と 100pkt である．図を見てみると LPI が使用されない No-EEE がもっとも消費電力が高く，ついで EEE となっている．Coalescing が実装された二つは高い削減率を記録した．

また図 2.2.4(b) はデータの遅延を示したものである．Coalescing が実装された EEE は NIC の Receiver 側にパケットを貯めおくという性質上，どうしても余計な遅延が発生してしまうが，その差は非常に小さいことが分かる．両者ともに遅延が大きく，とくに消費電力が優秀だった Coalesce-2 は遅延が大きすぎるという問題が明らかになっている．遅延と消費電力のトレードオフを比較した結果，もっとも優れたのは Coalesce-1 である．この結果より，EEE に Coalescing 機能を付加することは非常に有効であるといえる．なお実際に EEE が削減できるのは各々のポート消費電力であり，非常にわずかなものでしかない．けれどネットワーク機器を停止させる必要がなく，常に起動させていなければならない環境においても効果を発揮できるという利点がある．開発者である Ken Christensen らが行った試算では，全米のネットワークに EEE を実装した場合，回線毎に平均 20 パーセント程度の電力削減が期待でき，最大で 8000 万ドルの電力費が削減できるとしている．今後は EEE に Coalescing 機能の実装が行われていく予定である．

2.3　OSPF と ECO-RP

次にネットワーク構造の最適化手法である ECO-RP を取り上げるが，先にネットワークルーティングプロトコルである OSPF，およびネットワーク機器の種類として，コアルータとエッジルータについての解説を行う．

2.3.1　コアルータとエッジルータ

ネットワークを構成するスイッチやルータは大きく分類するとコアとエッジに分けることができる．本稿ではそれぞれをコアルータ，エッジルータと呼ぶことにする．またネットワーク図を用いて解説を行う際はノードと呼称する．

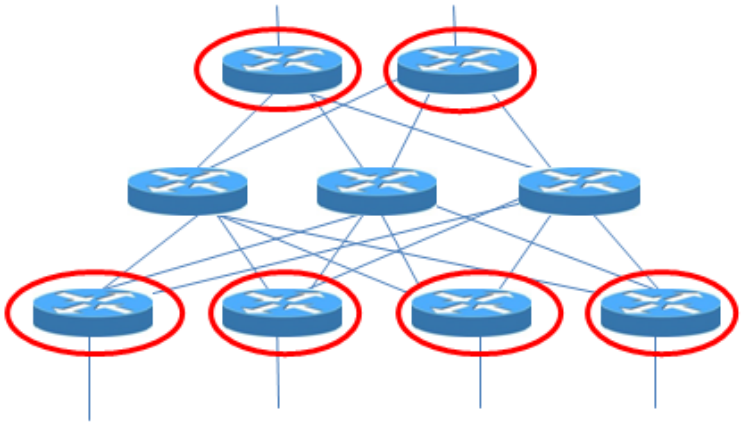


図 2.3.1: ネットワーク例図 (エッジルータとコアルータ)

図 2.3.1 はネットワークの一例を示したものである．ネットワーク図の中でも端に位置し，赤丸で囲まれたノードはエッジルータと呼ばれる．これらは内部の通信だけでなく，ネットワーク外部の通信をも受け持つため，必ず経路上に存在するという特性を持ち，原則として常時動き続けなければならない．一方，先の定義に含まれない，経路の途中に存在するノードはコアルータと呼ばれる．こちらはエッジルータを相互に接続する役目を果たしているが，ルーティング方式によっては経路に含まれない回線やコアルータも存在する．これについては次の OSPF とネットワーク構造の最適化にて詳しく述べる．

2.3.2　Open Shortest Path First(OSPF) とネットワーク構造の最適化

Open Shortest Path First(OSPF) とは最短経路導出アルゴリズムが組み込まれた経路探索プロトコルである．OSI 参照モデルのネットワーク層に位置し，異なる複数のネットワークをつなぐために，エリアおよび自律システム (Autonomous System, AS) という概念を用いてネットワークを管理する．OSPF のもっとも重要な要素は OSPF コストと Dijkstra 法を用いた最短経路導出であり，ルータが定期的に

互いの情報をアドバタイジングしあうことで，自動かつ柔軟にルーティングテーブルを作成する能力があるため障害に強いという特徴を持つ．反面，ルーティングに Dijkstra 法を用いるため，コストの低い回線にトラフィックが集中しやすく，輻輳やトラフィック溢れが発生しやすいという問題も抱えている．このコストは回線帯域によって自動で計算されるほか，1 から 65535 の間で自由に設定することもできる．OSPF コストを I とした場合，一般的なコスト導出式は次のとおりである．

$$I = C/B \tag{2.3.1}$$

ここで B はコストを設定するインターフェースの帯域，C は任意の整数である．初期の OSPF では $C = 100$ と定義されていたが，1000BASE-T，10G-BASE-T 等のインターフェースの改良とともに速度の上昇が進み，帯域 B が大きくなったことで，状況に合わせて任意に設定できる機器が多くなっている．そこで本研究では，$C = 1000$ を定義し OSPF コストの導出式として用いるものとした．また I は整数であり 1 以下にはならず，計算で 1 以下になった場合は強制的に 1 として設定される．

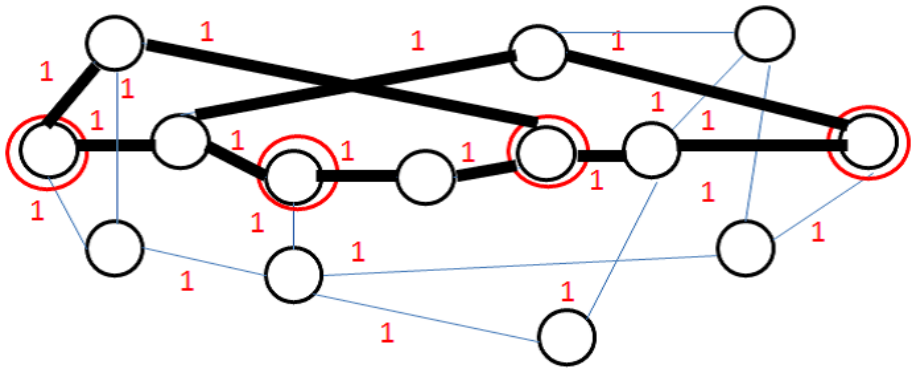


図 2.3.2: Dijkstra 法によるルーティング (黒太線)

図 2.3.2 は実際に Dijkstra 法を用いてルーティングを行ったものである．黒丸はノードでありルータを示すが，このうち赤丸で囲まれているノードはエッジルータ，囲まれていないものはコアルータである．それぞれのノード間をつなぐ回線は青線で，割り当てられた OSPF コストは赤字で記載されている．通常，OSPF コストは定義式または手動で設定されるが，例ではトラフィックを考慮しないものとし，すべてを 1 と定義した．先の図のうち黒い太線になっている部分が選択された経路であり，エッジルータ同士を互いに接続しているが，現在の経路は上側に偏っていることが分かる．このうち経路に含まれていないコアルータや回線も同様に動き続けているが，障害発生時にしか経路を見直さないという OSPF の特性からデータが流れてこない．このとき消費される電力を無駄と位置付け，電源を切るなどして削減しようというのがネットワーク構造の最適化である．先のネットワークを例にとった場合，Dijkstra 法による最適化を用いると 5 ノード，9 リンクの削減が可能になる．

2.3.3　ECO-RP について

先に解説したOSPFは現在，一般のネットワークに広く利用されている技術である．設定されたOSPFコストに応じてDijkstra法が最短経路を導出し，ルーティングテーブルを決定する．通常のOSPFは，経路の途絶や輻輳といった障害が発生しない限りルーティングテーブルの見直しや更新をすることがない．ここで現時点のトラフィック情報を収集し，経路決定の鍵となるOSPFコストを任意の時間毎に更新するようにしたものがECO-RPである [9].

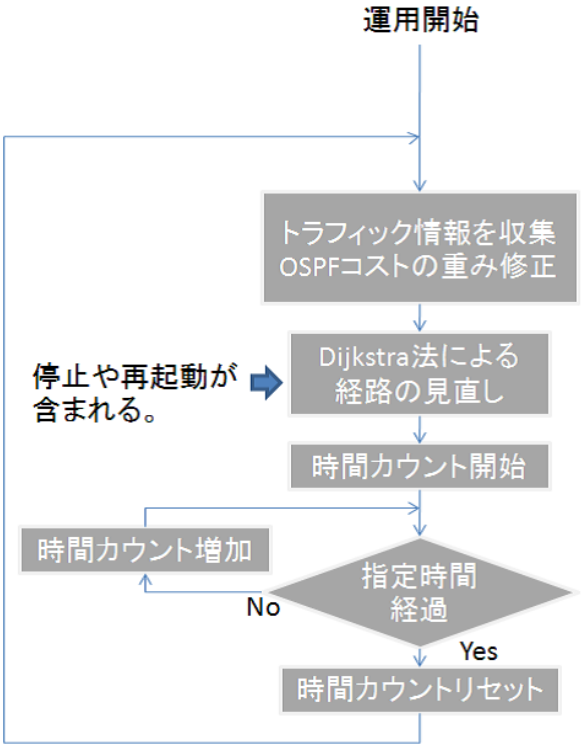


図 2.3.3: ECO-RP の動作フローチャート

図 2.3.3 に ECO-RP の動作フローチャートを示す．アルゴリズムは次の三つからなる．

- トラフィック情報に応じたOSPFコストの動的更新
- OSPF(Dijkstra法) による最短経路導出
- 選択されたコアルータの動的電源切りかえ

このうちECO-RPが独自に採用しているものは，トラフィック情報に応じたOSPFコストの動的更新である．常に最新の情報を用いることで，適切な経路選択を図っている．OSPFによる最短経路導出は先に示したとおりであり，次にOSPFコストの動的更新とコアルータの動的電源切り替えについて解説していく．

2.3.4　OSPF コストの動的変更

先に述べたとおり ECO-RP にはトラフィックに応じた OSPF コストの動的更新が実装されている．このコストは任意の時間毎に更新することができるため，常に構造の最適性を維持できるという特性を持っている．

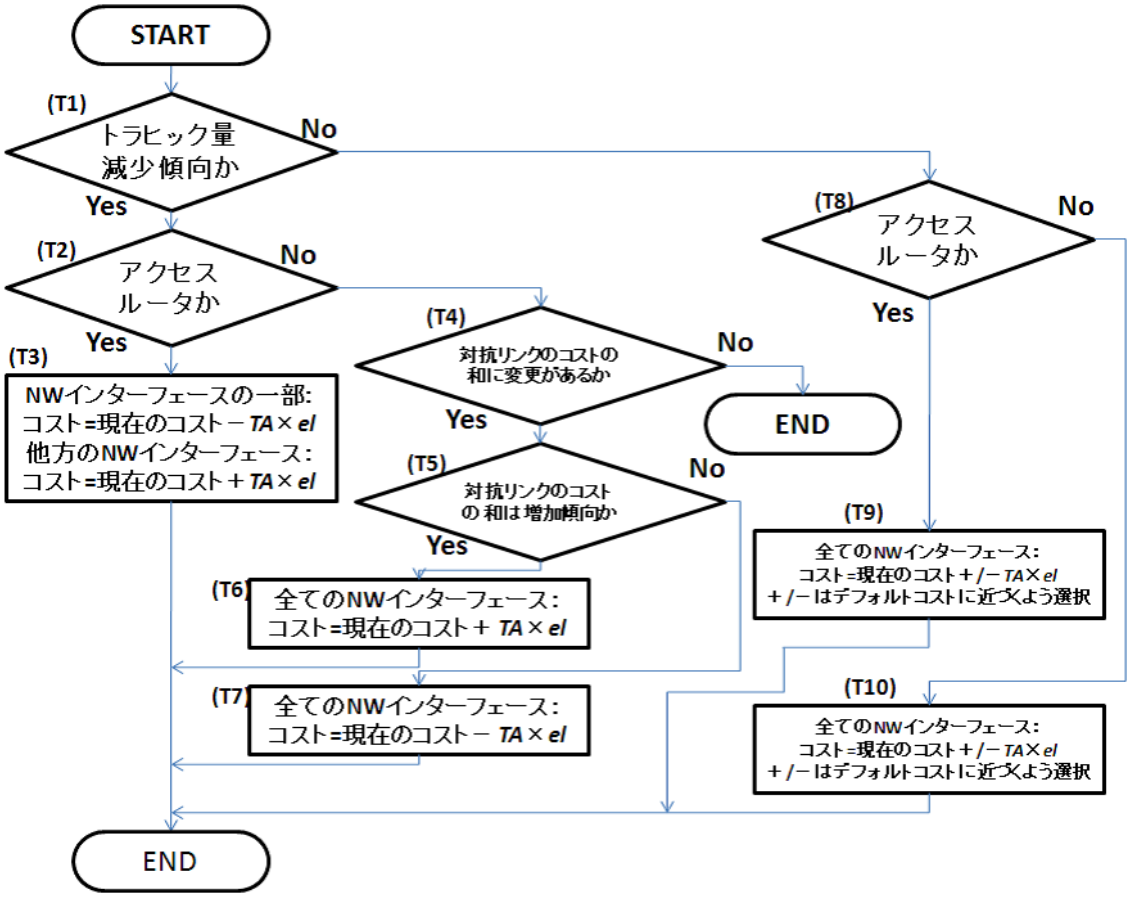


図 2.3.4: 荒井らが提案した ECO-RP の動的 OSPF コスト更新 [12]

図 2.3.4 は実際に ECO-RP に使われている更新アルゴリズムをフローチャートにしたものである．変数 TA は前回の更新時のトラフィックと現在を比べた差分であり，el は彼らが独自に設定した係数になっている．動的更新によるコスト変動は，元となる最初のコストからプラスマイナスの一定範囲内にしか変動しない．el の値は 0.2，0.4．0.6 と設定されており，el が 1 に近づくほど変化量は大きくなる．図のフローチャートは START から始まり，各々のルータやスイッチについて状況を判断していく．T1 ではネットワークを流れるトラフック量が減少しているか，上昇しているかの判別を行う．その後，T2 と T8 でルータの識別が行われるが，ここでいうアクセスルータとはエッジルータのことである．トラフィックが減少傾向の場合，エッジルータは奇数偶数で回線のコストをプラスマイナスに切り替え (T3)，コアルータの場合は向かってくるリンクにコスト変動がある場合のみ処理を行い，コスト上昇時には

コストを上げ，下降時には下げることで経路を集約させることを目的とする (T4,T5,T6,T7)．ECO-RP のコスト更新は基本的にトラフィックが上昇しているときのみであり，さがるときには初期のコストに近づくように変動する (T9,T10)．ECO-RP は機器の性能にかかわらず，特定の回線に通信を集中させることにより，コアルータの削減を目的としている．そのため現時点では，そこに起こる輻輳等については考慮されていない．機器の情報を考慮しないため削減率は低く，提案者である荒井らが行った 27 台のエッジルータ，22 台のコアルータを用いたシミュレーション実験では，4.2 パーセントから 7.4 パーセントの削減率にとどまっている．

2.3.5　コアルータの電源切り替え

トラフィックの流れないコアルータに使用されている電力が往々にして無駄になることは先にも少し述べた．ここではネットワーク構造を最適化するにあたり，停止できるコアルータの条件を記述する．

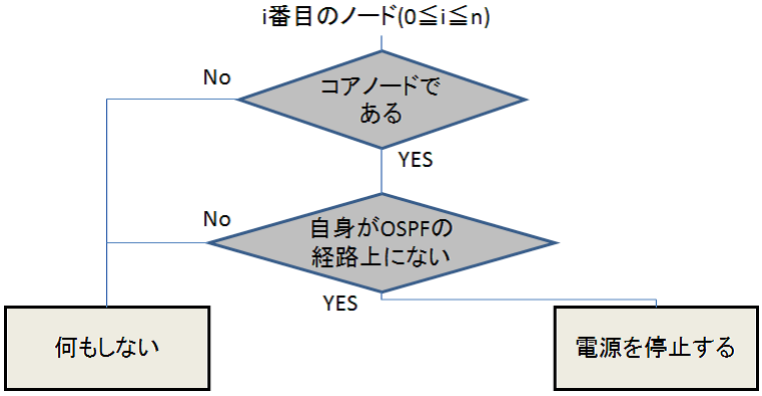


図 2.3.5: コアルータの停止条件

図 2.3.5 は停止できるコアルータの条件をフローチャートにしたものであり，n 個のノードが存在するネットワークにおける i 番目の機器の電源停止判断を行っている．対象となるノードがエッジルータでないこと，対象となるノードが Dijkstra 法で導かれる最短経路上に存在していないことが停止の条件となる．しかしながら通常のネットワークの場合，パケットが全く流れない状態というのは実現が難しい．そこで ECO-RP では OSPF コストを動的に変更し，通信経路を任意の回線に集中させることで，パケットの流れない回線を作り出す手法を使用している．経路から外れたコアルータは OSPF のアドバタイジングパケット以外が到着しなくなるため，電源を切っても通信には影響がない．

2.4 問題点と解決手法

2.4.1 従来手法の問題点

ここまで従来手法として EEE と ECO-RP について解説を行ってきた．両者の問題点として，次のようなことがあげられる．EEE はポートの電力をわずかに減少させるため，環境によっては満足な電力削減ができないという問題がある．この問題点を解決するため，コアルータの動的電源切り替えを実装した ECO-RP はより多く問題を抱えてしまった．

- 経路選択に機器の消費電力が考慮されない
- 経路選択に機器の性能や情報が考慮されない
- OSPF コストにより不適切な経路選択が発生する可能性

次に，これらの問題点について解説を行い，解決方法を提示する．

2.4.2 経路選択に機器の消費電力が考慮されない

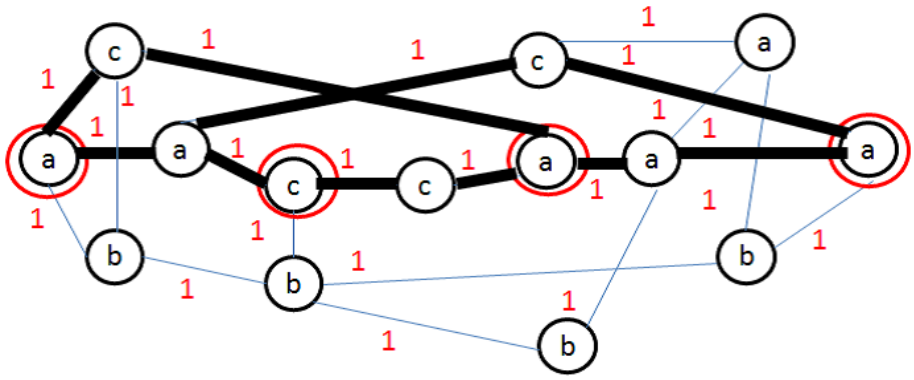


図 2.4.1: Dijkstra 法で経路選択されたネットワーク例

図 2.4.1 は OSPF の説明にて用いたネットワーク図例にノードごとの番号を割り振ったものである．番号が同じものは同じルータとし，性能や基礎消費電力は変わらないものとする．今，上の図では赤丸が示すエッジノード同士をつなぐ黒線が経路となって示されているが，OSPF の項でも述べたとおり，経路が上側に偏っているのがわかる．さらに言うならば，下側に位置した b の番号が振られたノードはどれも経路に属してはいない．逆に a の番号が振られたノードはすべて経路に所属していることが分かる．現在はコストも同一であり，完全に最短経路を通るルーティングが行われている．もし今処理能力の差がない，あるいはどちらの機器を使用しても問題がないと仮定した場合，a が b の何倍もの消費電力を持つルータだとしたら差分が無駄になる．処理するトラフィックが少ないのなら，高性能でも多

くの電力を必要とする機器は休ませ，性能がある程度低かったとしても電力が低いもので運用したほうが，消費電力の面からも，ネットワーク運用の面からも効率的である．ノードごとの消費電力を考慮できない現状では，異なる機器が偏在する環境においては能力を発揮することができない．そこで本研究では，実際のネットワーク機器を用いて消費電力を測定し，モデル化した電力情報を用いて経路選択を行う手法を提案する．消費電力が小さい機器にはデータが流れやすく，大きい機器には流れにくくなれば，経路選択は消費電力の面から最適化され，ネットワーク全体の電力は下がるはずである．

2.4.3　経路選択に機器の性能や情報が考慮されない

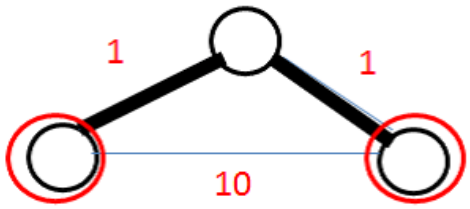


図 2.4.2: エッジルータ同士が並ぶネットワーク例

図 2.4.2 はエッジルータ同士が互いに隣り合うネットワークを模した図である．赤丸で囲まれたエッジルータは，OSPF コストが 10 の回線で接続されている．一方，迂回路として，OSPF コスト 1 の回線，および別のコアルータが存在しており，図中黒線を見るとわかるとおり，Dijkstra 法を用いた OSPF では遠回りの経路を選択しているのがわかる．もし意図的に設定し，隣り合う経路を避けるように設定されたのであれば問題はないが，仮に OSPF が自動でこのようなコストを設定したとなれば解決すべき問題である．消費電力と利便性はトレードオフの関係にある．輻輳が起こり，サービスがダウンしない範囲であるのなら，ある程度の遅延を享受することで消費電力を下げることができる．すなわちエッジルータ同士が隣り合う場合は，できるだけ隣り合うノードを遣うように誘導してやる必要がある．またエッジルータ同士にかかわらず，消費電力の低いノードが隣り合う場合には，つながる回線のコストを低くしてやることで，低消費電力な機器をつなぐ回線にトラフィックを集中させることができる．これにより不要な機器を削減できるため，上の図のような問題は起こらなくなる．

2.4.4　OSPF コストにより，不適切な経路選択が発生する可能性

図 2.4.3 は OSPF コストの設定によって発生する不適切な経路選択の例を示したものである．OSPF によるルーティング計算は Dijkstra 法によって行われるため，トラフィックはよりコストの小さい回線へと流れていく．結果として，図のような場合は経路が拡散してしまい，本来なら止められる機器までトラフィックが流れてしまう事態になっている．右側のノードがトラフィックを流さなかったとしても，OSPF コストが 10 の回線は，左側のノードとの通信に使用され続ける．したがって右側の OSPF コスト 1 に流れているトラフィックも，左側の OSPF コスト 10 の経路を通ることで，右のノードと回

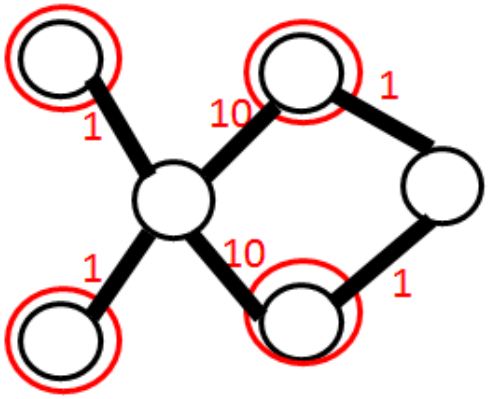


図 2.4.3: 不適切な経路選択例

線を削減することができるようになる．この問題を解決する手法としては経路の重複判定があげられる．ネットワークの通信は，輻輳が起こらない範囲ならばできるだけ集中していたほうが，利用の面からも，電力的な面からも効率がよい．ここで重複度の定義を新たに加え，何本の経路が回線に集中しているかを測り，一定数以下の経路を一定数以上の経路に変更することで，無駄な経路を省こうという試みである．もし，一つのノード間でしか使われていない経路を省くことができるのなら，右側の回線にはデータが流れなくなり，経路は OSPF コスト 10 の回線に統合される．これにより，右のノードは電源を落とすことが可能になる．しかしながら，本研究が目的としているのは，機器の電力情報を用いて，異なるネットワーク機器が偏在する環境の消費電力最適化をはかるものであり，機器情報を使用しない手法については考慮していない．よってこの問題は将来に解決されるものとして残すこととする．

2.5 おわりに

本研究では，ネットワーク全体でみた電力削減率を向上させるため，OSPF コストを用いた経路選択に含まれないコアルータの動的に電源切り替えを行う手法を提案する．本章ではネットワーク消費電力を削減する従来手法として，ポート毎の削減を行う EEE，OSPF の経路に所属しないルータを削減する ECO-0RP について解説を行った．また電源切り替えを実装するにあたって発生する問題のうち，機器の消費電力が考慮されない，機器の性能，情報が考慮されないという問題を考慮し，解決のための分析を行った．本研究におけるネットワークの最適化計算は，実際のものではなくシミュレーションで行われる．実際のネットワーク運営についても，結果を出してから OSPF コストを変動させ，適用するのが常である．これに先立って実機にて消費電力の測定をおこない，シミュレーション用のパラメータをモデル化する．電力情報のモデル化は 3 章に，消費電力と機器情報を用いたネットワーク構造の最適化手法は 4 章に記載する．

第3章　電力情報のモデル化

3.1　はじめに

本研究ではネットワーク構造の最適化と効率のよい消費電力削減を目的としており，消費電力情報を用いた OSPF コストの更新と，シミュレーションによるネットワーク構造の最適化を行う．ここで実際にシミュレーションを用いた最適化を行うためには，モデルとなるネットワーク機器の消費電力を用意する必要がある．本章では，実際に機器の消費電力を測定し，最小二乗法による近似を用いてモデル化することとした．今回使用するのは D-link 社製の L2+スイッチである DGS-3426，DGS-3450 および，EXTREME 社製 L3 スイッチの Alpine3408 である．

3.2　実験環境と実験手法

3.2.1　ネットワーク機器の概要

表 3.2.1: ネットワーク機器の概要

	DGS-3426	DGS-3450	Alpine3804
Normal interface	1000BASE-T 24 ports	1000BASE-T 48 ports	1000BASE-TX 8 ports
Extension interface 1	1000BASE-SX 4 ports		1000BASE-SX 4 ports
Extension interface 2	10G-BASE-LR 2 ports		100/10BASE-T 48 ports
Switch fabric	88Gbps	136Gbps	32Gbps
Packet forwarding	65.47Mpps	101.19Mpps	24Mpps

表 3.2.1 に実験で使用した装置の概要を示す．DGS-3426 に搭載されているインターフェースは 1000BASE-T が 24ports，1000BASE-SX が 4ports，10G-BASE-LR が 2ports，DGS-3450 は 1000BASE-T が 48ports，1000BASE-SX が 4ports，10G-BASE-LR が 2ports であり，Alpine3480 は 100/10BASE-T が 48ports，1000BASE-T が 8ports，1000BASE-SX が 4ports である．これらの光インターフェースの Small Form factor Pluggable(SFP) 及び 10Gigabit Small Form Factor Pluggable(XFP) と呼ばれる増設インターフェース Giga Bit Interface Converter(GBIC) はすべて増設済みとした．

3.2.2　実験装置の概要

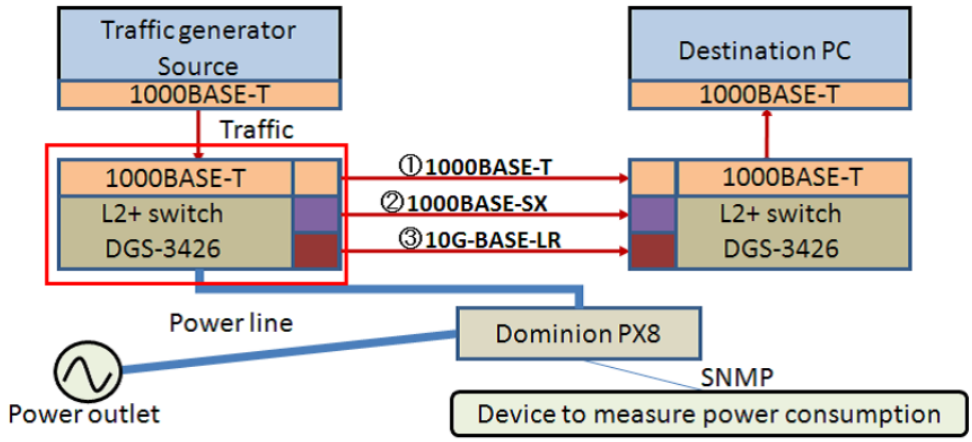
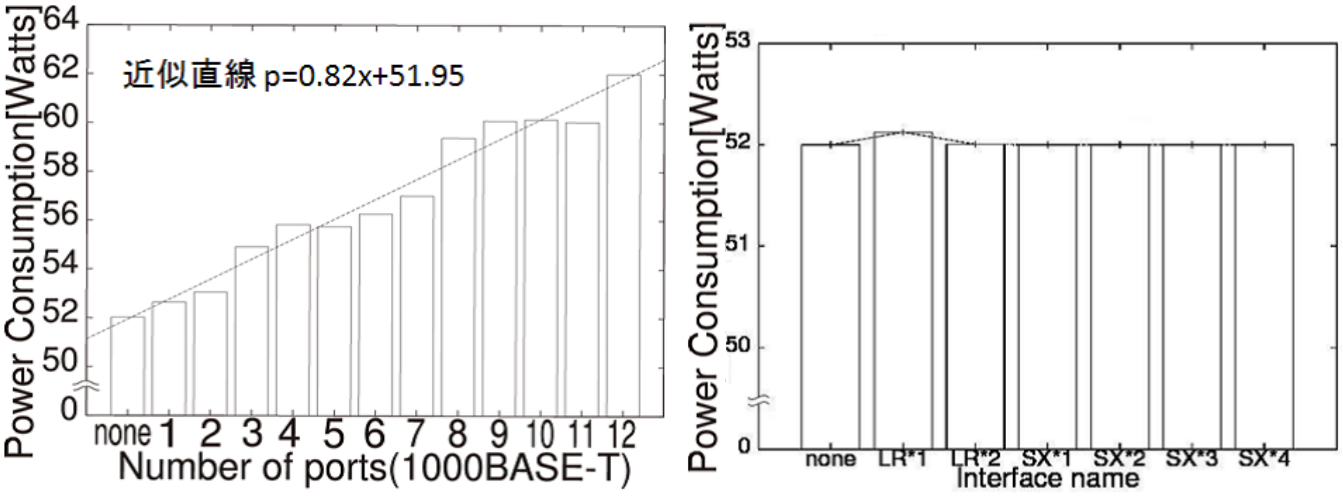


図 3.2.1: 実験装置の概要

図 3.2.1 に電力測定実験に使用する装置の図を示す．図中の赤矢印はトラフィックの流れ．青い太線は電源ライン，青い細線は LAN 回線である．今回は高速ネットワークを対象としているため，1000Mbps 以上の速度を持つインターフェースに限って測定を行うものとした．測定に際しスイッチは二つ同時に接続するが，実際に電力測定を行うのは図中の赤い線で囲まれた側のスイッチである．供給される電力はタップ型の電力測定装置 Dominion PX8 の MIB 変数 (1W スケール) に値として保存されるため，別のコンピュータから SNMP を用いて取得する．実験では 1 秒毎にデータの取得を行い，100 秒間を 1 セットとし，35 セット行うものとした．測定項目は，待機時消費電力の他，ポートの Link up 数による電力の違いについても計測した．またトラフィックジェネレータを用いてダミーデータを送信することで，トラヒック負荷をかけた際の電力変動についても計測した．これらの調査は全てのインターフェースの組み合わせで実施した．これらのデータからそれぞれの平均値を算出し，グラフ化して解析を試みた．

3.3 実験結果と考察

3.3.1 Link up ポート数と消費電力の関係

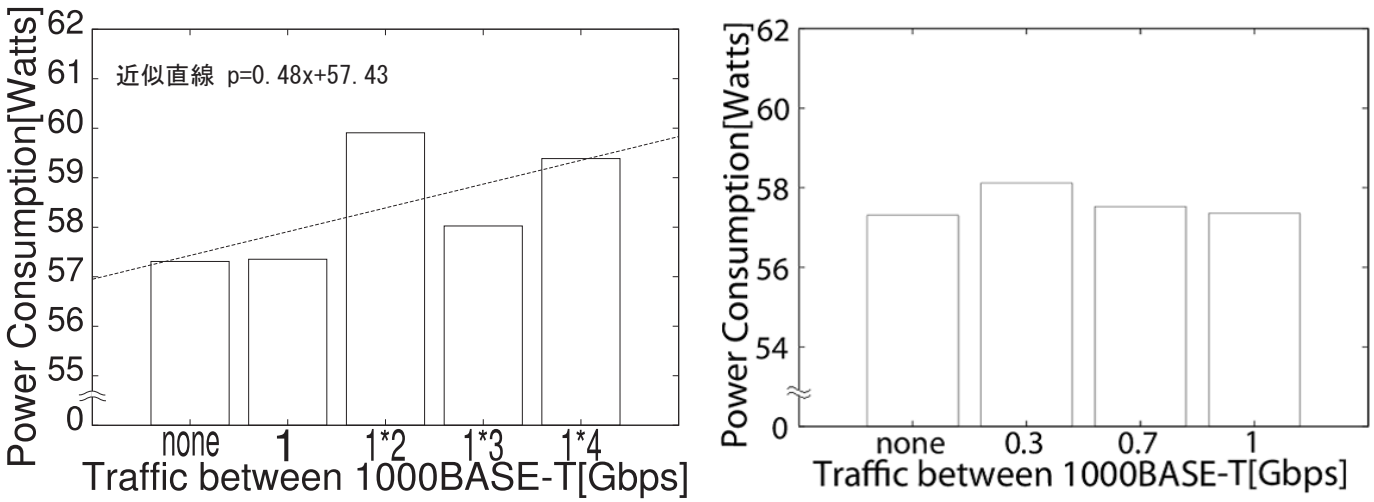


(a)1000BASE-T の消費電力　　(b)1000BASE-SX,10G-BASE-LR の消費電力

図 3.3.1: DGS-3426 の消費電力 (Link up ポート数)

図 3.3.1(a) に，DGS-3426 における 1000BASE-T インターフェースのアクティブ状態 (Link up) ポート数と消費電力の関係グラフを示す．グラフの y 軸は消費電力，x 軸は Link up しているポート数を表すが，”none”とは Link up しているポートがない状態，基礎消費電力を表しており 52W となっている．それぞれの値を最小二乗法で近似したものが p であり，グラフ中に近似直線として記載されている．線はポート数に比例するように上昇しており，一つのポートにつき，0.82W の消費電力が必要となることが分かる．この結果より，1000BASE-T インターフェースのポートが Link up するとより多くの電力が必要になることがわかる．一方図 3.3.1(b) は DGS-3426 に搭載されている光インターフェースの消費電力を計測したものである．先とどうように”none”が基礎消費電力，図中”LR”は 10G-BASE-LR を，”SX”は 1000BASE-SX を示し，”*1”は Link up させたポートの数を表している．図を見てもわかるとおり，これらのインターフェースではポートをリンクアップさせても消費電力が増えることはなかった．よって DGS-3426 において，Link up 消費電力がかかるのは 1000BASE-T のインターフェースだけといえる．この実験に伴い，各 GBIC の消費電力の測定も行っが，光インターフェースのトランシーバである SFP,XFP は増設された時点で電力が増えることが確認された．増設されたあとは，使われているいないにかかわらず電力が変化しないと考えられる．なお本研究において GBIC はすべて増設済みであるものと仮定するため，この消費電力は考慮しないものとする．

3.3.2　トラフィックと消費電力の関係



1000BASE-T の消費電力　　1000BASE-SX,10G-BASE-LR の消費電力

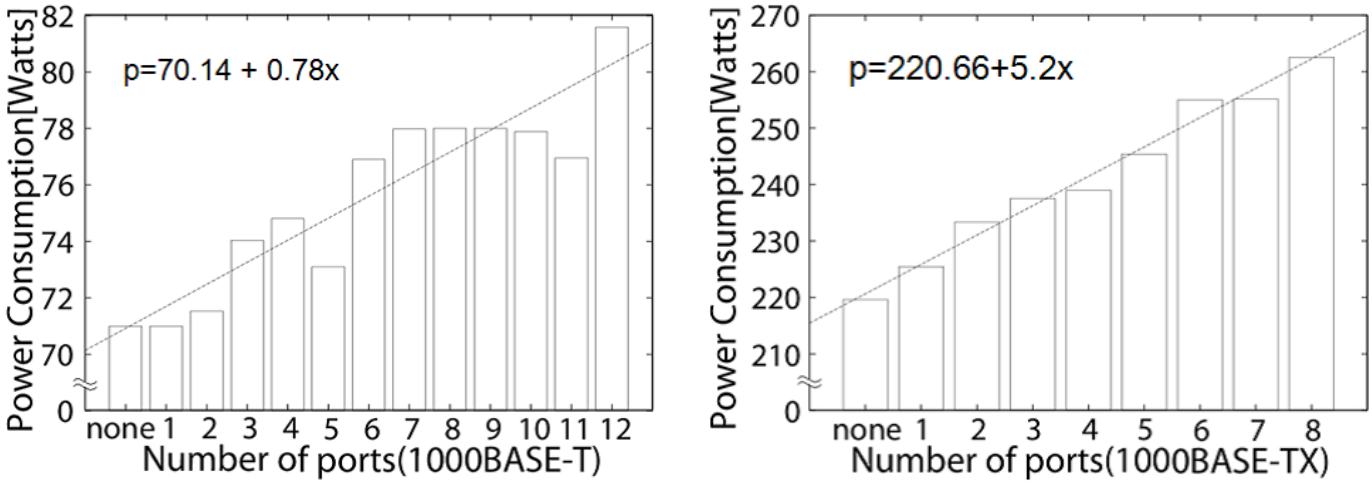
図 3.3.2: DGS-3426 の消費電力 (トラフィック)

図 3.3.2(a) は DGS-3426 の 1000BASE-T インターフェースのポートにトラフィック負荷をかけ，電力の変化を測定したものである．図中”none”は負荷をかけていない状態を示している．あらかじめ 4 つのポートを Link up させているため，若干の誤差はあるものの，先の図 3.3.1(a) の”4”とほぼ同じ値を示している．ここで”1”は 1 つのポートに 1Gbps の負荷を，”1*2”は 2 つのポートにそれぞれ 1Gbps の負荷を与えたことを示している．近似直線 y では，基礎消費電力 57W に加え，1 ポートあたりの増加電力が 0.48 となっている．よって 1000BASE-T では多数のポートに大きな負荷がかかった場合，電力の上昇がみられる．一方，このほかの 1000BASE-SX，10G-BASE-LR については，組合せのいかんにかかわらず消費電力の上昇は見られなかった．また 1000BASE-T における電力上昇も，回線の帯域を完全に使いきるようなトラフィックを流した場合のみであり，現実にそのような状況が長時間起こることには疑問が残る．図 3.3.2(b) は図 3.3.2(a) と同条件にて一つの 1000BASE-T インターフェースポートに 1Gbps 未満のトラフィック負荷をかけたものであるが電力上昇はほとんど見られない．この結論から，トラフィック負荷による電力上昇は，長時間にわたる計測ではほとんど影響せず，無視されても問題がないものであるいうことが言える．

3.4 結論と消費電力のモデル化

3.4.1 別の機器の結果と結論

ここまで DGS-3426 を用いて実際の消費電力測定を行ってきた．ポートをアクティブな Link up 状態にして測定した実験では，1000BASE-T のみで電力の上昇がみられた．またトラフィック負荷をかけて測定した実験においては，大きな負荷をかければわずかな電力が上昇するものの，通常の通信では 1Gbps を常に使いきることは少なく，無視しても問題ない電力であると結論付けた．また DGS-3426 の他，DGS-3450，Alpine3804 という 2 つの機器についても同様の測定を行ったところ，1000BASE-T，1000BASE-TX にのみ電力の上昇が見られた．



DGS-3450 の消費電力 (1000BASE-T)　　Alpine3804 の消費電力 (1000BASE-TX)

図 3.4.1: ネットワーク機器の消費電力 (Link up ポート数)

図 3.4.1(a) は DGS-3450 の 1000BASE-T インターフェースの消費電力を示している．基礎消費電力”none”はおよそ 71W．最小二乗法による近似 p ではポート毎に 0.78W の上昇がみられた．また図 3.4.1(b) は Alpine3804 の 1000BASE-TX インターフェースで測定したものである．基礎消費電力は 220W，近似式の結果より，ポート毎 5.2W の消費電力増加があることがわかる．両者とも，Link up で増加したのは 1000BASE-T/TX インターフェースのみであり，1000BASE-SX および 10G-BASE-LR での電力増加は見られなかった．こちらも DGS-3426 と同様に GBIC 増設時の増加が多くみられた．トラフィック負荷についても同様であり，1000BASE-T，1000BASE-TX ともにわずかな増加がみられたが，1Gbps 以下のトラフィックではほとんど増加がみられなかった．また，光インターフェースではまったく変化が見られなかった．これまでの結果から，シミュレーションパラメータに必要な情報は，基礎消費電力と 1000BASE-T および 1000BASE-TX の消費電力増加量であると結論付けた．トラフィック負荷による増加量は，流れるデータの量と比べると極めて小さく，また常に帯域を使いきるようなトラフィックが発生していることは稀なため，本研究においては考えないものとした．

3.4.2　消費電力のモデル化

表 3.4.1: ネットワーク機器の消費電力モデル

番号	機器名	消費電力モデル	ポート
(a)	DGS-3426	52+0.82x	1000BASE-T
(b)	DGS-3450	70+0.78x	1000BASE-T
(c)	Alpine 3804	221+5.2x	1000BASE-TX

これまでの実験結果より，シミュレーションに用いる消費電力モデルを作成する．表 3.4.1 は先の測定結果をまとめ，基礎消費電力とポートが link up した際に増える電力を実際にモデル化したものである．このとき消費電力モデルは，各々の機器で測定された電力の平均値を最小二乗法で一時近似したものである．ここで x はポート数を表しており，係数は 1 ポートあたりの消費電力である．また $(x = 0)$ のとき，一つのポートも Link up していない基礎消費電力を示している．

3.5　おわりに

本章ではシミュレーションに使用する消費電力パラメータを決定するため，最小二乗法を用いた消費電力情報のモデル化を行った．結果として 1000BASE-T に代表される，より対線使用のポート数によってのみ消費電力が変動することが確認された．光通信インターフェースやトラフィック流量の変化では変動は見られなかった．このモデルデータを用いて 4 章にて最適化手法の提案を行い，5 章では実際にシミュレーションを行っていく．

第4章　消費電力の最適化手法の提案

4.1　はじめに

本項では2章にて既存手法としてEEEとECO-RPの解説を，3章ではシミュレーションに使用する電力モデルの作成を行ってきた．EEEはポートごとの削減しかできないため削減率が低く，削減率の改善を図ったECO-RPは経路選択時に機器ごとの消費電力を考慮できないという問題があった．本章ではこの問題点を解決するための最適化手法として，DCP(Define cost with Performance)，DCN(Define cost with Neighbors)，DCNP(Define cost with neighbors and Performance)の三つの手法を提案する．

4.2　提案手法の位置づけ

ECO-RPも提案手法も，最適化にDijkstra法を使うということはかわらない．重要なのはECO-RPには動的OSPFコスト更新があり設定される変数elの値によって変動幅が変わるということである．

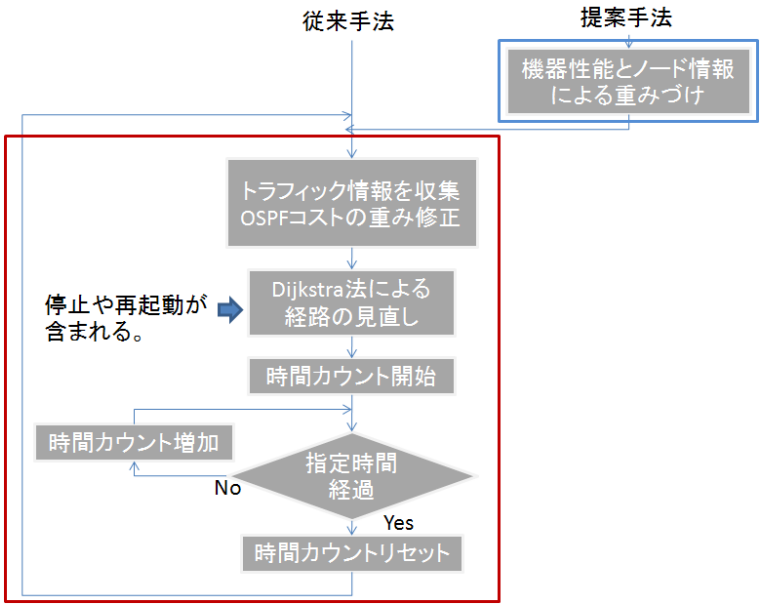


図 4.2.1: 従来手法と提案手法

図4.2.1は本研究における提案手法の位置づけを示したフローチャートである．赤線で囲まれた部分はECO-RPがすでに実装している部分，青線で囲まれた部分が，本研究で提案しようとしている部分である．本研究の目的は，機器の消費電力や情報を考慮することで適切な初期コスト設定を行おうというものである．単体でのネットワーク最適化も行えるほか，ECO-RPが実装しているような動的OSPFコスト更新アルゴリズムと連携して動くことで，常に最適性を保ち続けること想定している．

4.3 DCP(Define cost with Performance)

2 章で定義した問題点の中に，ECO-RP をはじめとする従来手法はルータの消費電力を考慮していないという問題があった．そのため OSPF コストを変動させてネットワーク経路を常に最適化したとしても，大きな電力を使用しているルータが削減できない可能性が残されている．そこで経路を決定する OSPF コストに，消費電力の概念を与えようというのが DCP の考え方である．消費電力の高い機器を避けるように OSPF コスト設定ができれば，除外されるべき機器にはトラフィックが流れなくなり，従来手法よりも効果的に電源を停止することができるようになるはずである．

4.3.1 DCP の定義式

あるネットワークに設定される初期 OSPF コストを，モデル化した電力情報を用いて最適化することを考える．今，ルータの基礎消費電力を W と表記する．互いに接続されるノード A の消費電力を W_a，ノード B の消費電力を W_b として，初期 OSPF コストの最適化係数 ρ の式を定義する．

$$\rho = (W_a + W_b)/2 \tag{4.3.1}$$

ρ はノード A，B の消費電力の平均をとったものになるが，両者の電力が大きければ大きいほど，ρ もまた大きな値になる．ここでは 2 章で定義した OSPF コストの導出式 (1) を用いて I を求める．この時，消費電力最適化コスト L は以下のようになる．

$$L = \rho \cdot I \tag{4.3.2}$$

初期 OSPF コスト I に係数 ρ をかけ合わせることで，消費電力の大きいノード間ではコストが大きくなり，小さいノード間では小さくなる．この結果，消費電力の大きいノードは経路から外されるため，ネットワークには消費電力の小さいノードだけが残る．以下に詳細なアルゴリズムを示す．

```
1  sub DCP(){
2    for($i=0;$i<$max;$i++){
3      for($j=$i+1;$j<$max;$j++){
4        if($dist[$i][$j]!=INF){
5          $dist[$j][$i]=$dist[$i][$j]*=($device_w[$node_type[$i]]+
6            $device_w[$node_type[$j]])/2;
7          if($dist[$i][$j]<1){
8            $dist[$i][$j]=$dist[$j][$i]=1;
9          }
10         $dist[$j][$i]=$dist[$i][$j]=int($dist[$i][$j]);
11       }
12     }
13   }
14 }
```

sub DCP は関数であり，呼び出された際に DCP によるコスト変動を行うアルゴリズムである．ここで i,j はノードの番号を示している．このアルゴリズムにおいて，それぞれの数値は $(i < j)$ であり，等しくはならない．また$dist[$i][$j] は OSPF コストを格納する配列であり，この場合は i 番目のノードと j 番目のノードを接続する回線のコストを示している．このプログラムにおける”INF”とは無限大を表す定数であり，距離コストが測れない状態，回線が物理的につながっていない状態を表している．また $device_w はネットワーク中に存在する機器の種類と基礎消費電力を関連付けて記録している配列変数である．$device_w[$node_type[$i] は i 番目のノードの基礎消費電力を返す．このほか 5，6 行目に DCP の定義式が出てきており，処理の終わりには導出された OSPF コストが 1 以下である場合に 1 に補正，少数を切り捨てるための整数化が行われている．次にこのアルゴリズム用い，実際にネットワーク構造の最適化を行ってみることにする．

4.3.2　DCP によるネットワークの最適化

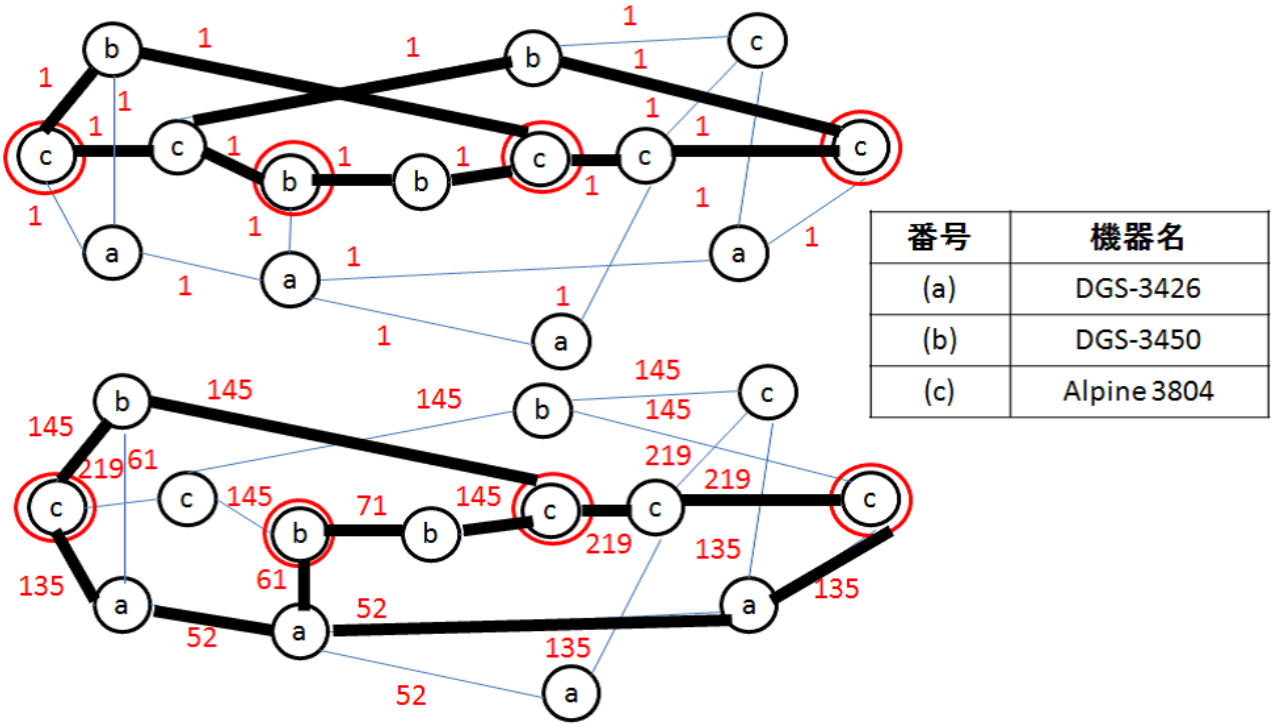


図 4.3.1: DCP による初期コスト更新と最適化

図 4.3.1 はネットワークの一例に対し，実際に DCP による OSPF コストの最適化を行ったものである．上側が元のネットワーク，下が DCP を適用したもので，英字と機器の対応は右の表のとおりである．このうち赤丸のノードがエッジルータ．黒線は Dijkstra 法で選択された経路となり，DCP が使われない場合は上側に偏っており，DCP 使用時には下側に偏る構成となっている．

表 4.3.1 は図 4.3.1 におけるそれぞれのネットワークの消費電力と，全体を 100 パーセントとした際の

表 4.3.1: OSPF と DCP の最適化電力

	消費電力(Watts)	消費電力(%)	ノード・リンク削減数
全体	1931.22	100	
OSPF	1453.04	75.24	5ノード・11リンク
DCP	1308.04	67.73	4ノード・10リンク

削減割合，削減ノードとリンク数を示したものである．DCP を用いない上側のネットワークを OSPF とし，用いる側を DCP で記している．ノード・リンク削減数はどちらも 5 ノード 10 リンク程度とほとんど違いがないが，消費電力の項において，OSPF 側は 75 パーセント程度まで削減できたのに対し，DCP は 67 パーセントと 10 パーセントの差，電力的には 150W 近い差が生まれている．これは，通常の OSPF コストの場合，ネットワークにおける電力的なボトルネックである (c) のノードを経路選択から除くことができなかったのに対し，DCP を使用したことで c を除外し，より消費電力の少ない a を含む経路を選択するようになったためである．この結果より，DCP を用いることで消費電力の高いノードを避ける経路を設定することができ，より省電力なネットワークを構築できるといえる．

4.3.3 DCP の問題点

DCP を用いると消費電力の大きいルータを避けるような OSPF コスト設定ができることは先に述べた．しかしながらこの手法は隣り合う機器の消費電力の平均を指針とするため，個々の機器の電力情報を厳密に考慮しているとはいえない．隣り合うノードの平均値を取るため，どちらか一方の消費電力が高くても片方の電力が低ければ，経路として選択されてしまい，不適切な機器がネットワーク上に存在してしまう場合がある．たとえばノード A の消費電力 $W_a = 50$，ノード B の消費電力 $W_b = 50$ の場合と，$W_a = 10$，$W_b = 90$ の場合，互いのノードを接続する回線の OSPF コストは同じになる．これによって，本来経路から除外されるべきノードが選択されてしまうこともある．また DCP の経路集約によって処理性能の劣る機器に過剰なトラフィック集中が起こってしまう可能性も考えられる．DCP が考慮するのは消費電力だけであり，処理性能や動作状況は一切考慮されない．よって，経路上に存在する重要度や依存度の高いコアルータも他と同様に経路から外してしまう．この結果，ネットワーク自体が非常に不安定になるという問題も残されている．さらに DCP を，ECO-RP のような動的 OSPF コスト更新アルゴリズムと一緒に使用する場合，消費電力をそのままコストに掛け合わせるという性質上，導出される値の差が巨大なものになり，コスト更新による変動幅ではネットワークの経路が変化しなくなる可能性も残されている．本研究ではこれらの問題点のうち，不適切な機器が経路に選ばれる問題，およびコストが巨大になる問題を考慮するため，さらに別の手法として隣接機器の情報を考慮した OSPF コストの最適化手法を定義することにした．性能の低い機器に対するトラフィック集中の問題については電力情報以外の処理性能を考慮すれば対処することは可能であり，重要度の高いルータに対しても専用の重みづけコストを用いるなどすれば対処は可能だが，ここでは消費電力情報を用いた OSPF コストの更新アルゴリズムを提案することを目的としているため，将来の課題として残すこととする．

4.4 DCN(Define cost with Neighbors)

先に定義した DCP は，OSPF コストを最適化する際にノードの消費電力の平均値を用いるため，状況によってはふさわしくない機器が選ばれてしまう可能性があった．そこで本項では先の問題を解決する手法として，Define cost with Neighbors(DCN) を提案する．DCN は隣接機器の情報を考慮した OSPF コストの最適化手法であり，機器の種類や隣り合う機器の組み合わせによってコストの変動値を設定することによって，経路選択を最適化するものである．

4.4.1 DCN の定義

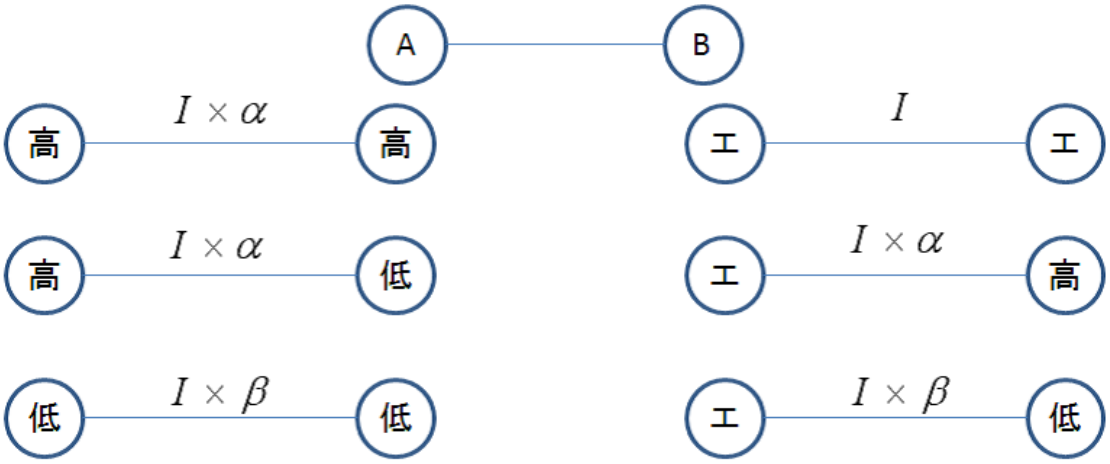


図 4.4.1: DCN の概念

図 4.4.1 は DCN の概念を表したものである．あるノード A と B が隣り合って接続されている場合，機器の消費電力を高と低にわけ，その組み合わせによって適切な係数を掛け合わせることで，OSPF コストの調節を図っている．ここで α はコストを増加させるための係数，β はコストを減少させるための係数である．これらを初期 OSPF コスト I と掛け合わせることで，最適なコストの導出を行う．

図 4.4.2 は，概念をフローチャート化したものであり．消費電力が小さいノードが，同様に小さいノードと隣り合っている場合，またはエッジノードと隣り合う場合のみコストを減少させ，それ以外の組み合わせではコストを増加させている．これにより，消費電力の小さいノード間が小さな OSPF コストを持つことになり，トラフィックの集約とネットワークの省電力化を行うことができる．さらに DCN は 2 章で問題点として触れたエッジノードが隣り合う問題にも配慮している．図中”エ”のノードはエッジノードを示しており，互いに隣り合っている場合は初期 OSPF コストを変化させないことで低く保つと同時に，このコストを手動で高く設定することにより，意図的に迂回させることもできるよう任意性を与えた．

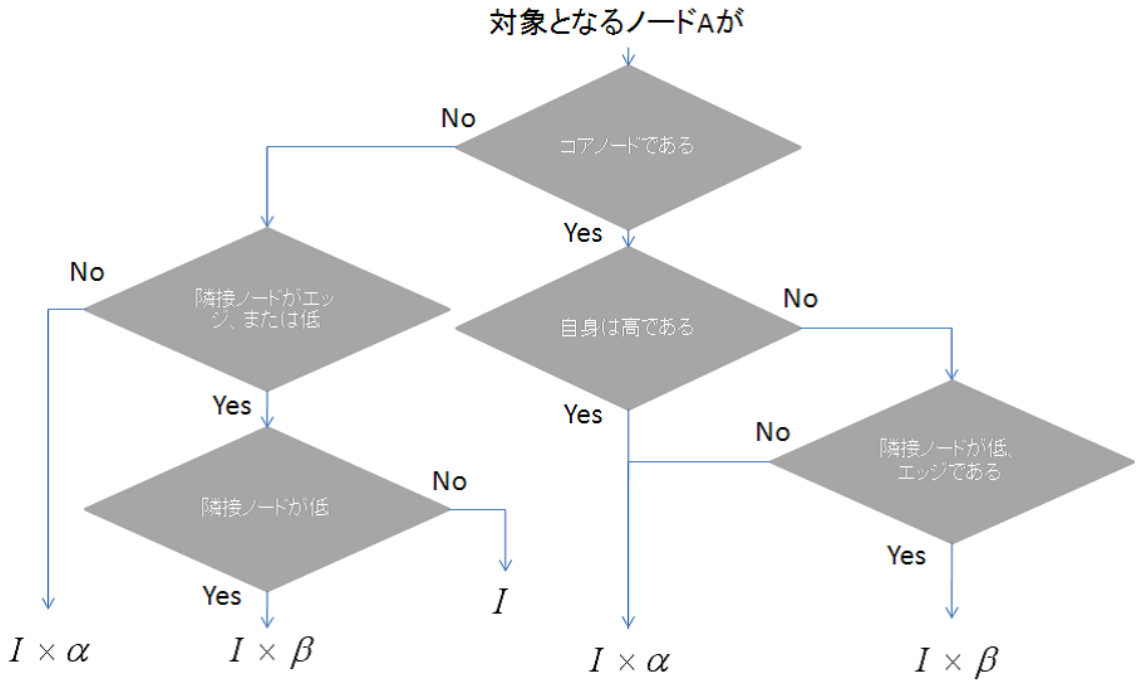


図 4.4.2: DCN の概念 (フローチャート)

4.4.2 基準値の定義

DCN のアルゴリズムを動かすためには，機器の消費電力を何らかの方法で高と低に分けなければならない．このときどこで基準線を引くかが問題となる．またこの基準は絶対的なものではなく，ネットワークを構成する機器によって相対的に決まらなければならない．そこで本研究では，シミュレーションに使用する機器の電力を平均したものを基準値として用いることにした．N 個のノードからなるネットワークがあるとする．今，機器の消費電力を $W_i(0 \leq i < N)$ であらわすとき，基準値 $Base$ を決める式は次のようになる．

$$Base = (\sum_{i=0}^{N-1} W_i)/N \tag{4.4.1}$$

今，消費電力モデルが表 3.4.1 に従うとき，$Base = 114$ となる．この値はネットワークを構成する機器の消費電力によって相対的に変動するため，同程度の消費電力を持つ機器が偏在する環境であっても適切な効果が期待できる．次に，これらの式を実装したアルゴリズムを記載する．

4.4.3 DCN のアルゴリズム

```
1  sub DCN(){
2    for($i=0;$i<$max;$i++){
3      for($j=$i+1;$j<$max;$j++){
4        if($dist[$i][$j]!=INF){
5          if(!$edge[$i]){
6            if($device_w[$node_type[$i]]<$Base &&
7              ($device_w[$node_type[$j]]<$Base || $edge[$j])){
8              $dist[$j][$i]=$dist[$i][$j]*=β;
9            }else{
10             $dist[$j][$i]=$dist[$i][$j]*=α;
11           }
12         }else{
13           if($device_w[$node_type[$j]]<$Base || $edge[$j]){
14             if(!$edge[$j]){
15               $dist[$j][$i]=$dist[$i][$j]*=β;
16             }
17           }else{
18             $dist[$j][$i]=$dist[$i][$j]*=α;
19           }
20         }
21         if($dist[$i][$j]<1){
22           $dist[$i][$j]=$dist[$j][$i]=1;
23         }
24         $dist[$j][$i]=$dist[$i][$j]=int($dist[$i][$j]);
25       }
26     }
27   }
28 }
```

ここでは DCN() が初期 OSPF コストの変更を行う関数である．DCP と同じく i,j はノードの番号を示しており，$dist もコストを格納する配列である．ただし本アルゴリズムには，DCP にはない隣接ノードを考慮する仕組みが含まれている．特に重要なのはエッジノードが隣り合う際に補正をかけないことであり，15 行目に実装されている．新しく定義された変数は$edge[$i] で，i 番目のノードがエッジノードである場合には 1 を，ない場合には 0 を格納する配列である．$node_type[$i] は i 番目のノードの種類を格納している配列であり，$device_w[$node_type[$i]] は i 番目のノードの基礎消費電力を返す．また$Base は先に求めた相対的な基準値である．これらを用いて条件分けを行った後，係数 α, β を用いてコストを変更している．次にこのアルゴリズムを用いてネットワークの最適化を行った．

4.4.4　DCN によるネットワークの最適化

DCN を用いるためには係数 α,β の値を決定する必要がある．しかしながらこれを決定するためには，実際にシミュレーションでネットワークの最適化を行う必要があるため，先に最適化の概要を解説することとした．

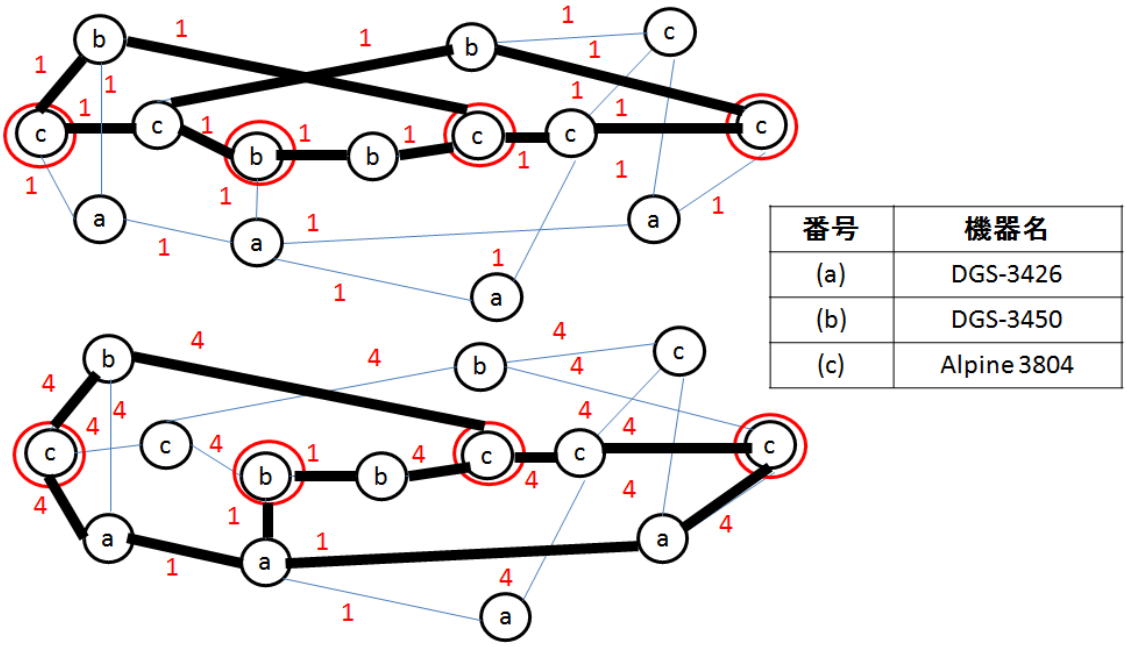


図 4.4.3: DCN による初期コスト更新と最適化

図 4.4.3 は実際に OSPF コストを最適化したものである．上側が元のネットワーク，下が DCN を用いたものになっている．今回はすべての OSPF コストが 1 のネットワークを例とするため，暫定的に $\alpha = 4$,$\beta = 1$ を設定した．元ネットワークが上側に偏っているのに対し，DCN は DCP の結果と同様に下側に偏っている．これは消費電力の高い機器である (c)Alpine3804 に接続されている回線の OSPF コストが α 倍された結果，一部が経路に含まれなくなったためである．

表 4.4.1: OSPF と DCN の最適化電力

	消費電力(Watts)	消費電力(%)	ノード・リンク削減数
全体	1931.22	100	
OSPF	1453.04	75.24	5ノード・11リンク
DCN	1308.04	67.73	4ノード・10リンク

表 4.4.1 はそれぞれの削減率を比較したものである．OSPF はコストに手を加えなかったものであり，およそ 75 パーセントにまで削減ができた．DCN は隣接情報を考慮して最適化を行ったものであり，こ

ちらの削減率は67パーセントである．これはDCPによる削減の結果と同じである．リンク数のノード数も同様であり，どちらも5ノード10リンク程度であるのに対し，電力はおよそ8パーセント，150Wの差がついている．これは消費電力の高い機器間のOSPFコストが高くなるように修正されたため，より電力の低い機器へとトラフィックが流れるようになったためである．この結果より，隣接情報を考慮するDCNはDCPと同様の削減効果が見込めるといえる．

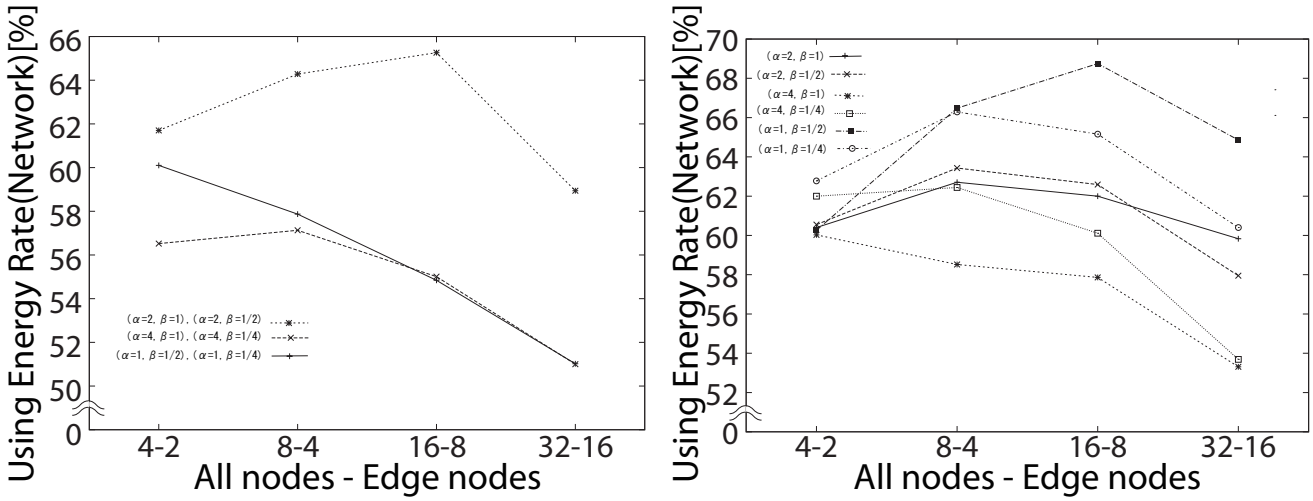
4.4.5 係数α,βの決定

DCNを用いるためには，OSPFコスト修正の係数となる$\alpha(1 \leq \alpha \leq X, X$は正の実数$)$と$\beta(0 < \beta \leq 1)$を決定しなければならない．DCNのネットワーク最適化の項でも述べたが，これを一意に決定するのは難しいことである．そこで先にもおこなったDCNによるネットワーク最適化をプログラムに実装し，実際にシミュレーションをおこなうことで，どのパラメータがもっともふさわしい削減率になるかを比較することにした．このとき，各種パラメータに大きすぎる値を利用することは避けなければならない．初期OSPFコストは人の手によってつけられることもあり，導出式を用いた場合でも，設定によっては大きな値が付くことがある．この結果として，それぞれの差がECO-RPなどの動的更新でプラスマイナスに変動する数値の範囲を超えてしまう，あるいはコスト更新をした場合，OSPFコストの定義域から外れてしまうという可能性も残されている．よってこの場合，係数α,βに設定される数値はできるだけ小さいほうがよい．そこで本実験では$(1 \leq \alpha \leq 4)$かつ$(0 < \beta \leq 1)$の範囲で数値を定義することにした．まずperlを用いてDijkstra法を実装し，次にDCNのシミュレーション実装を行う．

表 4.4.2: シミュレーション条件

OS	Ubuntu9.10(Linux 2.6.31-22-generic)
CPU	Intel Celeron Processor SU2300
使用言語	Perl
ネットワーク生成(ノードの種類選択)	乱数(rand関数)で選択
ネットワーク生成(回線の生成)	rand() mod 2を使用、奇数偶数で有無を決定
シミュレーション実行回数	500
備考	ノードの孤立時に計算破棄、ネットワーク再構築

表4.4.2はシミュレーションの条件を示したものである．シミュレーションに使用したマシンはUbuntu9.10，Linuxカーネルは2.6.31-22-genericである．今回は構築にperl言語を使用．全体ノードとコアノード数を入力し，機器の種類とネットワークの構造，回線生成は乱数を用いて行うものとした．生成時にエッジノードが孤立した場合には，結果を破棄して最初からやり直す方式をとっている．実験では係数をパラメータとして入力し，それぞれ500回行い，平均を求めて比較を行った．

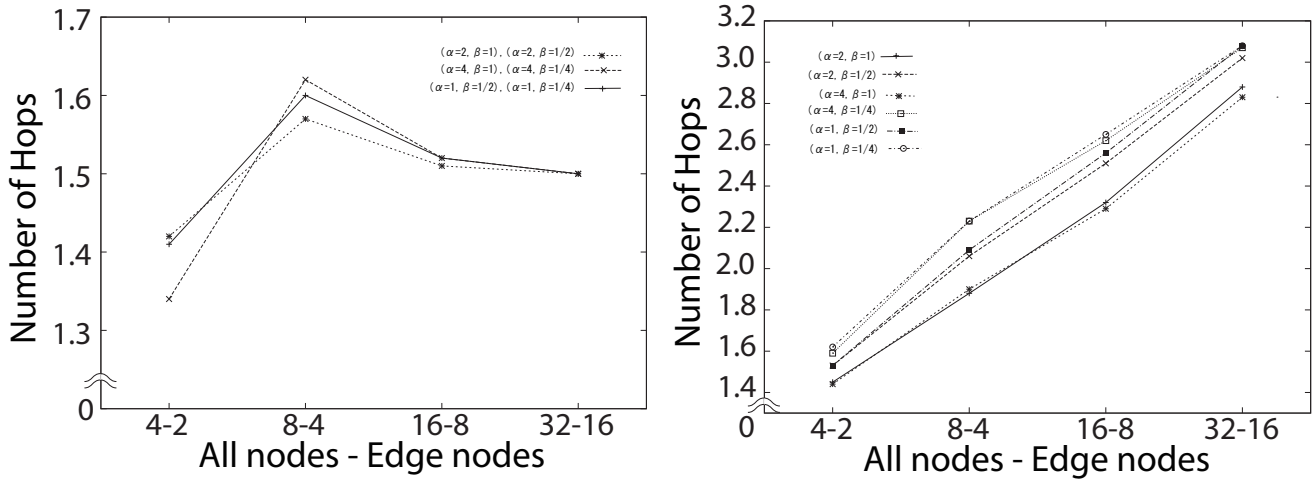


(a) 均一な初期コスト設定 (全て 1)　　(b) 乱数による初期コスト設定

図 4.4.4: 係数とそれぞれの消費電力平均

図 4.4.4 は実際に作成したシミュレータで実験した結果，得られた平均消費電力ぱ一センテージである．y 軸は消費電力．x 軸は全体ノード数とエッジノードの数を示しており，”4-2” とは全体ノード 4 つに対し，エッジノードが 2 つという意味である．また図 4.4.4(a) は OSPF コストの導出式にて初期コストを設定したものですべてのリンクコストは 1 になっており，図は 1 から 100 の乱数を用いて初期コストを設定した．(a) では $(\alpha = 1, \beta = 1/2, 1/4)$ のが消費電力が高く，$(\alpha = 4, \beta = 1, 1/4)$ がもっとも低くなっている．これは OSPF コストに 1 以下の数を設定することができず，少数になった結果がすべて 1 に整形されたため，$(\alpha = 4, \beta = 1, 1/4)$ では補正係数を使用しない場合と一致したためである．この結果より，OSPF コストが均一な環境においては，コストを増加させる係数 α の値が大きいほうが，より高い削減率を誇るといえる．

一方乱数で初期コストを決定した図 4.4.4(b) では，初期コストが均一のものにくらべて消費電力が高くなる傾向がみられたが，最良のパラメータについてはほぼ同じ結果が見られた．全体ノード数が多くなると，削減が難しくなるということを考慮した場合，もっともすぐれた結果を残したのは $(\alpha = 4, \beta = 1/4)$ である．もっとも結果が悪かったのは，先と同様に $(\alpha = 1, \beta = 1/2)$ であった．しかし消費電力の結果だけで，すぐれた係数を決定することには疑問が残る．そこで別の指針としてエッジノード間のホップ数を採用し，これを比較することにした．



(a) 均一な初期コスト設定 (全て 1)　　(b) 乱数による初期コスト設定

図 4.4.5: 係数とそれぞれの平均ホップ数

図 4.4.5 のうち (a) は，初期コストが均一であるネットワークのホップ数を示したものである．(b) は 1 から 100 の乱数を用いて初期コストを設定した際の平均ホップ数を示している．ここで y 軸がホップ数，x 軸は全体ノードとエッジノードの表記である．(a) ではどのグラフもほぼ同じ点を通っており，最終的に一つの値へと収束している．どの方式もほとんど差はなく，差も 0.1 以下であった．しかし (b) のほうでは明らかな差が見られた．(b) 図中”32-16”の時点で，最悪の数値は 3，パラメータは $(\alpha = 1, \beta = 1/2)$ である．一方，最良の数値は $(\alpha = 4, \beta = 1)$ の時の 2.6 であり，0.4 の差が生まれている．この結果より，乱数をコストに用いたネットワークのほうが，ホップ数的に最短経路を通ることが難しく，また先の結果より電力削減も難しい傾向にあるということが言える．また DCN を用いる場合 $(\alpha = 4, \beta = 1)$ の係数を使用することで，定義された数値の範囲内ではすぐれた削減が行えることがわかった．

4.4.6　DCN の問題点

DCN では α と β を補正係数として用いるが，ECO-RP などの OSPF コスト動的更新アルゴリズムの動作を阻害するという懸念から，大きな値を使えないという問題がある．先の実験は小さな数値の範囲で最適なものを選択したわけだが，制限を超えて定義できる自然数と比較した場合，削減率が最適であるという保証はされていない．実際にどの程度の値を入れた場合に問題が起こるのかを確かめるためには実験を行わなければならないが，係数の値が小さいほうがより安全であることは間違いない．この問題は DCP でも発見された問題であり，消費電力をそのままコストに乗算するため，状況によってはコスト値の最大を超えてしまう可能性が残されている．ECO-RP 等の OSPF コストの動的更新アルゴリズムと連携して動作することを考えた場合，個々の数値が巨大になり，差が大きくなりすぎると効果が失われてしまう．これらの問題を解決するためには，可能な限り一般的な OSPF コストに近い数値で，なおかつ係数が必要とされないであろう手法が必要になる．そこで本研究では，先に定義した DCP および DCN の概念を用いて，残された問題を解決するための手法，DCNP を提案することにした．

4.5 DCNP(Define cost with Neighbors and Performance)

先に定義された DCP には，隣接機器が考慮できない問題．電力の平均値を用いるため，適切でない経路が選択されてしまう問題に加え，導出される最適化コストの値が大きくなりすぎ，ECO-RP 等の OSPF コスト動的更新プログラムの動作を阻害してしまう恐れがあった．また DCN も最適化コストの値を小さく抑えるため，小さな係数しか使えないという問題を抱えており，係数による最適性は保証されていない．この問題を解決するため，本項では機器の電力情報，隣接情報の両者を用いた新しい手法，Define cost with Neighbors and Performance(DCNP) を提案する．DCNP は導出されるコストの値を低く抑え，なおかつ係数を必要としないことを目標として作成された OSPF コスト最適化アルゴリズムである．

4.5.1 DCNP の定義

DCP では係数に消費電力を用いていたため，最適化コストが増大しやすく，DCN についても係数に大きな値を使用できないため，最適化の性能が低くなるという問題があった．そこで DCNP では導出される OSPF コストを低く保ち，なおかつ消費電力の削減を効果的に行うため，DCP で用いた消費電力のモデルをパーセンテージへと写像し，正規化することにした．今，あるネットワークにおける i 番目の機器の消費電力を W_i，そのネットワーク内にある機器でもっとも高い消費電力の値を W_{max} とする．このとき正規化された値 Wpp_i は次の式で表すことができる．

$$Wpp_i = \frac{W_i}{W_{max}} \cdot 100 (0 < Wpp_i \leq 100) \tag{4.5.1}$$

表 4.5.1: 基礎消費電力モデルの正規化

番号	機器名	電力分布(Wpp)
(a)	DGS-3426	23.53
(b)	DGS-3450	31.68
(c)	Alpine 3804	100

表 4.5.1 はモデル化した基礎消費電力に正規化を行ったものである．0 から 100 の範囲に写像しなおしており，最大値によって相対的な決定を行っている．この式をもとに，OSPF コストの最適化係数 $\$rho$ と，消費電力最適化コスト L を求める式を定義する．今，あるネットワークにおいて隣り合うノード A，ノード B を定義する．ノード A の消費電力を正規化したものが Wpp_a，ノード B が Wpp_b とするとき，それぞれの式は次のようになる．

$$\rho = (Wpp_a + Wpp_b)/2 \tag{4.5.2}$$

$$L = \rho \cdot I \tag{4.5.3}$$

ここで I は初期 OSPF コストを示している．これらの式は DCP の拡張であり，係数 ρ の値も狭まくなるため，結果的に導出される L は DCP のものよりも低い値に収まる．さらに DCNP では，DCP で考慮できなかった隣接機器の情報を考慮するアルゴリズムの実装を試みる．

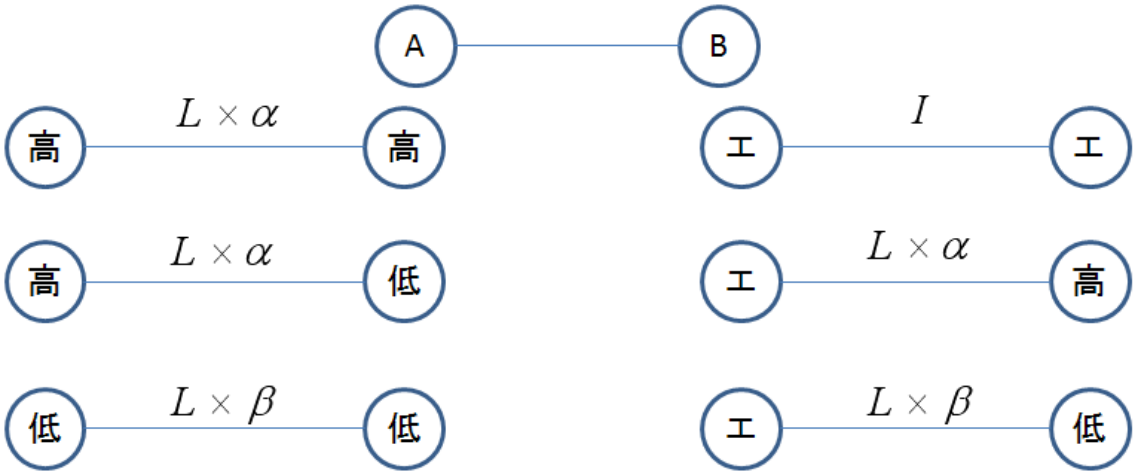


図 4.5.1: DCNP の概念

図 4.5.1 は DCN にも実装した，隣接機器の情報を考慮するアルゴリズムである．あるノード A と B が隣り合っている場合，両者をつなぐリンクのコストの導出法を示している．ここで L は消費電力最適化コスト，I は初期 OSPF コストである．α はコストを増加させる係数で，定義域は $(1 \leq \alpha \leq X)$. ただし X は自然数になる．β はコストを減少させる係数であり，定義域は $(0 < \beta \leq 1)$ である．高は消費電力が高いノード，低は低いノードを表しており，低いノード同士がつながるときにはコストを下げ，それ以外の場合ではコストを上昇させている．また図中”エ”はエッジノードを示しており，エッジノード同士が連なる場合にはコストを初期コスト I のまま変化させないようにしている．この定義は DCN のものと同じである．

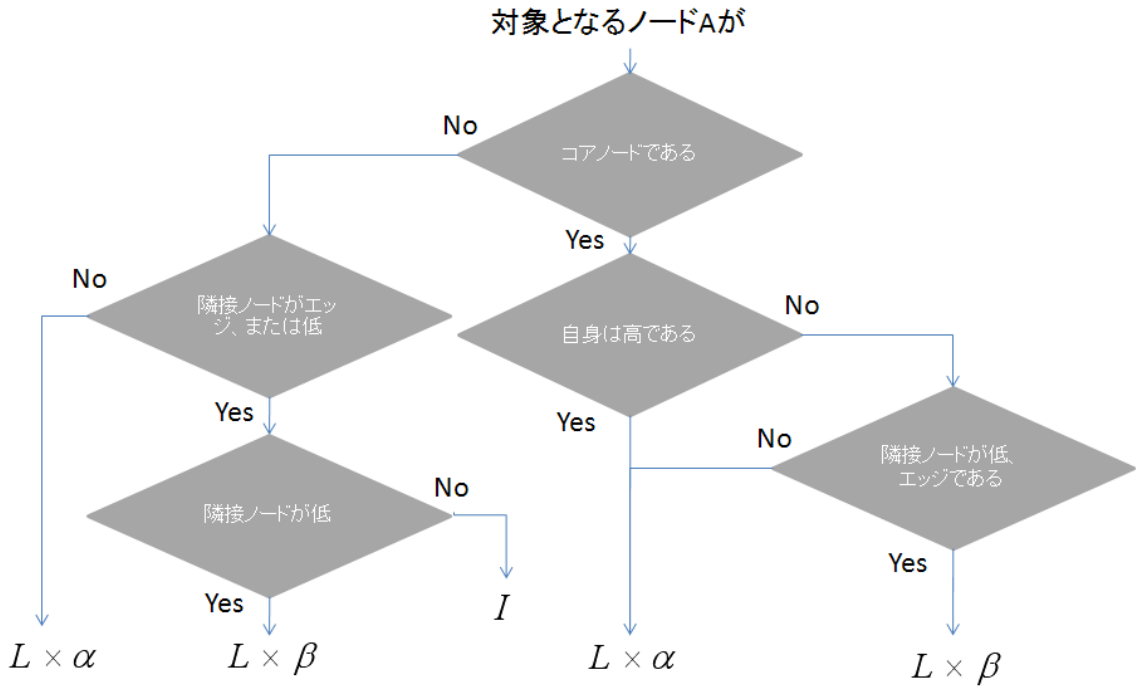


図 4.5.2: DCNP の概念 (フローチャート)

図 4.5.2 は図 4.5.1 の概念をプログラムに実装するため，フローチャートとして表現したものである．対象となるノード A の種類によって，異なる処理を行うように設計されており，実際のシミュレータにもこれと同様のものを実装した．

4.5.2 基準値の定義

DCNP のアルゴリズムを動かすためには，DCN と同様に，機器の消費電力を高と低に分けなければならない．このときどこで基準線を引くかが問題となる．またこの基準は絶対的なものではなく，ネットワークを構成する機器によって相対的に決まらなければならない．DCNP では DCN の定義を引き継ぎ，シミュレーションに使用する機器の電力を平均したものを基準値として用いることにした．今，N 個のノードからなるネットワークがあるとする．今，機器の消費電力を $Wpp_i(0 \leq i < N)$ であらわすとき，基準値 W_{base} を決める式は次のようになる．

$$W_{base} = (\sum_{i=0}^{N-1} Wpp_i)/N \tag{4.5.4}$$

今，消費電力モデルの正規化が表 4.5.1 に従うとき，$W_{base} = 55.05$ となる．次にこれらの定義を実装したアルゴリズムを示す．

4.5.3 DCNP のアルゴリズム

```
1  sub DCNP(){
2    for($i=0;$i<$max;$i++){
3      for($j=$i+1;$j<$max;$j++){
4        if($dist[$i][$j]!=INF){
5          if(!$edge[$i]){
6            $dist[$i][$j]*=($Wpp[$node_type[$i]]+$Wpp[$node_type[$j]])/2;
7            if($Wpp[$node_type[$i]]<$W_base &&
8             ($Wpp[$node_type[$j]]<$W_base || $edge[$j])){
9              $dist[$j][$i]=$dist[$i][$j]*=β;
10           }else{
11             $dist[$j][$i]=$dist[$i][$j]*=α;
12           }
13         }else{
14           if($Wpp[$node_type[$j]]<$W_base || $edge[$j]){
15             if(!$edge[$j]){
16               $dist[$i][$j]*=($wpp[$node_type[$i]]+$Wpp[$node_type[$j]])/2;
17               $dist[$j][$i]=$dist[$i][$j]*=β;
18             }
19           }else{
20           $dist[$i][$j]*=($Wpp[$node_type[$i]]+$Wpp[$node_type[$j]])/2;
21           $dist[$j][$i]=$dist[$i][$j]*=α;
22           }
23         }
24         if($dist[$i][$j]<1){
25         $dist[$i][$j]=$dist[$j][$i]=1;
26         }
27         $dist[$j][$i]=$dist[$i][$j]=int($dist[$i][$j]);
28       }
29     }
30   }
31 }
```

DCNP のアルゴリズムは，先に提案した DCP および DCN が持つ 2 つの概念を組み合わせて構成されている．$dist 等の配列変数は DCP,DCN と同じなため，ここでは先と違う変数について解説していく．まず$Wpp は表 4.5.1 の正規化された基礎消費電力モデルを保持する配列である．アルゴリズム中に記載された$Wpp[$node_type[$i]] は i 番目のノードの基礎消費電力を正規化した数値を格納している配列になる．また$W_base は正規化された基礎消費電力の平均値，DCNP における基準値を表してい

る．本アルゴリズムではDCPやDCNのように消費電力をのそのまま判断基準に用いるのではなく，一度正規化をした値を用いることで，コストの値を極力小さくすることを考慮している．またDCNでも使用した係数α, βを使用することで，機器の種類に応じたOSPFコストの更新を行うことを目的としている．

4.5.4　DCNPによるネットワークの最適化

DCNPの動作を確認するため，実際にネットワークの最適化を試みる．DCNPでOSPFコストを最適化するためには，係数α，βの値を導出する必要がある．これにはシミュレーションが必要となるため，先に係数補正なしのDCNPを用いて動作を確認することとした．設定する係数の値は$\alpha = 1$,$\beta = 1$である．これはエッジノード以外の補正を行わないことを示している．

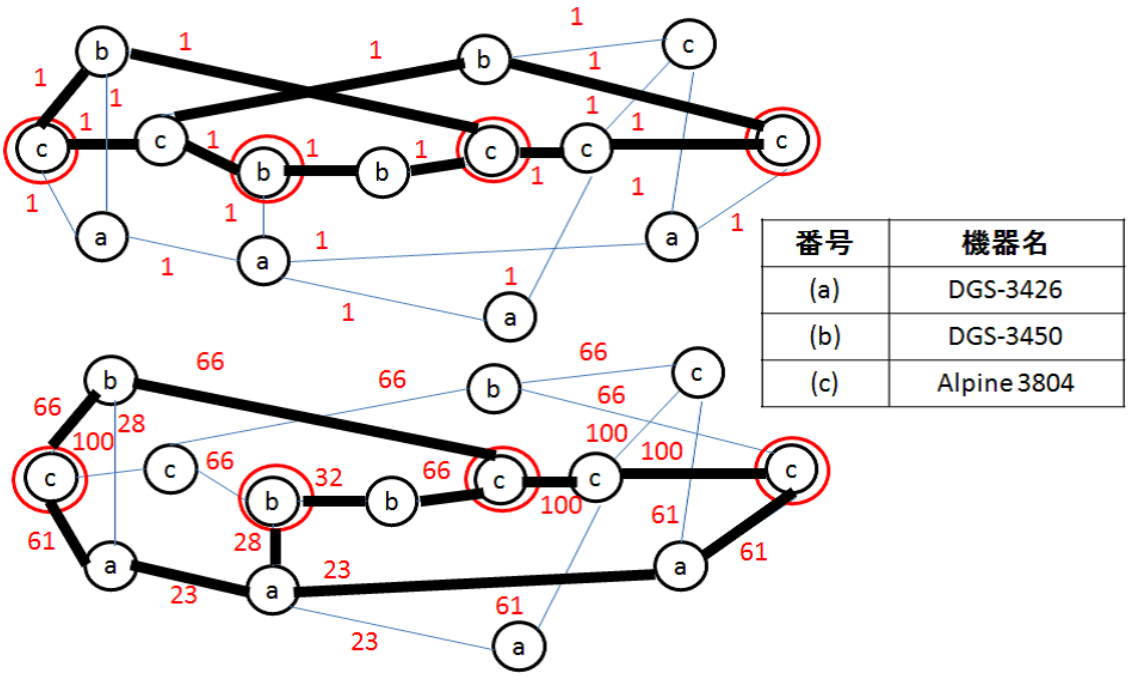


図 4.5.3: DCNPによる初期コスト更新と最適化

表 4.5.2: OSPFとDCNPの最適化電力

	消費電力(Watts)	消費電力(%)	ノード・リンク削減数
全体	1931.22	100	
OSPF	1453.04	75.24	5ノード・11リンク
DCNP	1308.04	67.73	4ノード・10リンク

図 4.5.3 は初期ネットワークおよび DCNP を用いて OSPF コストを定義しなおしたネットワークである．赤丸のノードはエッジルータをしめしており，黒い線が Dijkstra 法によって選択された経路である．初期ネットワークのコストは $C = 1000$ の OSPF コスト導出式を用いたため，すべてのコストが 1 となっている．このネットワークにはエッジノードが隣り合う例はなく，さらに係数による補正が無効となっているため，消費電力最適化コスト L がそのままコストとなっている．削減率は表 4.5.2 に示してあり，全体を 1931.22W としたとき，DCNP は 1308.04W，67.73 パーセントと 32 パーセントの消費電力を削減できたという結果になった．ノードの削減数は 4 ノード・10 リンクとなっており，削減率と削減数ともに DCP および DCN の結果と同じである．しかしながら DCNP は通常の OSPF の結果に勝っており，なおかつ DCP が設定するコストよりも小さな経路コストを設定して削減を行った．また DCN のような係数を使用しない状態で削減を行えたことから，コストを小さくでき，係数を使用しなくても済むという，すぐれた手法であると言える．

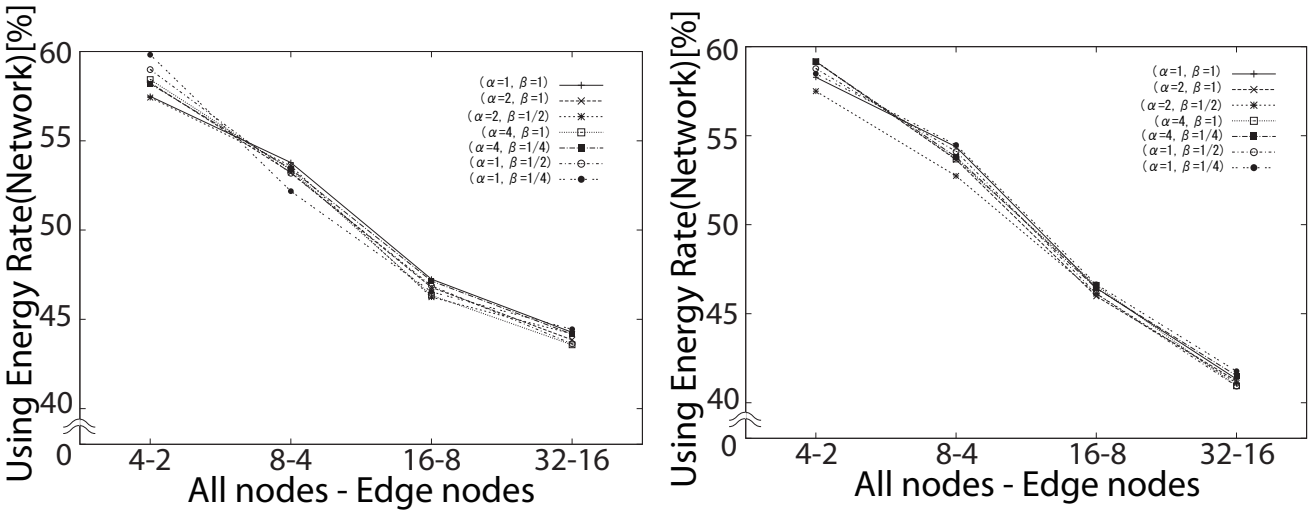
4.5.5 係数 α,β の決定

先の最適化例では補正を使用しない係数を設定したが，ここで再度，DCNP のアルゴリズムにふさわしい係数の導出を試みる．

表 4.5.3: シミュレーションの条件

OS	Ubuntu9.10(Linux 2.6.31-22-generic)
CPU	Intel Celeron Processor SU2300
使用言語	Perl
ネットワーク生成(ノードの種類選択)	乱数(rand関数)で選択
ネットワーク生成(回線の生成)	rand() mod 2を使用、奇数偶数で有無を決定
シミュレーション実行回数	500
備考	ノードの孤立時に計算破棄、ネットワーク再構築

表 4.5.3 はシミュレーション実験の条件を記載したものである．使用 OS は Ubuntu9.10，カーネルは Linux となっており，設定条件は DCN と同じである．またシミュレーション回数も同様の 500 回とし，エッジノード孤立が起こった場合の再計算も実装した．

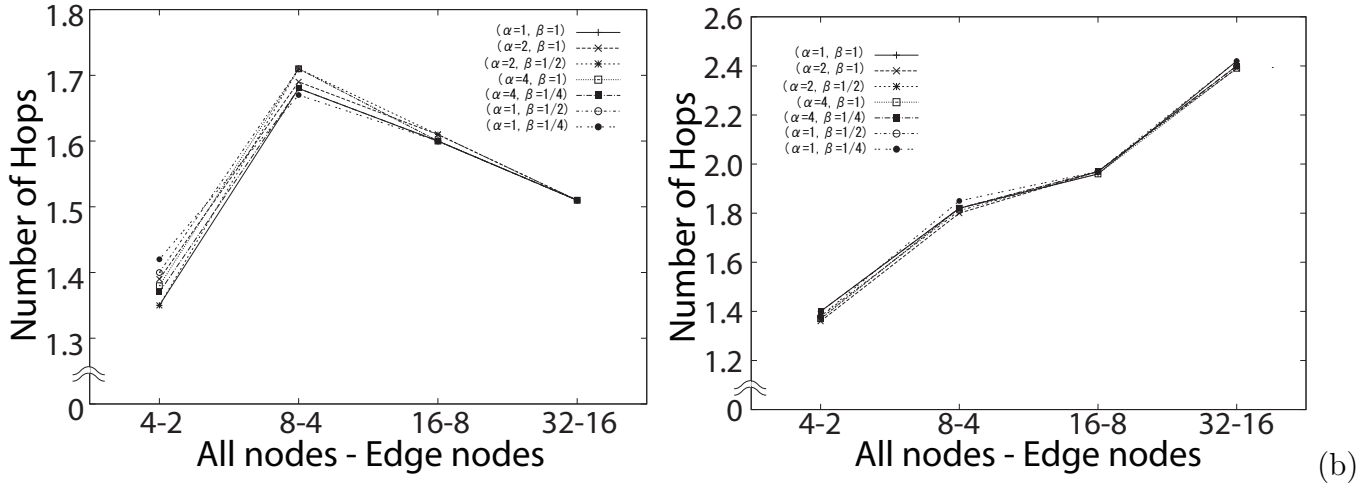


(a) 均一な初期コスト設定 (全て 1)　　(b) 乱数による初期コスト設定

図 4.5.4: 係数とそれぞれの消費電力平均

図 4.5.4(a) は初期コスト I に $C = 1000$ とした OSPF コスト導出式を用いたもので，それぞれのコストが 1 に固定される．図は 1 から 100 までの値を乱数で選択したもので，それぞれの初期コストはランダムである．今，32-16 を例として取り上げるとき，隣接情報の補正なしを示す係数 $(\alpha = 1, \beta = 1)$ は 44.27 パーセントの消費電力，もっとも結果が良かった $(\alpha = 4, \beta = 1)$43.57 パーセントと，1 パーセント以下の差に収まった．もっとも結果の悪かった係数は $(\alpha = 1, \beta = 1/4)$ のとき，44.44 パーセントであるが，最良との差も 1 パーセント以下に収まっている．図 4.5.4(b)，乱数コストを使用したシミュレーションでは，$(\alpha = 1, \beta = 1)$ のとき 41.33 パーセント．最良の結果は $(\alpha = 1, \beta = 1/4)$ のときで 40.93 パーセント．最悪の結果は $(\alpha = 1, \beta = 1/4)$ のときで 41.77 パーセントであった．乱数を用いたシミュレーションでも，OSPF コスト導出式を用いたものと同様に，それぞれの差は 1 パーセント未満に収まっている．ネットワーク規模が小さいうちは 1 パーセントあたりの消費電力も小さく，機器の変化や回線数により消費電力に影響が及びやすいが，規模が大きくなると 1 パーセントあたりの電力が大きくなり，機器の種類による影響や誤差も及びにくくなる．それでいてなお 1 パーセント未満に収まる結果であるということは，DCNP ではどのような係数を用いても補正を行わない $(\alpha = 1, \beta = 1)$ の結果と大差がないという結論を導くことができる．最適な係数を導けないのならば，補正そのものをしないほうがよいという観点からも，この結果を採用するほうがよいと考えられる．

またネットワークを構築する上で，ホップ数は非常に重要な要素である．各エッジノードからエッジノードまでのホップ数は，可能な限り小さいほうが好ましい．よって次に DCNP の性能をホップ数の面から考察することにした．



(a) 均一な初期コスト設定 (全て 1)　　(b) 乱数による初期コスト設定

図 4.5.5: 係数とそれぞれの平均ホップ数

図 4.5.5(a) は OSPF コスト導出式を用いて設定した初期コストに DCNP を適用した際のネットワークのホップ数を，図 4.5.5(b) は乱数で初期コストを設定したものに DCNP を適用した際のホップ数である．(a) 図中”32-16”は全体ノード数 32 のうち，エッジルータ 16 個であることを示している．この 32-16 ではすべての係数パラメータにおいて，1.51 を示しており，もっとも分散したのは，機器の種類による影響を受けやすい 4-2 での結果であった．最適化問題は，ネットワークの規模が大きくなるほど難しくなり，なおかつ機器の種類による影響を受けなくなるために結果が安定する．このことから考えると，どの係数を使用しても結果はほぼ変わらないといえる．(b) の乱数によるコスト生成の結果も同様で，32-16 の結果は 2.4 に集中しており，すべての値が 0.1 以下の誤差に収まっている．また他すべてのネットワークにおいて，それぞれの誤差が 0.1 以下なるなど，係数の値を変更してもホップ数に変化が見られないということがわかった．この結果および，先の消費電力の結果より，DCNP は係数による OSPF コスト補正を用いても消費電力およびホップ数の改善は見られず，補正を使用しないパラメータ，$(\alpha = 1, \beta = 1)$ で十分な削減効果が期待できるといえる．

4.5.6　DCNP の概念を再定義

先の実験より，DCNP は係数 (α, β) にどのような値を入れてもほとんど変化がなく，両者が 1 でも十分な電力削減を得られることがわかった．これは係数を用いた補正が不必要であることを示しており，DCNP の概念を再定義する必要がある．ここでは隣接情報を用いた補正の定義を見直し，再決定する．

図 4.5.6 は図 4.5.2 に示された DCNP の概念図を修正し，エッジノード間のみの補正に切り替えたものである．ここで I は初期コストを，L は消費電力最適化 OSPF コストを示している．I は任意，またはランダムで決定されるほか，ノード A，B 間におけるコスト L の導出式は次式で示される．

$$\rho = (Wpp_a + Wpp_b)/2 \tag{4.5.5}$$

$$L = \rho \cdot I \tag{4.5.6}$$

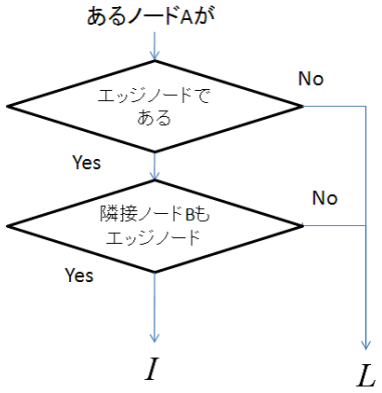


図 4.5.6: DCNP の概念 (フローチャート)

ここで Wpp は 表 4.5.1 にて消費電力を正規化した数値，ρ は消費電力最適化のための重みづけ係数になっている．隣接ノードが互いにエッジノードのとき補正を残す理由は，DCNP で求められる数値は必ず $(I \leq L)$ であり，I のほうが経路に選択されやすく，なおかつ手動で任意に設定が可能なためである．次に，これらの式を用いて再定義された DCNP のアルゴリズムを記載する．

```
1   sub DCNP_RE(){
2     for($i=0;$i<$max;$i++){
3       for($j=$i+1;$j<$max;$j++){
4         if($dist[$i][$j]!=INF){
5           if(!$edge[$i] || !$edge[$j]){
6             $dist[$j][$i]=$dist[$i][$j]*=($Wpp[$node_type[$i]]+$Wpp[$node_type[$j]])/2;
7           }
8           if($dist[$i][$j]<1){
9             $dist[$i][$j]=$dist[$j][$i]=1;
10          }
11          $dist[$j][$i]=$dist[$i][$j]=int($dist[$i][$j]);
12        }
13      }
14    }
15  }
```

ここで定義された DCNP_RE は DCN で定義された係数 (α, β) を使用せず，エッジノード同士が隣り合う環境のみを 5 行目で考慮している．また先に定義された式は 6 行目に実装されている．以降の実験では再定義された DCNP_RE を DCNP として用いるものとする．

4.6 おわりに

本章では機器の消費電力を用いた消費電力削減手法として 3 つの手法を提案した．消費電力を用いて OSPF コストを変更する DCP，係数を用いる DCN，正規化した消費電力情報を用いる DCNP である．実験の結果，DCN では $(\alpha = 4\beta = 1)$ の係数を使用した場合もっとも結果が良くなることが分かった．また DCNP には係数が必要なく，正規化した消費電力を用いるだけで十分な性能が期待できることが分かった．次に 5 章にてシミュレーション実装を行い，従来手法との比較を行っていく．

第5章　比較実験と結果

5.1　はじめに

本稿の第2章では，従来手法としてEEE，ECO-RPの解説を行った．また第3章では実際の機器の消費電力を測定して消費電力モデルを作成し，第4章では提案手法としてDCP，DCN，DCNPを提案し，動作の確認と係数の導出を行った．本章ではそれらを用いてシミュレータを作成し，各々の手法における消費電力，ホップ数，トラフィック溢れ発生率の性能を比較する．また異なるトラフィックモデルを用いて実験を行い両者の差を比較することで，トラフィックに応じて動的にコストを変更するECO-RPの特性を把握するとともに，トラフィックモデルの妥当性を証明する．はじめに実験に用いるシミュレータの解説を行い，シミュレーションの条件，使用するトラフィックや定義を記述する．また実験結果は提案手法同士の比較，ECO-ROとの連携，従来手法との比較，の3つにわかれており，それぞれを分割して記載する．本実験では最初に提案手法同士を比較し，もっともすぐれた結果を残したものを従来手法との比較にかけた．続いて異なるトラフィックでの結果を導出，比較し，最後に特徴や問題点を考察した．

5.2　実験環境，条件と手法

5.2.1　シミュレータの構造

図5.2.1は本実験で使用するシミュレータの動作をフローチャートで示したものである．何も指定しない場合の動作をOSPFのDijkstra法による最短経路選択と位置付け，それぞれの削減手法を追加で設定していくものとした．DCP，DCN，DCNPによるコストの最適化は，初期コストに対して一度だけ行われるものであり，EEEはOSPFの経路選択によりポートの電力を削減する．ECO-RPはトラフィックに応じてOSPFコストを変動させるため，ループ内に組み込まれている．このループにはTIMEが設定されており，入力するトラフィックデータの個数が該当する．繰り返しに応じてカウントを増やしていき，TIMEを超えた時点でトラフィックデータが終了したと判断，データを処理してシミュレーションを終了する流れである．次に，フローチャートを実装する言語，実行する環境，入力するパラメータについて記述する．

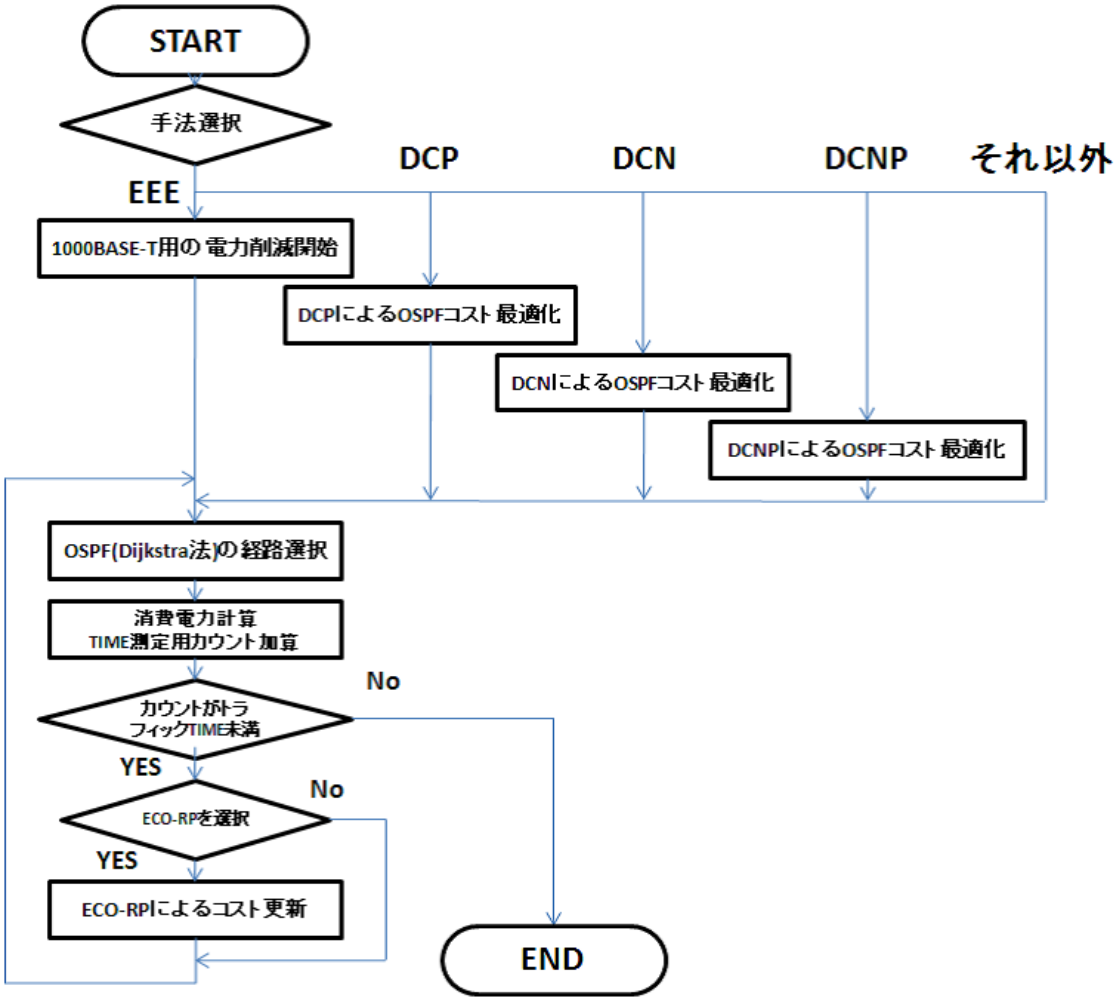


図 5.2.1: シミュレータの構造 (フローチャート)

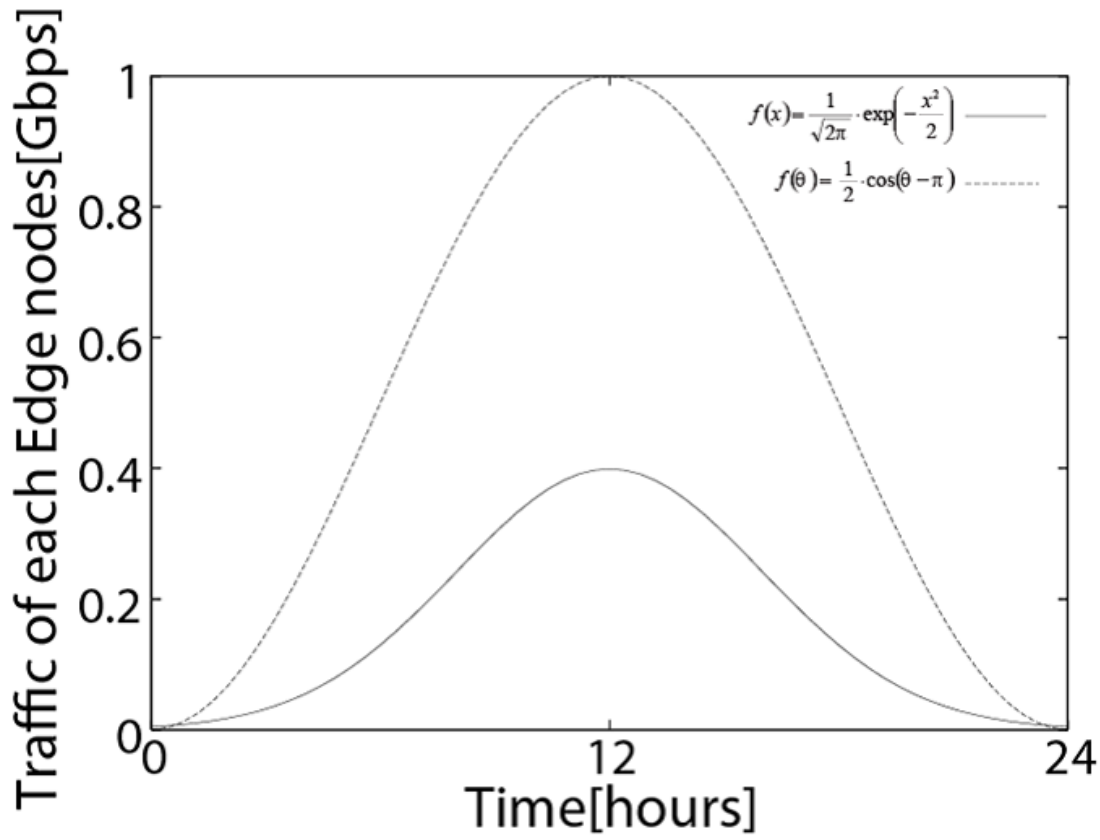


図 5.2.2: トラフィックモデル

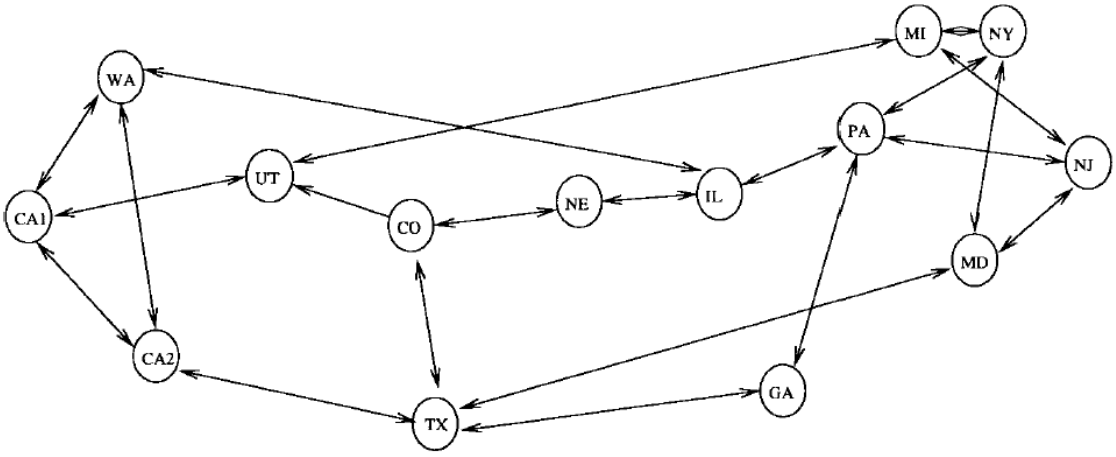


図 5.2.3: NSFNET T1(1989 年)[13]

表 5.2.1: シミュレーションの仕様と実験条件

実験番号	(A)	(B)
使用トラフィック	標準正規分布	
トラフィックTIME数	61(-3≦TIME≦3を60分割)	
OS	Redhat Linux 5.4	Ubuntu9.10 Linux 2.6.31-22-generic
CPU	AMD Opteron 2.6GHz × 4	Intel Celeron SU2300 1.2Ghz × 2
使用言語	Perl	
ネットワークトポロジ	NSFNET T1 backbone	乱数による生成
ノードの種類選択)	定義された電力モデルの中から、乱数を用いて決定	
回線の生成		乱数にて、奇数のとき生成
シミュレーション実行回数	500	
備考		ノードの孤立時に再計算

5.2.2 実験条件と各種パラメータ

表 5.2.1 はシミュレータの仕様と実験条件である．実験は (A)NSFNET T1 のネットワークトポロジ，(B) ノードの種類と回線をランダムで生成，の 2 つを用いて行う．まず図にトラフィックモデル図を示す．本実験では，標準正規分布をトラフィックモデルとして使用する．ここでは y 軸がネットワーク使用率となっており，最大値を 1Gbps とした際，一つのエッジノードから流れるデータ量を示している．また x 軸は時間の流れを示している．標準正規分布とは $(\mu = 0, \rho^2 = 1)$ となる正規分布であり，次式で定義できる．

$$f(x) = \frac{1}{\sqrt{2\pi}} \cdot \exp(-\frac{x^2}{2}) \tag{5.2.1}$$

図 5.2.2 にもあるとおり，この式は $x = 0$ で左右対象となるグラフを形成する．実験ではこのグラフを比較実験のトラフィックモデルとして利用．0.1 毎に分割して 61 個のデータを作成した．このグラフをエッジノードから流れ出るトラフィックの流量として利用する．また一つのエッジノードから流れるトラフィックは，すべてのエッジノードに対し均一量で流れていくものと仮定する．仮に全体ノード数 17，エッジノード数 4 というネットワークがあったとする．今トラフィック TIME が”12”の数値を使用するとき，Traffic は 0.4Gbps となる．つまり一つのエッジノードから流れでるトラフィックは 400Mbps となる．これが各々のエッジノードに分割されて流れるため，エッジ一つあたりに流れるトラフィックの量は 133Mbps となる．またこれらの定義と同様に，ECO-RP の動作と連携状態の性能を比較するためのトラフィックを定義した．

$$f(\theta) = \frac{1}{2} \cdot \cos(\theta - \pi) \tag{5.2.2}$$

先の式は図 5.2.2 に書かれているコサイン波形のものである．位相をずらしている以外に変更点はなく，形状も正規分布をモデルとしたグラフと似ている．本実験では，図中にプロットされた正規分布

を，各手法の消費電力，ホップ数，トラフィック溢れ発生率を図るための指針とする．またこれらの結果をコサイン波形による結果と比較することで，トラフィックに応じたOSPFコストの動的更新をもつECO-RP，および提案手法と連携時における消費電力がどのように変化するのかを把握する．これによって異なるトラフィックでの動作傾向の違いを適切に把握することができる．

実験(A)では実際のネットワークに対する試験として，図5.2.3に示されるNSFNET T1 backboneネットワークのトポロジを使用する．NSFNET T1は1989年にアメリカ科学技術財団によって稼働された北アメリカ全土に及ぶ学術ネットワークであり，現在のインターネットの前身にもなったものである．本実験ではノードの個数とトポロジのみを継承し，ノードの種類とについては定義した電力モデルの中から乱数で決定される．またコストもOSPFコストの導出式，および1から100の乱数を用いて設定するものとする．本実験ではエッジノード数を徐々に増やし，その違いによる削減率の変化を調べて比較する．これと並行して(B)ではノードの個数とトポロジをランダムで生成するシミュレータを用いた実験を行う．この実験はネットワーク内に存在するノードの総数を変更し，規模による削減率の違いを調査，比較するものである．

続いて使用機器を見ていく．(A)はUbuntu9.10，intel Celeron SU2300 × 2のコンピュータで計測．(B)はRed hat Linux 5.4，AMD Opteron × 4のコンピュータで計測を行った．前者は1Gbyteメモリのラップトップコンピュータ，後者は24GByteのメモリを持つ大容量並列計算器である．使用した言語はPerl，ネットワーク生成時に選ばれる機器の種類は乱数によるランダム選択とした．また，回線については先のシミュレーションと同様に乱数を発生させ，奇数のとき回線をつなぎ，偶数のときはつながない，という選択式にした．今回ネットワークに設定する回線の帯域は全て1Gbpsである．これにトラフィックモデルを用いた実験を行う．それぞれのトラフィック変化のTIME数が60回の計算を必要とする．この60回を1回のシミュレーションとして，として全体のシミュレーション回数は500回．ノードの孤立が発生した際には，ネットワーク生成を最初からやり直すものとする．本実験では消費電力率，ネットワークノードホップ数，トラフィック溢れの発生回数を測定し，平均値の導出を試みる．またそれをもとにグラフを生成し，比較と考察を行う．また今回は提案手法と従来手法ECO-RP，EEEのほか，一般的に広く使用されるOSPFを用いた実験を行い，評価のための指針とした．従来のOSPFには消費電力を削減する機能は付属していないが，本実験では経路に属さないノードを停止できるものと仮定した．

5.2.3 初期OSPFコストの設定

本実験で使用するネットワーク回線に対するコストの付与は，OSPFコストの導出式および1から100の乱数を用いて設定する．今，ネットワークの初期コストをIとするとき，式は次のとおりである．

$$I = C/B \qquad (C = 1000) \tag{5.2.3}$$

Bはインターフェースの通信帯域$(Mbps)$であり，Cは任意の整数である．IおよびOSPFコストの値は整数値でしか定義ができない．少数は切り捨てであり，1以下になった場合には1の値に補正される．また本実験では$C = 1000$と定義した．電力のモデルは1000Mbpsの通信インターフェースを対象としているため，導出式を用いた場合，ネットワークの全コストが1に固定される．また導出式とは別に，$(1 \leq I \leq 100)$の範囲で乱数を用いたコスト設定を行うものとした．値の決定にはperlのrand関数を用いるものとする．続いて各評価の詳細について記載する．

5.2.4 消費電力削減率

提案手法の性能を測るため，消費電力削減率の評価を行う．ここで取り上げる消費電力とは，個々のネットワーク機器や回線を含むネットワーク全体の消費電力である．一般に不要とされる電力をどれだけ削ることができるかは，各手法の性能を図る上でもっとも重要な項目である．しかしながらネットワークの消費電力は構造や機器の種類によって大きく異なるため，電力そのものを評価の指針とするのは難しい．そこで本項ではネットワークの消費電力をパーセンテージで示すこととした．全体実験回数500回のうち，i回目に構築されるネットワークの全体消費電力をW_i，さらにトラフィックTIMEのある地点jで最適化されたネットワークの消費電力をW_{i_j}とすると，消費電力のパーセンテージ平均W_{rate}は次の式で求められる．

$$W_{rate} = \frac{\sum_{i=1}^{500} \sum_{j=1}^{TIME} W_{i_j}}{\sum_{i=1}^{500} \sum_{j=1}^{TIME} W_i} \cdot 100 \tag{5.2.4}$$

ここで定義する消費電力パーセンテージは，ネットワーク全体の消費電力に対する，削減後の消費電力の割合である．そのため平均値を求めたい場合には各ネットワーク構造，トラフィック状態における削減後電力を合計し，同様に合計したネットワーク全体の電力で除算することで平均割合を求めることができる．これを100倍したものが消費電力平均のパーセンテージである．この値は全体ノード数とエッジノード数の組み合わせ毎にほぼ固定の値をとるため，各手法ごとの値を比較することができる．本実験では各手法における消費電力のほか，どの手法を用いても削減することのできないエッジノードの消費電力合計を理想の消費電力値Idealとして定義した．この値に近い手法ほど，優れた消費電力削減率を持つ手法であるといえる．また一切の消費電力削減を行わない場合の消費電力をNo-reductionとし，グラフ中の100パーセントとした．

5.2.5　平均ホップ数

消費電力の比較に続き，エッジノード間のホップ数に対する比較を行う．ホップ数はあるエッジノードからエッジノードにトラフィックが流れる際，いくつのノードを経由したかで測定される．ここで重要になるのが，ネットワークの理想ホップ数である．あるネットワークにおいてリンクに設定されたコストが全て均一だと仮定すると，各エッジノード間を Dijkstra 法によってルーティングした結果の経路は，コストの面はもちろん，ネットワークの構造から見ても必ず最短の経路を通る．これが理想のホップ数であり，どれだけすぐれたルーティング手法を用いても物理的に回線を増やさない限り，このネットワークホップ数を下回ることはできない．本実験ではこの値をホップ数における理想の値と定義し，グラフ中 Ideal として定義した．Dijkstra 法は OSPF 内に設定された最短経路導出アルゴリズムであるため，本実験ではそれぞれの手法を OSPF と比較することで，理想ホップ数により近い手法の選出を行った．次にホップ数の導出式を説明する．まずエッジノード間のホップ数のみを返す関数を $hop(n, m)$ と定義する．このとき n と m にエッジノードではない番号の組み合わせが入った場合，$hop(n, m)$ は 0 を返すものとする．今，ネットワークにある全てのノード数を N，エッジノード数を E とするとき，任意のネットワーク構造かつトラフィック TIME の一状態における平均ホップ数 $hop_{ave}(j)$ は次の式で表される．

$$hop_{ave}(j) = \frac{\sum_{n=1}^{N-1} \sum_{m=n+1}^{N} hop(n, m)}{(E-1)(\frac{E}{2})} \tag{5.2.5}$$

式の分子はネットワーク全体のホップ数を表しており，分母は互いのエッジノード間に張られたリンク数の合計を示している．なおこの式では $(n \neq m)$ であり，なおかつ n と m に入る同じ数値の組み合わせは，一度しか計算されない．たとえば $(n = 1, m = 3)$ と $(n = 3, m = 1)$ は同じものとしてカウントされ，一度しか計算が行われないものとしている．この式によって得られるのは，トラフィック TIME 中における一時点の平均ホップ数であるため，実験回数分の平均値を求めるためには，トラフィックの全状態をカバーした，ネットワーク構造毎の平均ホップ数を求める必要がある．ここで，実験の結果として導出される平均ホップ数 hop_{entire} の式を定義する．

$$hop_{entire} = \frac{\sum_{i=1}^{500} \sum_{j=1}^{TIME} hop_{ave}(j)}{500 \cdot TIME} \tag{5.2.6}$$

式では全体の平均を求めるため，各トラフィック状態と実験回数分のホップ数を合計し，それを回数で割るという手法を用いている．この結果導出されるのが，任意の全体ノード数とエッジノード数で実験した際の平均ホップ数である．

5.2.6　トラフィック溢れ状態の定義

本実験では各手法の消費電力削減率とホップ数のほか，回線集約によるトラフィック溢れへの陥りやすさを計測，比較する．本来，回線の混雑状況を図る指針としては輻輳があるが．し実際の輻輳発生条件は，機器や状況によって大きく異なり，必ずしも特定の状況を定義できるものではない．よってここ

では性能評価の指針としてトラフィック溢れを採用し，回線の最大容量を超えた時点で明らかな通信不能状態と判断し，トラフィック溢れ状態と定義することにした．本実験ではネットワーク全体の流量を常に監視し，トラフィック溢れの発生率を導出することとした．今，いずれかの回線がトラフィックの最大容量を超えたトラフィック TIME の合計数を T_o，全体のトラフィック TIME 数を T_{all} とした場合，トラフィック溢れの発生率 $P_{overflow}$ は次の式で表される．

$$P_{overflow} = T_o/T_{all} \cdot 100 \tag{5.2.7}$$

ここで $P_{overflow}$ は，ある任意のネットワーク構造に対し，指定されたトラフィックを用いてネットワークの最適化と運用を行った際，トラフィック TIME 毎にいずれかの回線で最大容量を超える確率である．また先の式は単一のネットワーク構造に対する式であり，実験回数 500 回を考慮していない．今，$P_{overflow}$ を P_i と置き換えると，各ネットワーク構造で回線の最大容量を超える平均確率 P_{ave} は次の式のようになる．

$$P_{ave} = \frac{\sum_{i=1}^{500} P_i}{500} \tag{5.2.8}$$

P_{ave} の値は多数のネットワーク構造の平均確率であり，全体ノード数とエッジノード数の組み合わせ毎にほぼ固定のものとなる．そのため異なる手法との比較が容易になるという利点を持っている．

このほか，特に重要な項目としてネットワーク構造の再計算がある．ネットワークの最適化を行ったとき，いずれかの回線が最大容量を超えてしまった場合には，構造の再計算をし直す必要がある．しかしながら今回は再計算の実装および問題の考慮をしないものとした．

5.2.7　ネットワーク電力の定義範囲

これまで一通りシミュレーション条件を解説してきたが，最後にネットワークの電力定義範囲について記述する．本稿における電力定義範囲とは，あるネットワークが，3 章で定義された電力モデルに沿って生成された際，とりえる電力の範囲である．今，ポート毎の消費電力を考えない者とした場合，ネットワーク全体の消費電力は次の表のように変動する．

表 5.2.2: ネットワークの規模と 1 パーセントあたりの消費電力

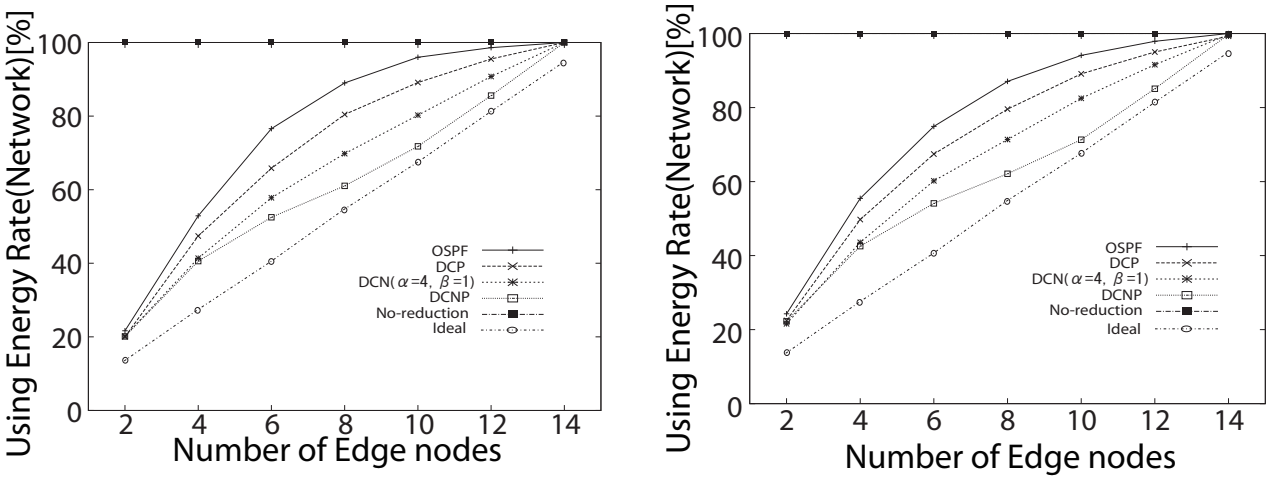
機器名	基礎消費電力
DGS-3426	52W
DGS-3450	70W
Alpine 3804	221W

全体ノード数	電力(100%)の範囲(W)	1%の範囲(W)
4	204 - 884	2.04 – 8.76
8	408 - 1768	4.08 – 17.52
16	816 – 3536	8.16 – 35.04
32	1632 - 7072	16.32 – 70.08
64	3264 - 14144	32.64 – 140.16

5.3 提案手法同士の比較(正規分布トラフィック)

提案手法と従来手法を比較する前に，本稿で提案した手法のうち，どの手法がもっともすぐれた性能を持つのかを測定，比較する必要がある．よってまず最初にシミュレータを用い，DCP，DCN，DCNP に対する比較実験を行った．評価するデータは，ネットワーク全体の消費電力，エッジノード間のホップ数の平均値，トラフィック溢れの発生率である．ここで DCN,DCNP に使用されるパラメータは第 3 章のシミュレーションより得られた結果を採用した．

5.3.1 消費電力削減率の評価



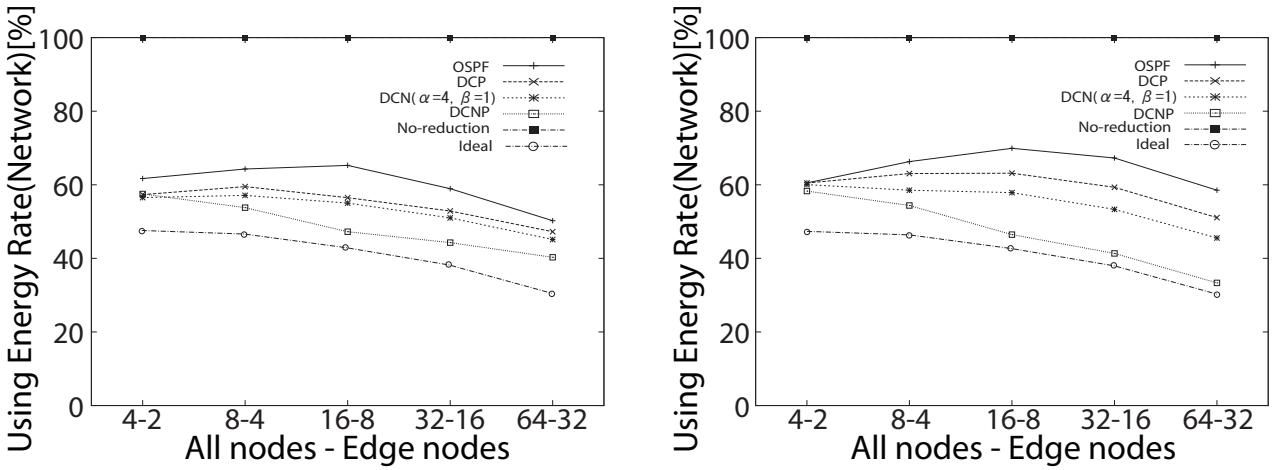
(a) 均一コスト使用時の平均消費電力　　(b) 乱数コスト使用時の平均消費電力

図 5.3.1: NSFNET T1 の最適化消費電力

図 5.3.1(a) はシミュレーション条件 (A) に従い，OSPF コスト導出式を用いた均一コスト設定を行った結果，得られた消費電力パーセンテージの平均値をプロットしたグラフである．全体ノード数は 14 に固定されており，y 軸はエッジノードの個数を示している．x 軸はネットワーク全体の消費電力を表しており，削減を行わない場合の消費電力を 100 パーセントとしている．削減を適応しない場合の消費電力は，図中 No-reduction の線で，削減率の限界であるエッジノードの消費電力は Ideal で示されている．今，エッジノード数”8”に焦点を当ててみる．もっとも削減率が低かったのは OSPF をもちいて行われたもので，消費電力の割合は 89.01 パーセント．一方でもっとも性能がよかったのは DCNP を用いた際の 61.04 パーセントであった．ネットワークノード数 16 における 1 パーセントあたりの消費電力を基準として用いる場合，もっとも少ない場合でも 1 パーセントあたりに占める電力の割合は 8W 程度になっている．今 OSPF と DCNP の消費電力割合の差は 27.97 パーセント．単純計算でも 223.76W，DGS-3426 が 4 台分以上の消費電力差がある．DCNP の次に性能がよかったのは DCN で 69.8 パーセント，次には DCP が 80.43 パーセントと続いている．また全体的に見ても，DCNP のグラフは常にもっとも下側に位置しており，OSPF および他の提案手法と比較してもすぐれた削減効果を発揮しているといえる．またすべてのグラフより，エッジノード数が少ないうちは削減率も高く，最終的にすべての例

で 100 パーセントまで上昇することが確認された．

続いて (b) は 1 から 100 の乱数を生成しコスト設定を行った結果，得られた消費電力パーセンテージの平均値グラフである．削減を行わない状態を No-reduction で表してあるほか，グラフ全体の消費電力割合は均一コストのケースよりも若干高いものの，形状はほぼ同じである．先と同じくエッジノード”8”を見てみることにする．もっとも性能が良かったのは DCNP で 62.12 パーセント．最悪は OSPF で 87.07 パーセントであった．差は 24.95 パーセント．電力定義範囲を参考にするとおよそ 199.6W，DGS-3426 四台分である．また全体的に見ても DCNP を用いた初期 OSPF コスト更新を行ったものは，常に他よりも低い消費電力となっている．よって，NSFNET T1 ネットワークのトポロジでは，DCNP がもっとも優れた削減率を提供したといえる．



(a) 均一コスト使用時の平均消費電力　　(b) 乱数コスト使用時の平均消費電力

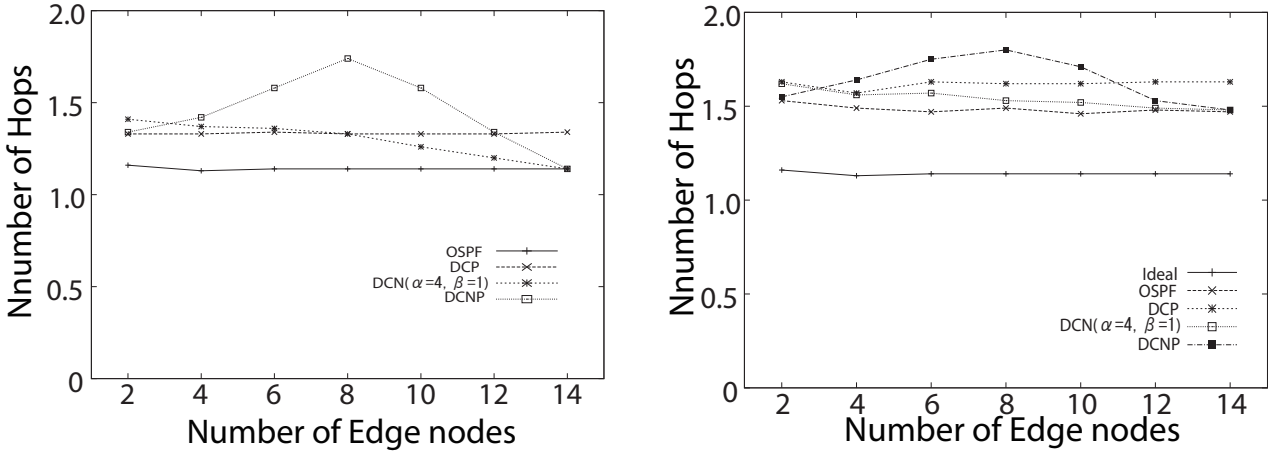
図 5.3.2: ランダム生成ネットワークの最適化消費電力

図 5.3.2 はランダムなネットワーク生成して実験を行った際の結果である．(a) では初期コストに OSPF の導出式を用いたコスト設定を行っている．x 軸の”16-8”は全体ノード数 16，エッジノード数 8 を意味している．この実験ではエッジノードの割合は変えず，全体ノードの個数を変化させることで，消費電力の推移を計測した．電力を削減しない場合を No-reduction とし，100 パーセントのグラフで表している．今，図中”16-8”について見てみる．もっとも性能が悪かったのは先と同じ OSPF で 58.94 パーセント，よかったのも同じ DCNP で 44.27 パーセントであった．およそ 14 パーセントの差，1 パーセントを 8W として消費電力に直すとおよそ 112W となり，こちらも決して小さくはない差といえる．全体を通して常に一番下にあったのは DCNP であり，結果が悪かったものは OSPF であった．また，”4-2”における DCNP の値は 57.49 パーセント．”64-32”のときは 40.28 であった．このことからエッジノードが増えると消費電力は上昇するが，同じ割合のエッジノード数でも，規模が拡大すると消費電力パーセンテージは下がることが分かった．これは経路が増えることによって，経路集約による削減数が増えているためだと考えられる．

続く (b) は乱数を初期コストに用いたものである．もっとも低い消費電力になったのは，図中”64-32”の DCNP で 33.34 パーセントである．逆に最も高かったのが OSPF で 58.51 パーセントとなっている．先の結果と同様に，今回も 20 パーセント近い差が付いており，DCNP は乱数をコストに用いたネット

ワークの削減にも有効であるということが言える．乱数を初期コストに使用する場合，OSPF では”4-2”のとき 60.5 パーセント．”64-32”のとき 58.51 と，ほとんど変化が見られないのに対し DCNP は 58.28 パーセントから 33.34 パーセントと 20 パーセント以上の電力削減が行われている．これは経路の数が増えることによってより電力的に効率のよいルーティングができたためと考えられる．全体を見てみても DCNP は常に低い位置を取り続けた．よってランダムネットワークで実験した場合，初期コストにかかわらず DCNP が有効であるといえる．

5.3.2　ホップ数の評価

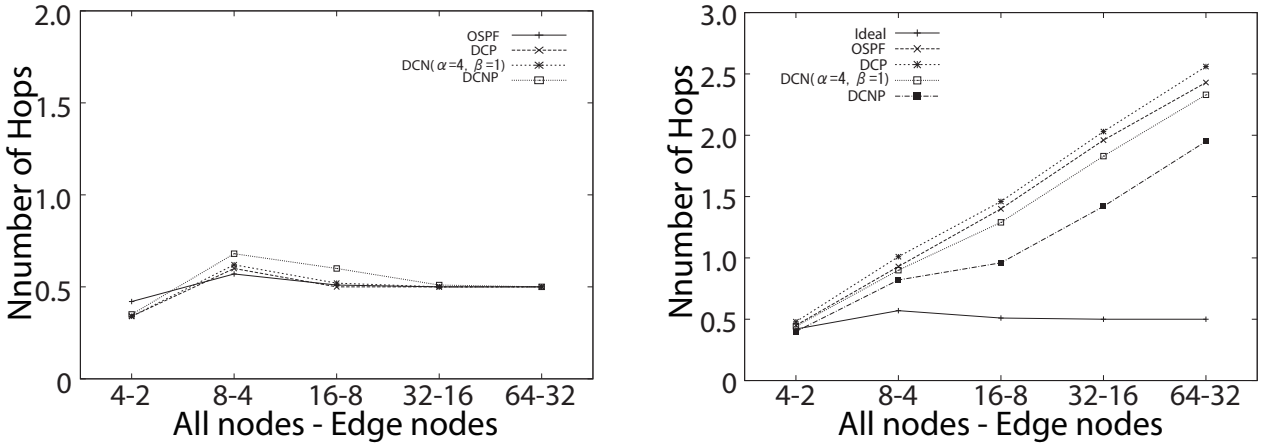


(a) 均一コスト使用時の平均ホップ数　　(b) 乱数コスト使用時の平均ホップ数

図 5.3.3: NSFNET T1 の平均ホップ数

次に平均ホップ数の比較を行う．図 5.3.3 はシミュレーション条件 (A) に従って実験を行った結果が示されている．このうち (a) は初期コストに OSPF コストの導出式を用い，エッジノードの個数を変化させて平均ホップ数を測定したグラフである．今，x 軸が”8”のときのホップ数について考えることにする．図中”8”は全体ノード数 14 に対しエッジノード数 8 を示す．先の平均ホップ数の説明でも記述したが，OSPF コストが均一であるネットワークにおいては，Dijkstra 法を用いたルーティング手法が，常に理想の値をとり続ける．ここで OSPF が優れた結果になっているのはそのためで，このグラフに近い数値と形状をもつグラフがより優れた平均ホップ数を持つ手法であるといえる．今 OSPF の値は 1.14，続いて DCN が 1.33，DCP が 1.33 と同一なのに比べ，もっとも結果が悪かったのは DCNP で 1.74 となっている．この 0.5 の差は大きく，OSPF を用いたルーティングでは，ホップ数が 2 となるノードは 5 台に 1 台であるのに対し，DCNP では 5 台に 3 から 4 台がホップ数 2 となる計算である．全体を通して見てみると OSPF と DCP にはほとんど増減がなく，DCN がノード数に応じて減少，DCNP は大きく上昇した後に下がるというグラフになっている．これは DCNP によるコスト変更によって不適切と判断された経路，ノードが迂回されたためだと考えられる．エッジノードの増加とともにホップ数が減る理由は，隣り合うノード間の考慮が大きく影響しているものと考えられ，同様の考慮を行う DCN にもホップ数の減少が見られた．

続いて初期コストへ 1 から 100 の乱数を設定して実験を行った結果のグラフを (b) に示す．プロットされているグラフのうち，”Ideal”は OSPF コスト導出式を用いた際，さらに Djkstra 法で経路計算された結果のホップ数であり，もっとも最短なパスで各エッジノードをつないだ際のホップ数を示している．また，コストに乱数を用いたことで全体的に平均ホップ数が上昇している．しかしながらそれぞれのグラフの形状は OSPF 導出式を用いたものと変化しておらず，x 軸”8”での DCNP がもっとも大きな値 1.8 を記録している．これは DCNP による機器の消費電力を考慮した経路集約が非常に優れるために迂回が発生しており，また”8”でエッジノードのばらつきがもっとも最大に達するためだと思われる．これを超えるとほとんどのエッジノード同士が隣り合うため DCN でも同様のホップ数減少が起こっている．実際，すべてのエッジノードが隣り合うようになる”14”では OSPF，DCN および DCNP のホップ数は一致している．



(a) 均一コスト使用時の平均ホップ数　　(b) 乱数コスト使用時の平均ホップ数

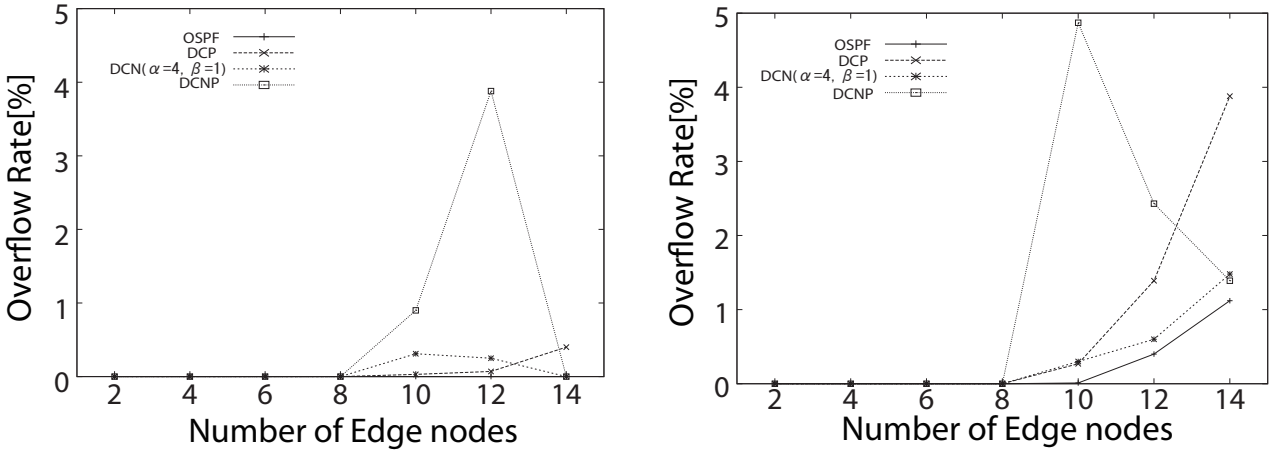
図 5.3.4: ランダム生成ネットワークの平均ホップ数

次に図 5.3.4(a)，シミュレーション条件 (B) を用いて実験を行った結果のグラフを見てみる．こちらも初期コストに OSPF コストの導出式を用いたものだが，理想の値となる OSPF のグラフ付近に値が集中している．このときホップ数が 0.5 付近に集中していることからも，ネットワーク生成においてエッジノード同士が隣接する環境が多数発生したものと考えられる．図中”8-4”で DCNP が 0.68，OSPF が 0.57 とともに最大値を記録した他は大きな差が見られなかったが，本グラフでも DCNP のホップ数が若干多い結果となった．このことより OSPF コストを導出式に用いた場合では，提案手法内では DCN がもっとも理想に近いグラフとなり，消費電力削減率に優れる DCNP の平均ホップ数は増えることが分かった．

最後に (b) を見てみる．こちらは乱数を初期コストに用いて導出した平均ホップ数をプロットしたグラフである．全体ノード数が増えるごとにホップ数も増えているのがわかる．これはネットワーク規模と回線が増大するためである．実際にグラフ全体を見てみると，理想の値”Ideal”にもっとも近いグラフは DCNP となっていた．今 (a) のネットワーク条件にもっとも近い”16-8”について考えてみる．これは全体ノード数 16，エッジノード数 8 でネットワークを構築したものである．このとき”Ideal”の値は 0.51，DCNP が 0.96 であるのに比べ，もっとも悪い結果となったのは DCP の 1.46 であった．

初期コストにOSPF導出式を用いた実験では，どの手法もほぼ同様のホップ数となっていたことを考えると，OSPF，DCP，DCNは乱数コストが設定された場合に最短経路を選択する力が，DCNPよりも弱かったといえる．また(b)の環境はエッジノード同士が隣り合う確率が(a)と比べて高いことも影響していると考えられる．そのためDCNとDCNPは，隣接ノードを考慮しないDCPよりも優れた結果を残している．先の実験より，もっとも電力の削減率が高かったDCNPは，NSFNET T1ネットワークでは平均ホップ数が高かったが，乱数生成によって無数のネットワークを最適化する実験においては優れた結果となることがわかった．このことからDCNPはより汎用性の高い手法であるということが考えられる．

5.3.3 トラフィック溢れ発生率の評価



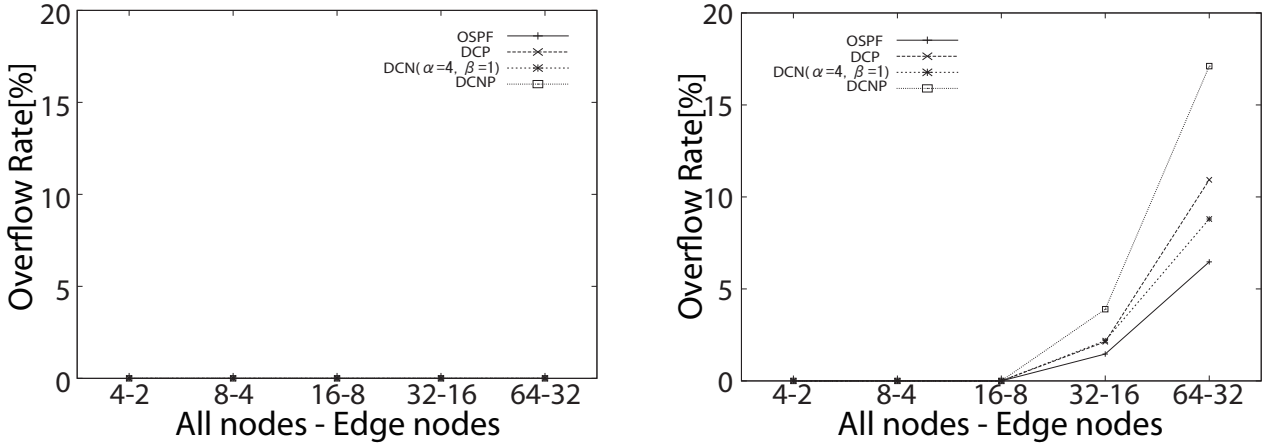
(a) 均一コスト使用時の発生率　　(b) 乱数コスト使用時の発生率

図 5.3.5: NSFNET T1のトラフィック溢れ発生率

ここまで消費電力の削減率，平均ホップ数と比較してきたが，最後にトラフィック溢れの発生率を比較する．図5.3.5(a)はシミュレーション(A)の条件，初期コストをOSPFコストの導出式を用いて均一に設定したものである．ここでx軸はエッジノードの個数，y軸は発生率を示している．ここで定義されるトラフィック溢れ発生率は，指定されたエッジノード数かつトラフィックを用い実験を行ったとき，ネットワーク中に存在するいずれかの回線で，トラフィック量が回線の最大容量を超える確率である．図中エッジノード数”8”まではどの手法も変化がない．しかし”10”でDCNPが0.9パーセントになっており，続くDCNが0.31パーセント，他の手法は0.1以下と明らかな差がみられた，また”12”ではDCNPが3.88パーセント，DCNが0.25となっているほかはすべての手法で0を記録した．これはDCNPが高い電力削減効果をもつ半面，強い集約性の結果としてトラフィックが一か所の回線に集中たためと考えられる．”14”で再び0に戻ったのは，全てのノードがエッジノードになったため，DCNPの隣接補正がかかり経路変更が起こったためと考えられる．同様にDCNも0に戻っている．

次に(b)，初期コストに乱数を用いた結果を見てみる．x軸，エッジノード数”8”までは変化がないのは同じであるが，”10”，”12”，”14”と大きな違いがみられた．”10”の時点ではDCNPがもっとも発生

率が高く，4.87 パーセントである．次が DCN の 0.3 パーセント，DCP の 0.27 パーセント，もっとも少なかったのは OSPF で 0.01 パーセントだった．ここから先，DCNP はエッジノードの隣接状態数が増えたことにより減少に転じるが，逆に急上昇したのが DCP である．”14”で他の手法が 1.4 パーセントに収束しているのにくらべ，DCP は単独で 3.88 パーセントと，DCNP の発生率を超えてしまっている．これは DCP という手法が乱数コスト使用時にエッジノードが増えた場合，DCNP 以上に経路の集約を行ってしまうためだと考えられる．しかしながら”14”は全ノードがエッジノードであるため，コアルータを削減する DCP では電力の削減効果がほとんど期待できない．よってこの場合”14”で DCP を用いることはほとんどなく，大きな問題ではないといえる．なお，提案手法の中でもっとも微増だったのは DCN で”10”のとき 0.3 パーセント，”12”が 0.6 パーセントと他の手法に比べて優れている．



(a) 均一コスト使用時の発生率　　(b) 乱数コスト使用時の発生率

図 5.3.6: ランダム生成ネットワークのトラフィック溢れ発生率

図 5.3.6(a) はシミュレーション条件 (B) の手法で，OSPF コストの導出式を用いて初期コストに均一なコストを設定し，実験を行った際の結果である．全ての状態において点が 0 にプロットされているとおり，すべての状態でトラフィック溢れは見られなかった．先の図でも全ノードのうち半分がエッジノードになるまで変化が見られなかったことから，OSPF コストが均一な環境においては，エッジノードの個数こそが発生率に大きく影響すると考えられる．

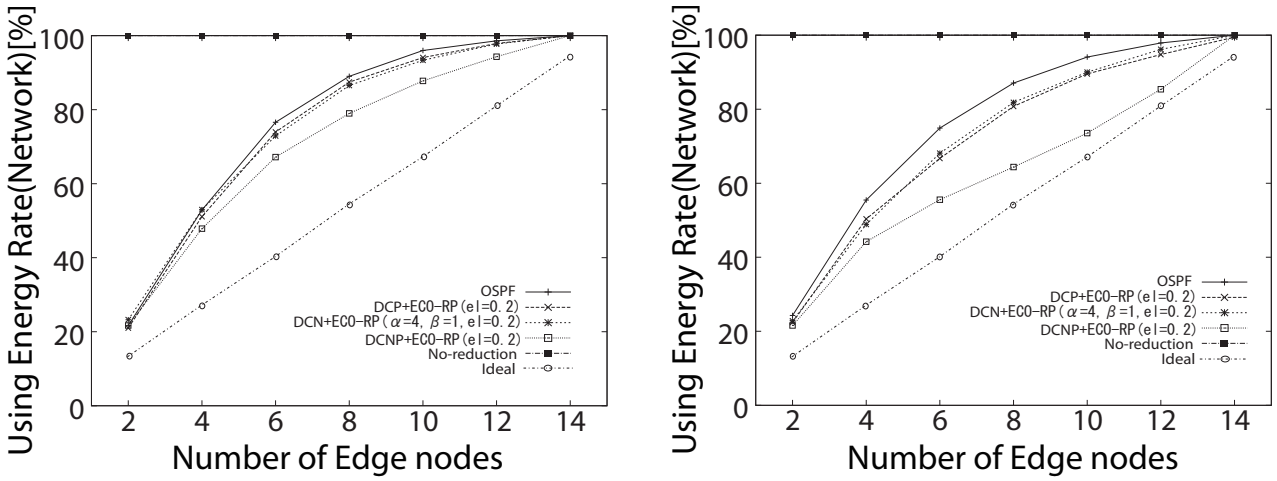
最後に (b) として，初期コストに乱数を用いた結果を示す．初期コストが均一の際は変化が見られなかったものの，こちらでは図中”32-16”から発生率の上昇が始まっている．今，このポイントについて見てみる．もっとも確率が高いのは DCNP で 3.9 パーセント，続いて DCN が 2.19 パーセント，DCP が 2.13 パーセント，もっとも低かったのは OSPF で 1.46 パーセントであった．”64-32”ではさらに差が顕著になっており，DCNP は 17.11 パーセント，もっとも低かった OSPF は 6.47 パーセントと，10 パーセント以上の差がついてしまっている．これは DCNP が優れた消費電力削減能力を提供する代わりに，強い経路集約性を持つためであり，電力的に無駄と判断されたノードを完全に省いてしまった結果，一部のノードにトラフィックが集中するためと考えられる．DCNP は提案手法の中では最も高い消費電力削減効果を持ち，ホップ数もそれほどの差がないか，どのようなネットワーク構造で用いても他より高い確率でトラフィック溢れ状態を起こしてしまい，ネットワーク構造の再計算が必要になる点で他の提

案手法に劣るといえる.

5.4 提案手法とECO-RPの協調動作(正規分布トラフィック)

先の実験で提案手法同士を比較した結果，どの手法もOSPFよりは優れる半面，DCNやDCPでは削減率が低く，DCNPでは削減率こそ高いものの高い確率でトラフィック溢れを招くという問題を抱えていた．しかしながらこれらの問題は，初期コストではなく，トラフィックに応じたコストの動的更新によって解決できる可能性がある．従来手法であるECO-RPはトラフィックに応じたOSPFコストの動的更新機能を備えており，ネットワーク内に存在する回線コストの半数を上昇，半数を下降させることで，トラフィックに応じた経路選択を実現している．またECO-RPの動作は初期コスト定義には影響を及ぼさないため，提案手法と連携した動作を行うことが可能である．よってここではECO-RPと提案手法が協調動作を行った実験の結果を分析し，もっともよい性能となる組み合わせを決定する．なおECO-RPには係数elという変化量調節係数が存在し，(0.2,0.4,0.6)と三種類の値が用意されているが，本研究では$el = 0.2$を採用した．

5.4.1 消費電力削減率の評価

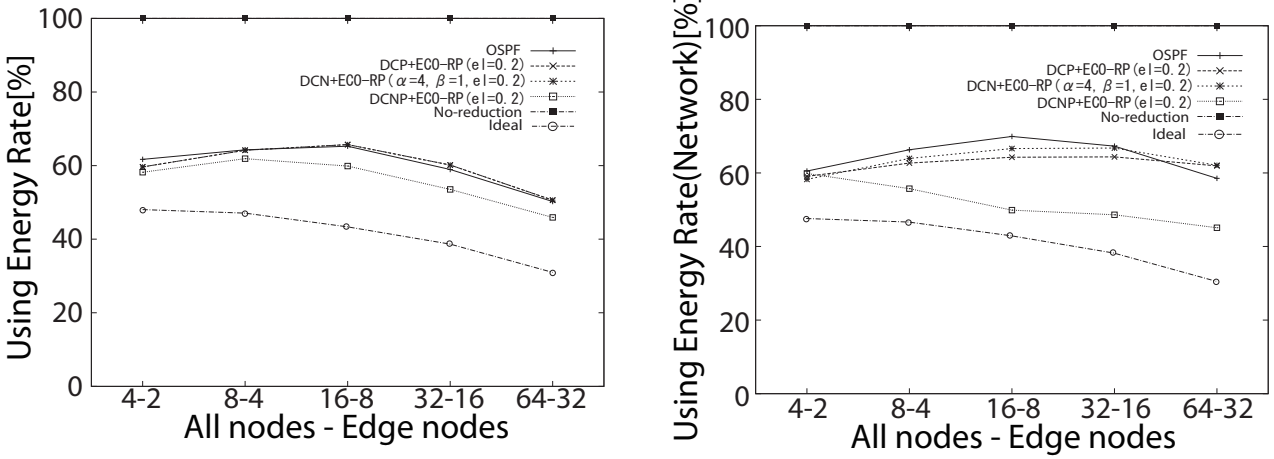


(a) 均一コスト使用時の最適化消費電力　　(b) 乱数コスト使用時の最適化消費電力

図 5.4.1: NSFNET T1の消費電力

図5.4.1は実験条件(A)に従い，NSFNET T1のネットワークトポロジを用いて計測したものである．両図におけるグラフNo-reductionは，一切の削減を行わないことを示しており，すべての状態において100パーセントとなっている．このとき(a)は初期OSPFコストに，コスト導出式を用い，均一な値を設定して行った削減実験の結果，平均消費電力パーセンテージを示している．グラフ全体を見てみると，もっとも下側に来たのはDCNP+ECO-RPつづいてDCP+ECO-RPとDCN + ECO-RPのグラフが重なって描写されている．もっとも削減率が低かったのは一切のコスト補正を用いない通常のOSPFであった．今，最も大きな差がついた”8”を例として見てみると，最良の性能であったDCNP+ECO-RPは78.99パーセント，続いてDCN+ECO-RPが86.52パーセント，DCP+ECO-RPが87.43パーセントとなったのに比べ，OSPFは89.01パーセントと上回った．

次に，初期コストに乱数を用いた実験結果 (b) を見ていく．各々のグラフは初期 OSPF コストに 1 から 100 の乱数を設定し，実験して得られた結果をプロットしたものである．もっとも結果がよかったのは DCNP+ECO-RP，続いて DCP+ECO-RP と DCN ＋ ECO-RP のグラフが重なっている．形状は均一コスト使用時と同じだが，乱数コスト使用による結果では DCNP+ECO-RP の消費電力が下降していることが分かった．もっとも差が大きかったのは”8”のときで，DCNP+ECO-RP は 64.39 パーセント，続いて DCN+ECO-RP が 80.77 パーセント，DCP+ECO-RP が 81.85 パーセントとほぼ同一である．乱数コスト使用時に性能低下を示す OSPF は 87.07 パーセントであった．これらの結果より，NSFNET T1 ネットワークにおける電力削減ではどちらのコスト設定を用いても DCNP と ECO-RP の連携がもっとも優れた結果を残すことがわかった．



(a) 均一コスト使用時の最適化消費電力

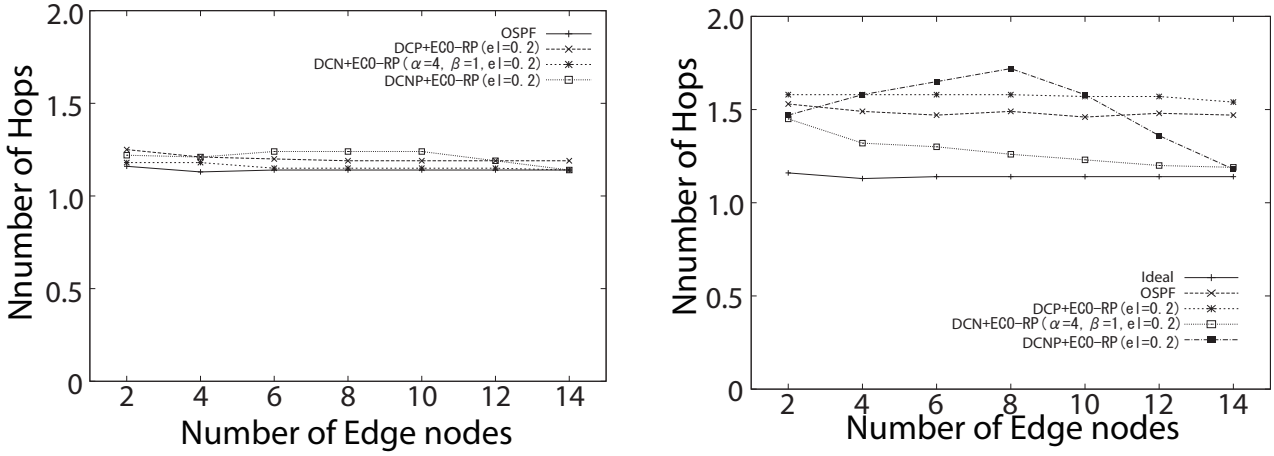
(b) 乱数コスト使用時の最適化消費電力

図 5.4.2: ランダム生成ネットワークの消費電力

図 5.4.2(a) は実験条件 (B) に従い，ネットワークの構造をランダムで生成して行った電力削減実験の結果である．図中には，消費電力削減を行わない手法の例として，No-reduction のグラフが描かれており，全ての状態で 100 パーセントを記録している．今，それぞれのグラフを見てみると，もっとも消費電力が低かったのは DCNP+ECO-RP であり，常に他のグラフの結果を下回っている．また OSPF，DCP+ECO-RP，DCN+ECO-RP のグラフはほぼ重なっており，優劣が付けられる状態とは言い難い．今”16-8”の結果を比較すると DCNP+ECO-RP が 59.88 パーセント，OSPF および残りの手法は 65 パーセントに固まっている．この結果より，初期 OSPF コストが均一なネットワークにおいては，先の結果と同様に DCNP の性能が高く，たとえ ECO-RP と連携したとしても，削減率は優れたままであることがいえる．

次に (b)，乱数コストを設定した際の最適化消費電力パーセンテージについて見ていく．(a) と比較して，各々の差がより顕著になっているが，DCNP+ECO-RP がもっとも優れた結果を出し続けていることについては同様である．先と同様”16-8”に焦点を当てた場合，もっとも優れた値は DCNP+ECO-RP で 49.88 パーセント，次いで DCN+ECO-RP が 64.26，DCP ＋ ECO-RP が 66.63，OSPF が最悪の結果として 69.92 を記録している．この結果より，DCNP は乱数コスト時の最適化に強く，ECO-RP との連携した際の消費電力削減能力も，他の従来手法に比べて高いといえる．

5.4.2　ホップ数の評価

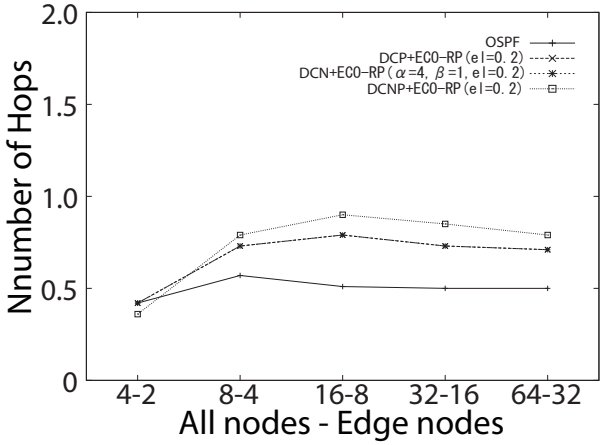


(a) 均一コスト使用時の平均ホップ数　　(b) 乱数コスト使用時の平均ホップ数

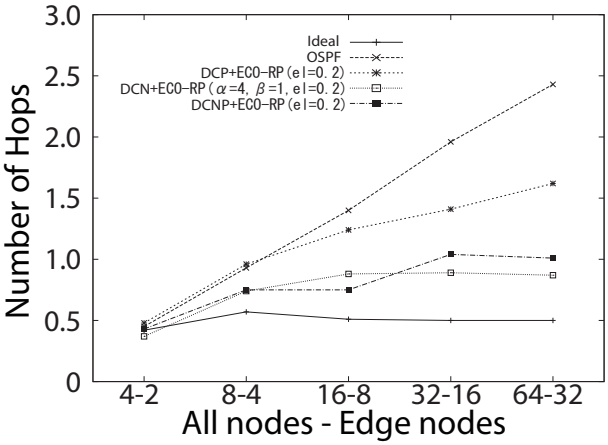
図 5.4.3: NSFNET T1 の平均ホップ数

図 5.4.3 は NSFNET T1 のネットワークトポロジを用いて最適化実験を行った際の，平均ホップ数をプロットした図である．実験条件は (A) に従っている．先の実験結果で DCNP のホップ数だけが突出していたように，この実験でも，DCNP+ECO-RP は他と比べてわずかに平均ホップ数が高くなっている．しかしながら，”8” の際の数値は DCNP が 1.24，理想の値である OSPF が 1.14 と，差は 0.1 であった．これは ECO-RP のコスト動的更新機能が，DCNP の消費電力的に不適切な機器を迂回する機能を弱めたため，最短経路を通る経路に近づいたためだと考えられる．

一方，初期コストに乱数を用いた (b) では全てのグラフが大きく変動した．今，先と同様に”8” の値を見てみると，”Ideal” の値が 1.17 であるのに対し，もっとも近かったのは DCN+ECO-RP の 1.26 であった．続いて OSPF の 1.49，DCP+ECO-RP の 1.58，最悪の結果になったのは DCNP+ECO-RP の 1.72 であった．コストに乱数を用いたことで，さらに DCNP の経路集約性が強化され，より消費電力の小さい経路にトラフィックが集中した結果，ホップ数は余計にかかる構造になってしまったと考えられる．また DCNP+ECO-RP の値は”8” を境に減少しているものの，全体的に見れば高い値を維持しており，ほぼ一定で変化のない DCN+ECO-RP のほうが性能がよい．このことから NSFNET T1 ネットワークを最適化する際，もっとも優れるのは DCN+ECO-RP であり，DCNP+ECO-RP は平均ホップ数の面では不適切であるといえる．



(a) 均一コスト使用時の平均ホップ数



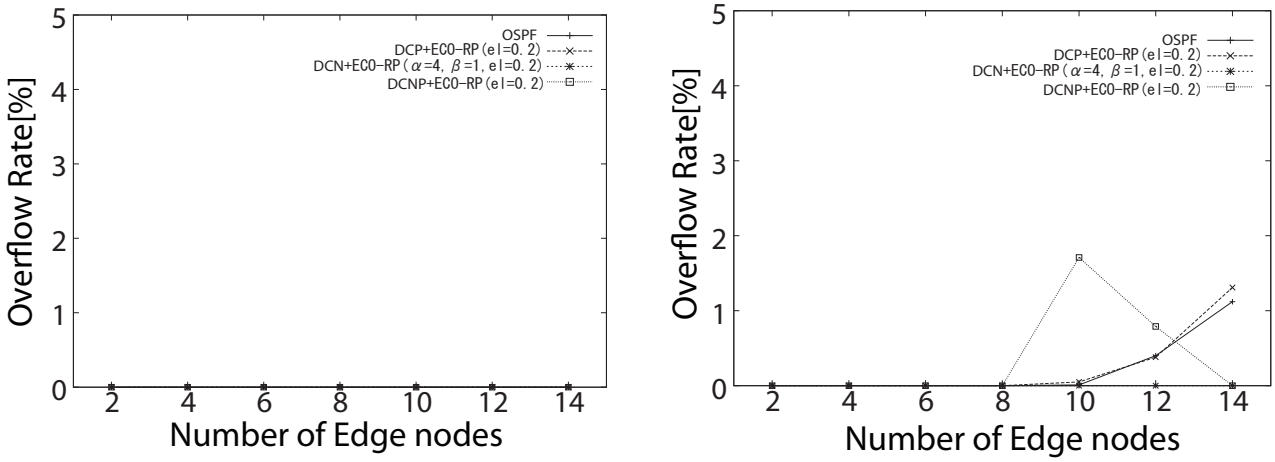
(b) 乱数コスト使用時の平均ホップ数

図 5.4.4: ランダム生成ネットワークの平均ホップ数

次に図 5.4.4，ランダム生成されたネットワークの平均ホップ数を見てみる．実験条件は (B) に従っている．まず (a) 均一コスト使用時の平均ホップ数を見てみる．図中 OSPF が理想のグラフを描いているが，他の手法がこの線に近づくことはない．全体的な変化では，DCN+ECO-RP と DCP+ECO-RP がほとんど重なっており，もっとも高かったのは DCNP+ECO-RP であった．今，16-8 を見てみると OSPF が 0.5，DCP+ECO-RP が 0.77 に DCN+ECO-RP が 0.79 とほぼ同一．DCNP+ECO-RP は 0.9 となった．均一コストを初期コストとして使用し実験した場合，差こそ大きくないものの，DCNP+ECO-RP は他手法に劣るといえる．

次に (b)，初期コストに 1 から 100 の乱数を設定した場合を見てみる．全体的に見てみると，理想の値”Ideal”の平均ホップ数が一定で変化していない一方，他のグラフ，とくに OSPF と DCP+ECO-RP はネットワーク規模に比例するように増大している．もっとも差が広がったのは”64-32”のときであり，”Ideal”の値 0.5 にもっとも近かったのは DCN+ECO-RP の 0.87，次いで DCNP+ECO-RP の 1.01 であった．ネットワーク規模の広がりとともに明らかな増加を見せた DCP+ECO-RP は 1.62，最悪の結果を記録したのは OSPF で 2.43 であった．OSPF による最短経路設定は，乱数コストが使用されるときには一切保障されないため，このような結果になると考えられる．また DCP は DCNP のように，エッジノードが隣り合う際の補正がないため，ネットワークノードが多くなればなるほど，遠回りが増えてしまい，また ECO-RP による意図しないコスト変動も相まって，平均ホップ数が上昇していると考えられる．

5.4.3　トラフィック溢れ発生率の評価

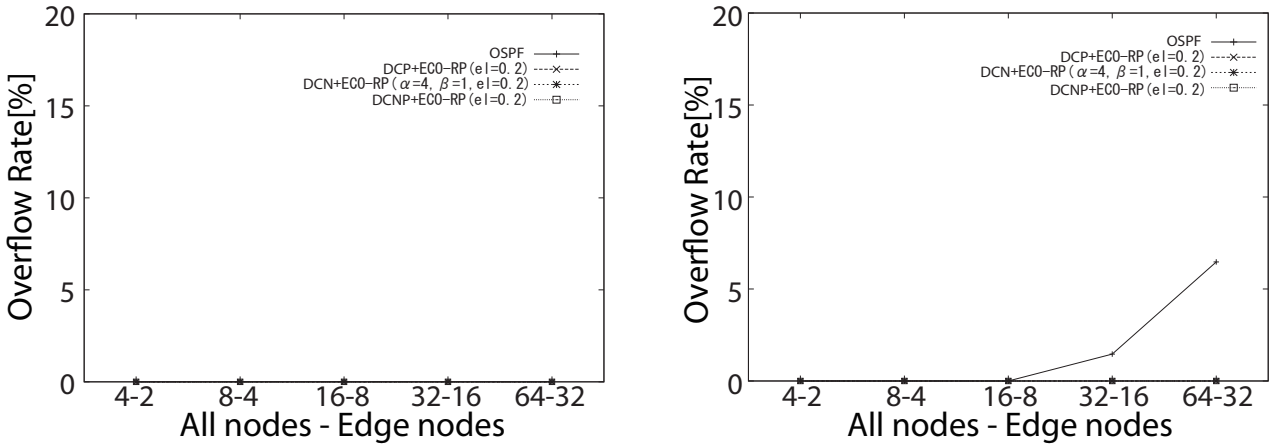


(a) 均一コスト使用時の発生率　　(b) 乱数コスト使用時の発生率

図 5.4.5: NSFNET T1 のトラフィック溢れ発生率

図 5.4.5 は NSFNET T1 のトポロジを用いて最適化実験を行った際の，トラフィック溢れ発生率を示している．このとき (a) はネットワーク回線全体を均一なコストに設定して実験を行った結果のグラフである．OSPF および提案手法の ECO-RP 連携では，全ての値が 0 になり，一切のトラフィック溢れが見られなかった．

一方 (b)，1 から 100 の乱数による初期コスト設定を用いたグラフでは”10”から発生率に変化が見られた．今，OSPF の結果を除くと，もっとも大きくなったのは”10”のとき，DCNP+ECO-RP の 1.71 である．またもっとも優れた結果となったのは DCN+ECO-RP で，常に 0 を記録していた．



(a) 均一コスト使用時の発生率　　(b) 乱数コスト使用時の発生率

図 5.4.6: ランダム生成ネットワークのトラフィック溢れ発生率

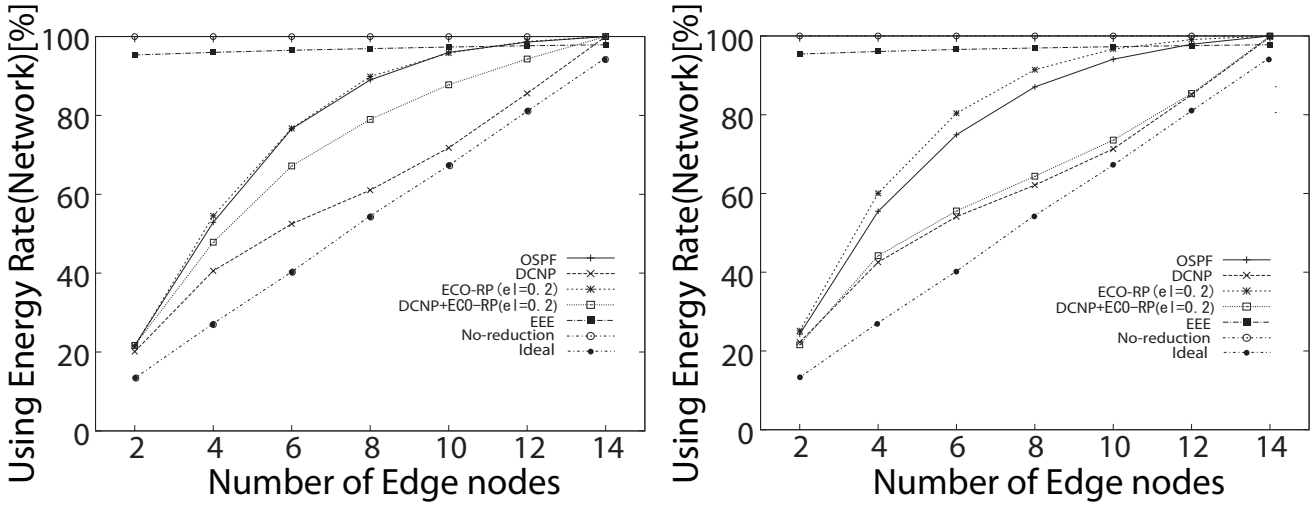
図 5.4.6 はランダムに生成されたネットワークを用いて最適化実験を行い，トラフィック溢れ発生率を測定した結果である．(a) 初期 OSPF コストに均一コストを使用した場合の結果では，全ての値が 0 となり，一切の変動がみられていない．またこの状態は (b) 初期コストに乱数を使用した場合の結果，でも同様で，”32-16”から上昇しているのは，ECO-RP 連携や補正を行っていない OSPF のものであり，ECO-RP との連携を行った提案手法は一切の上昇を見せなかった．このことより，ECO-RP と連携した提案手法は回線容量を超えるようなトラフィック集中が起こらなくなり，適度に通信が分散していると考えられる．

これらの結果を見てみると，差は小さいものの DCNP+ECO-RP の発生率が悪いように見える．しかしながら値も小さく，5.4.5(b) の”10”で確認された上昇は，その後”12”では下降に転じており，DCP+ECO-RP の結果と反転しているため，必ずしも結果が悪いといえるものではない．またこのデメリット以上に，DCNP+ECO-RP の消費電力削減能力が高い可能性もある．よって本実験で結論は出さず，もっとも消費電力削減能力の高かった DCNP，および DCNP+ECO-RP を合わせて，EEE と ECO-RP と比較することで，従来手法との差を導きだし，評価を行うこととした．

5.5 提案手法DCNPと従来手法(正規分布トラフィック)

提案手法同士の比較，およびECO-RPとの連係動作の比較により，もっとも消費電力削減効果の高く，連携性能の高い手法はDCNPであることが分かった．またDCNPの問題点であった高い発生率も，ECO-RPと組み合わせることで減少させることができた．この結果を用い，従来手法であるEEE, ECO-RP単体との性能比較を行う．消費電力削減率，平均ホップ数，トラフィック溢れ発生率の比較を行い．それぞれの手法の特徴の考察，もっとも優れた効果を持つ手法の決定を行う．

5.5.1 消費電力削減率の評価



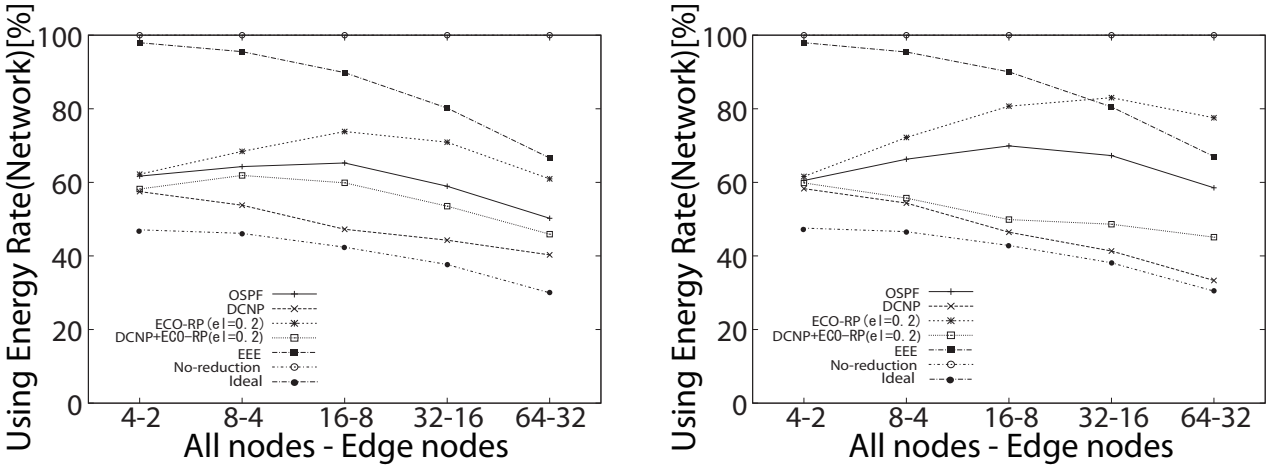
(a) 均一コスト使用時の最適化消費電力　　(b) 乱数コスト使用時の最適化消費電力

図 5.5.1: NSFNET T1の消費電力

図5.5.1は各手法の消費電力をパーセンテージで示したものである．NSFNET T1をトポロジを用い，回線コストにOSPF導出式を用いた均一の値を設定した他は，シミュレーション(A)の条件に従って行った実験結果になっている．これまでの比較と同様に，x軸はエッジノード数，y軸は消費電力のパーセンテージである．また，一切の削減を行わない状態をNo-reductionのグラフで示す．ここで(a)は初期コストの設定にOSPFコスト導出式を用い，回線コストを均一に設定した場合の結果を示している．今，エッジノード数”8”を見てみると，もっとも性能がよかったのはDCNPで61.04パーセント，反面，性能が悪かったのはEEEの96.95パーセントであった．またグラフ全体で見てみると，EEEの値は常に最大消費電力近くにあり，OSPFとECO-RPはほとんど同じ消費電力をたどっている．どのグラフも最終的に100パーセントへと収束していくが，唯一EEEのみは”14”で消費電力が97.92パーセントと，100パーセントにはならなかった．このうちもっとも緩やかな上昇をたどったのはDCNPであった．

(b)は初期コストに1から100の乱数を用いて設定し，各手法の最適化電力のパーセンテージを導出したものである．今エッジノード数”8”を見てみる．この実験の結果では，DCNPが62.12パーセント，DCNP+ECO-RPが64.39パーセントと，ほぼ同じ値をとっている．また先の実験と同様にEEEの消

費電力がもっとも高く，96.94 パーセントとなった．全体を見てみると DCNP と ECNP+ECO-RP がもっとも低い消費電力となり，ほとんど同じ推移を示す他，単体の ECO-RP と OSPF も近い値を推移している．EEE には過剰ノードを削減する機能がないため他と比べて必然的に消費電力が高くなってしまうが，全てのノードがエッジノードになって削減が不可能になった場面でも，安定した削減が提供できるという利点もある．実際”14”の時点の消費電力パーセンテージは 97.79 パーセントであった．他が全て 99.5 パーセント以上と，100 パーセント近い値を示す状況と比較すれば効果は明らかである．



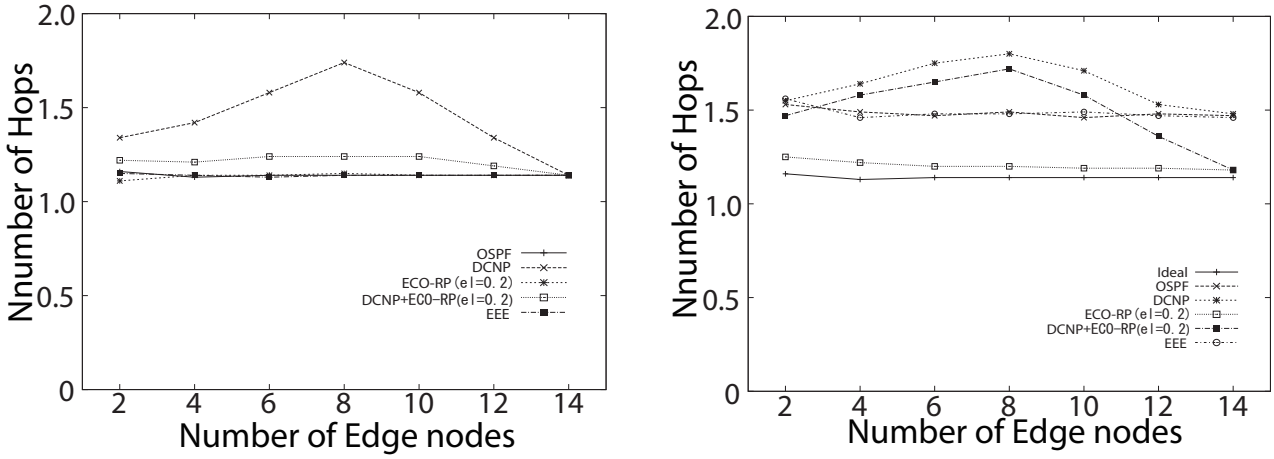
(a) 均一コスト使用時の最適化消費電力　　(b) 乱数コスト使用時の最適化消費電力

図 5.5.2: ランダム生成ネットワークの消費電力

次に図 5.5.2 について見てみる．こちらは全体ノードとエッジノード数の組み合わせによりランダムなネットワーク構造を生成して実験を行ったものである．x 軸は全体ノード数とエッジノード数の組み合わせ，y 軸はネットワークの消費電力をパーセンテージで示している．今，シミュレーション条件 (A) の NSFNET T1 のトポロジに近い数値である”16-8”を見てみると，もっとも低い消費電力は DCNP を用いた場合であり，44.27 パーセントであった．もっとも高くなったのは先と同じ EEE であり，80.18 パーセントである．また全体を見てみると，EEE の消費電力がネットワーク範囲の拡大とともに低下しているのがわかる．このことより，EEE による消費電力削減率はエッジノード数で変化することはなく，全体ノード数が増えると減少するといえる．またどの手法も”16-8”を境として，平均消費電力の割合が低下していることが分かった．

(b) はランダム生成ネットワークを用いて，消費電力の最適化を行ったものである．シミュレーション条件は (b) である．どの値を見てみても DCNP が優れた消費電力を記録し，もっとも小さかったのは”64-32”の 33.34 パーセントであり，同一ネットワーク規模で比較した場合，次いで DCNP+ECO-RP の連携が 45.09 パーセントと優れた結果を残した．また最悪の結果となったのは ECO-RP の 77.54 パーセントであった．NSFNET T1 トポロジでは最悪の結果を記録していた EEE は 67.01 パーセントと，ネットワーク規模が大きくなるほど消費電力のパーセンテージが下がっており，単体の ECO-RP よりも優れた結果を残している．これは ECO-RP が DCNP のような隣接情報や消費電力を考慮した機能を持たず，複雑かつランダムなコストが偏在する環境における削減に弱いためであり，EEE はネットワーク規模が大きくなればなるほど削減効率があがるため，このような結果になったと考えられる．

5.5.2　ホップ数の評価

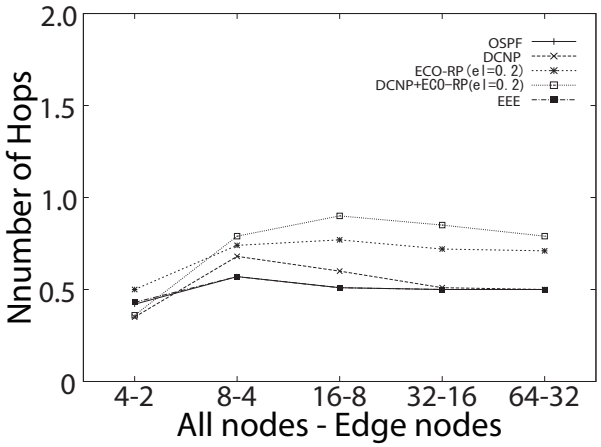


(a) 均一コスト使用時の平均ホップ数　　(b) 乱数コスト使用時の平均ホップ数

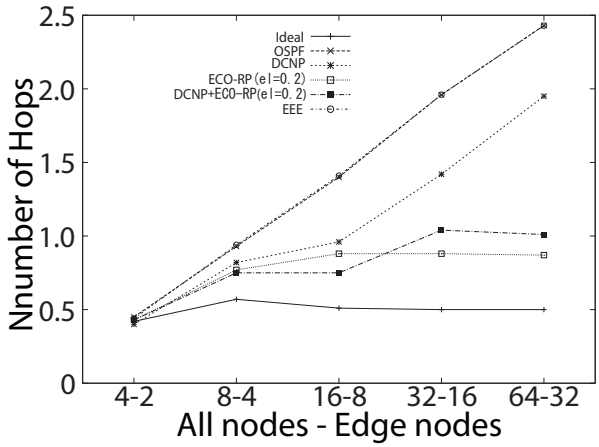
図 5.5.3: NSFNET T1 の平均ホップ数

消費電力の評価に続いて，ホップ数の評価を行う．図 5.5.3 は NSFNET T1 のトポロジを用いて実験したときの平均ホップ数を結果として示している．このうち (a) は初期コスト設定に OSPF コスト導出式を用い，全コストを 1 に設定して実験した際の平均ホップ数である．ネットワークの全コストが均一になっている際，OSPF によって求められる経路は最短であり，最良なホップ数であることは先にも述べた．したがって評価基準と方法は，対象となる手法のホップ数が OSPF の平均ホップ数に近ければ近いほど良いと判断できる．今図 (a) を見てみると，OSPF の平均ホップ数は 1.2 周辺で推移しており，全体的に変化していない．これにもっとも近いのは OSPF をそのままルーティングに用いる EEE であり，ついで ECO-RP であった．また DCNP はその経路集約性上，ホップ数は他に劣る傾向があり，エッジノード数”8”の時点では最大ホップ数である 1.69 を記録している．しかしながら DCNP+ECO-RP では”8”のとき 1.28 と平均ホップ数が改善されている．これは DCNP が持つ強すぎる経路集約性が，ECO-RP によって若干弱められたためだと思われる．

続く (b) はネットワークの初期コストに 1 から 100 の乱数を設定して実験した結果の平均ホップ数である．乱数をコストに用いた実験では全体的に平均ホップ数が増加することが，先の実験からわかっている．平均ホップ数を見てみても，理想の値”Ideal”に重なっている手法のグラフはない．この実験で最大ホップ数を記録したのはエッジノード数”8”のときであり，DCNP の 1.81 である．その次に DCNP+ECO-RP が 1.7 で続き，その次に EEE と OSPF のグラフが 1.45 で重なっている．このネットワークにおいてもっとも性能が良かったのは単独の ECO-RP で 1.22 であった．なおこの時の”Ideal”の値が 1.17 であることを考えると，本ネットワークにおいては，ECO-RP がもっとも優秀な平均ホップ数となることが確認できた．平均ホップ数による削減率選択を行う際は，DCNP は適さないといえる．



(a) 均一コスト使用時の平均ホップ数



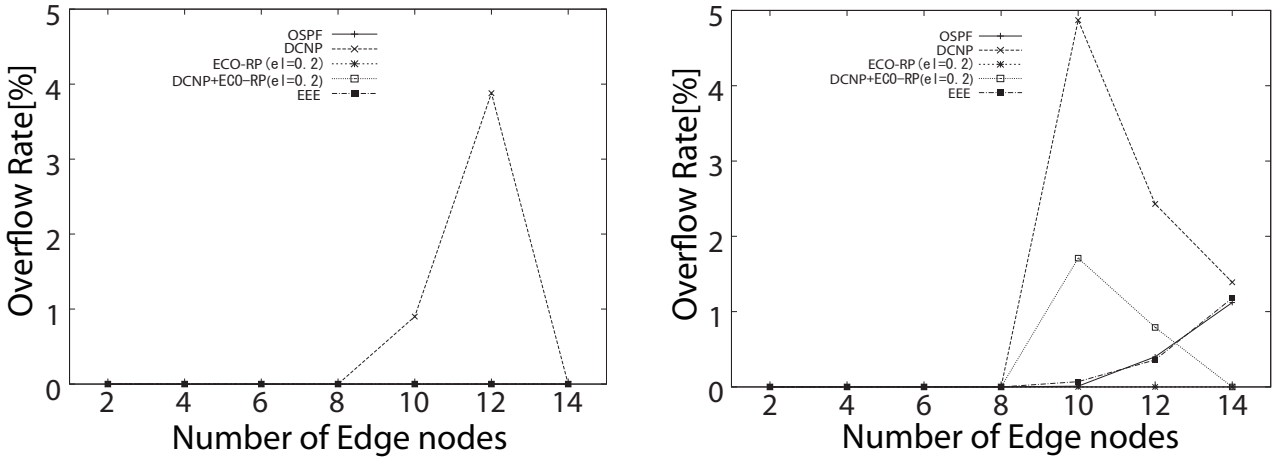
(b) 乱数コスト使用時の平均ホップ数

図 5.5.4: ランダム生成ネットワークの平均ホップ数

次に図 5.5.4 に移る．こちらはランダムネットワーク生成を用いた実験結果のグラフである．今 (a) 均一コスト時の平均ホップ数を見てみる．エッジノード数の割合に変化はないが，全体ノード数が増えているため，ネットワークの規模が拡大している．このグラフ中，もっとも大きなホップ数を記録したのは”16-8”であり，DCNP+ECO-RP の 0.9 である．このときも OSPF と EEE は 0.51 と最適なホップ数を取り続けている．ついで性能がよかったのは DCNP の 0.6 であった．DCNP と DCNP+ECO-RP が先の実験と逆転した結果になってしまったのは，先の実験において隣接情報の考慮が適切に反映される場面が少なかったからだと考えられる．NSFNET T1 のトポロジは相互接続されていないノードが多かったが，ランダム生成においてはいくらでも接続される可能性がある．よってエッジノード同士が隣り合った場合が多く発生したため，DCNP は削減性能で劣る DCNP+ECO-RP のホップ数を下回ったと考えられる．また単独の ECO-RP も DCNP のグラフに劣っている．無数に経路が存在する大規模ネットワーク環境では経路の集約性がホップ数の削減にも直結するため，より集約性能の高い DCNP が優れた結果を残したと考えられる．

次の (b) はネットワークをランダム生成して実験した平均ホップ数である．NSFNET T1 を用いた実験と違い，全体数が大きく変動するため，ホップ数も比例っして増大している．このうちもっとも大きな差がついたのは”64-32”であり，全体ノード数 64，エッジノード数 32 である．今，”Ideal”が 0.5 であるとき，もっとも近い平均ホップ数になったのは ECO-RP で 0.87，ついで DCNP+ECO-RP が 1.01 であった．また DCNP も大きく性能を落とし 1.95，もっとも最悪の結果になったのは OSPF と EEE のとき 2.43 であった．各回線の OSPF コストが異なるようになると，OSPF は設定されたコストによる最短経路を設定するため，ホップ数で最短となる経路を導出できなくなるためである．このことはルーティングに OSPF を用いている EEE も同じである．同様に DCNP も消費電力の高い機器を含む経路を確実に避けるため，自然とホップ数が多くなってしまうと考えられる．そのため半分のコストを増加させ，半分のコストを減少させる ECO-RP は，消費電力の高い機器を避けることがなく，平均ホップ数の増加もゆるやかであった．この結果より，提案手法である DCNP は平均ホップ数が高くなる傾向にあり，これを重要視する環境には不向きであると言える．

5.5.3　トラフィック溢れ発生率の評価

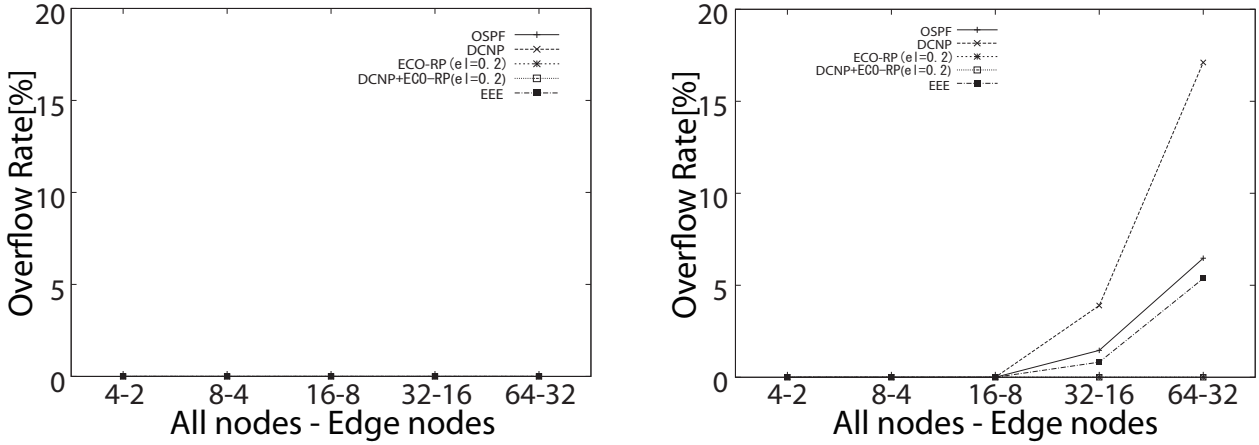


(a) 均一コスト使用時の発生率　　(b) 乱数コスト使用時の発生率

図 5.5.5: NSFNET T1 のトラフィック溢れ発生率

これまで平均消費電力パーセンテージ，平均ホップ数を比較してきたが，最後に各種法における回線のトラフィック溢れ発生率の平均値を比較する．図 5.5.5 は NSFNET T1 のトポロジを用いて実験したものである．このうち (a) は初期コスト設定に OSPF コスト導出式を用い，全コストを 1 に設定して実験した際の平均ホップ数を示している．x 軸エッジノード数を変化させても，多くの手法は 0 パーセントのまま動くことはなかった．全体を通して y 軸の発生率に動きがあったのは，DCNP のみである．このうちに最大を記録したのは”12”の 2.4 パーセントであった．発生率が”14”で 0 に戻る理由は，エッジノードが隣り合う際に一切の補正を行わない，という DCNP の隣接情報を利用したコスト修正により，ルーティングが OSPF を用いた状態と同じになるからである．

(b) は初期コストに 1 から 100 の乱数を用いて実験を行った際の，各手法におけるトラフィック溢れ発生率を示している．この図ではどのグラフもエッジノード数”8”までは変化がない．しかしながら”10”では大きな差がついている．このときもっとも大きな値になったのは，DCNP の 3.12 パーセントであり，次いで DCNP+ECO-RP の 0.69 パーセントである．DCNP を用いた初期 OSPF コストの変更では急激な経路集約が起こるため，回線容量をあふれる確率が大きくなっている．また”10”で 0 付近を記録している OSPF と EEE であるが，エッジノード数の上昇にともない発生率も上昇し，結果的に”14”では両者とも DCNP の値を超えている．このうちすべての状態を通して 0 であり，変化しなかったのは ECO-RP である．ECO-RP には全体経路における半数の経路コストを上昇させ，残り半数のコストを低下させるという働きがある．よって DCNP のような過度のトラフィック集中は発生せず，もともとトラフィックが集中している回線にもデータが流れにくくなることで，トラフィックの均一化が行われたと考えられる．また DCNP+ECO-RP における発生率の最大は”10”のとき 0.5 パーセント程度と，DCNP と比較してすぐれた結果を残している．DCNP 単体では急激なトラフィック溢れ状態を招くが，ECO-RP と連携して動作することで発生率を極めて効果的に減少させることができるといえる．



(a) 均一コスト使用時の発生率　　(b) 乱数コスト使用時の発生率

図 5.5.6: ランダム生成ネットワークのトラフィック溢れ発生率

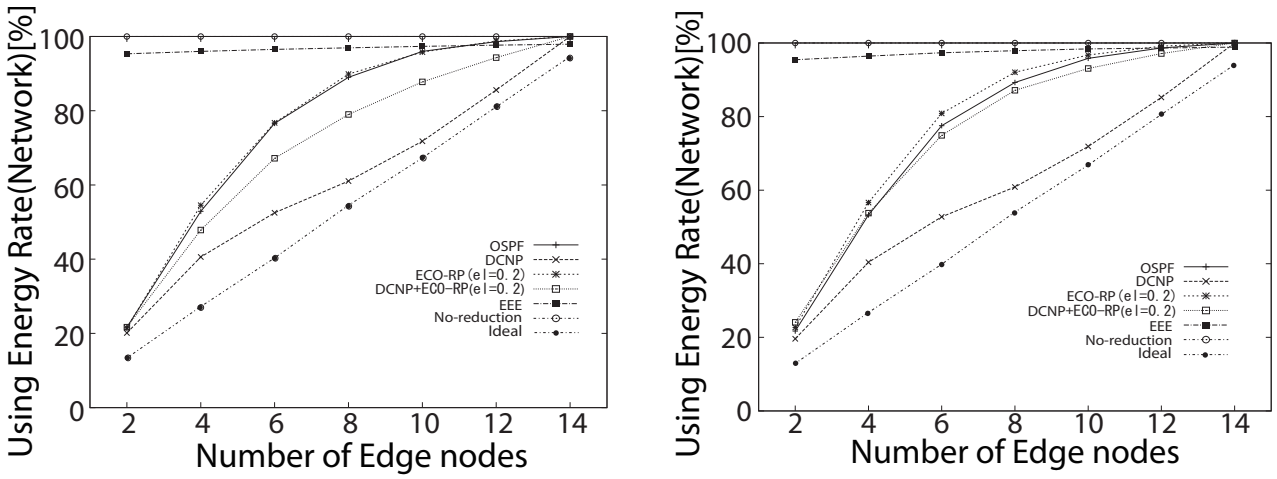
図 5.5.6 はネットワーク構築に条件 (B) のランダム生成を用いたものである．このうち (a) グラフを見てみても，x 軸の変化による値の変動は起こっていない．結果としては全てが 0 であり，トラフィック溢れは起こらないと確認できる．この図を図 5.5.4(a) のグラフ形状と比較することで，OSPF コストが均一なネットワークにおいては，ネットワークの規模よりも，エッジノード数の変化によって発生率が異なるということが確認できた．

次の (b) は乱数コスト使用時のトラフィック溢れ発生率を示している．OSPF コストが均一に設定された環境では，発生率がすべて 0 であり変化がなかったが，乱数を用いることで，”32-16”以降に変化が起こるようになった．このうち最大値を記録したのは”64-32”のとき，DCNP の 17.11 パーセント，ついで OSPF の 6.47 パーセント，同様に OSPF を利用する EEE の 5.37 である．全体を通して 0 パーセントを維持し変化がなかったのは DCNP-DCO-RP と ECO-RP の 2 つであった．ECO-RP は初期 OSPF コストのいかんにかかわらず発生を抑制する機能に優れ，さらに本来は経路集約性の強い DCNP も ECO-RP と連携することで発生率を抑制することができるといえる．

5.6 正規分布とコサイン波形トラフィックの比較

これまでの比較実験は，正規分布を元としたトラフィックモデルを基準に測定した結果を用いて行ってきた．しかしながら ECO-RP のようにトラフィックの数値に応じてコストの値が変動するアルゴリズムの場合，一つのトラフィックもでるだけの比較では，その特徴や能力をつかみきれないことがある．そこで比較実験の最後に，異なるトラフィックを用いた実験測定を行い，各手法の消費電力，ホップ数，トラフィック溢れ発生率にどのような違いが起こるのかを分析する．本項では，基準である OSPF，提案手法でもっとも性能のよかった DCNP，トラフィックに応じた動的 OSPF コストを実装した ECO-RP，DCNP と ECO-RP の連携，ポート毎の消費電力削減を実現する EEE のグラフについて比較を行う．

5.6.1 平均消費電力パーセンテージの比較

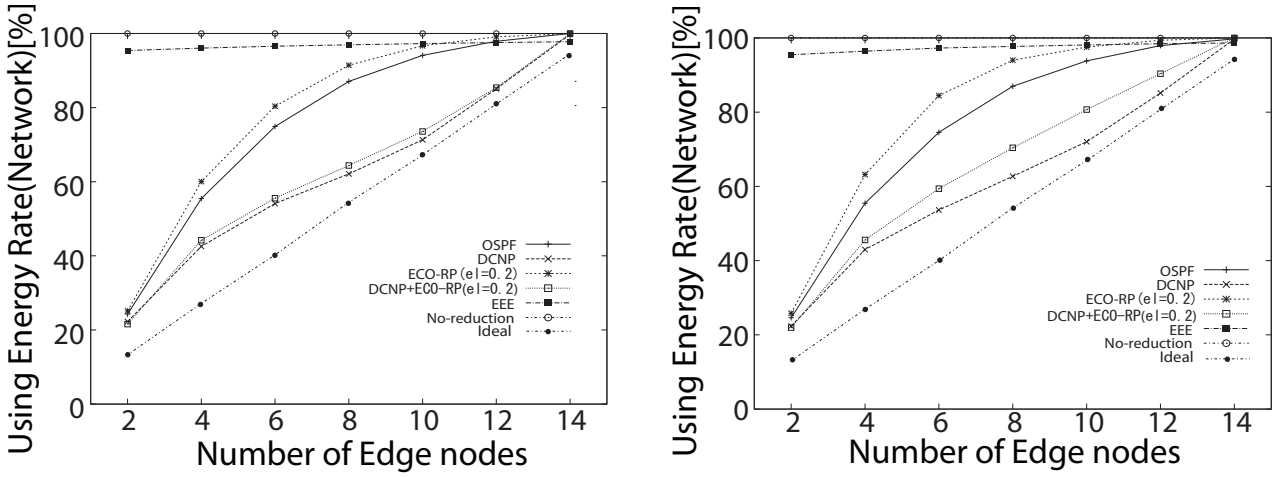


(a) トラフィックに正規分布を使用　　(b) トラフィックにコサイン波形を使用

図 5.6.1: NSFNET T1(均一コスト使用時) の平均消費電力

図 5.6.1 は NSFNET T1 トポロジを用い，全てのコストに均一な値を設定して実験，導出を行ったものである．(a) は正規分布，(b) はコサイン波形を用いた結果となっている．コスト導出にトラフィックの値が影響しない OSPF，DCNP，EEE では，わずかな差こそあるものの，どちらも同じ形状のグラフになっているのがわかる．一方 ECO-RP を使用した際の消費電力には差がみられた．まず ECO-RP 使用時，形状は同じものの差がみられた．もっとも大きな差になったのはエッジノード数”6”のとき，(a)76.06 パーセントに対し，(b)80．86 パーセント．4.82 パーセントの差である．他のすべての差はこの範囲内に収まるほか，NSFNET における消費電力 5 パーセントの程度の差は，1 パーセントあたりの消費電力がそれほど大きいものではないことから考えても，ほとんどないものと考えられる．続いて DCNP+ECO-RP のグラフを見てみる．全体を見てみると (a) では DCNP と OSPF の間に位置しているグラフが，(b) では OSPF がわに近づいてしまっている．このうち最も大きな差がついたのは”6”のとき，(a) で 65.22 パーセント，(b) で 74.88 パーセントであった．差は 9.66 パーセントである．およそ 10 パーセント近い差になってくると，小さいネットワーク規模でも比較的大きな差になってしまう．し

かしながら消費電力の順位はDCNPに続いて二番目であることに変わりはなく，DCNP ＋ ECO-RPでも変わらぬ優位性があることは事実である．この結果よりECO-RPや，DCNP+ECO-RPではOSPFコストが均一でありさらにトラフィックの上限値が高くなると電力削減効果が落ちるが，既存手法よりは優れた削減能力を提供することがわかった．

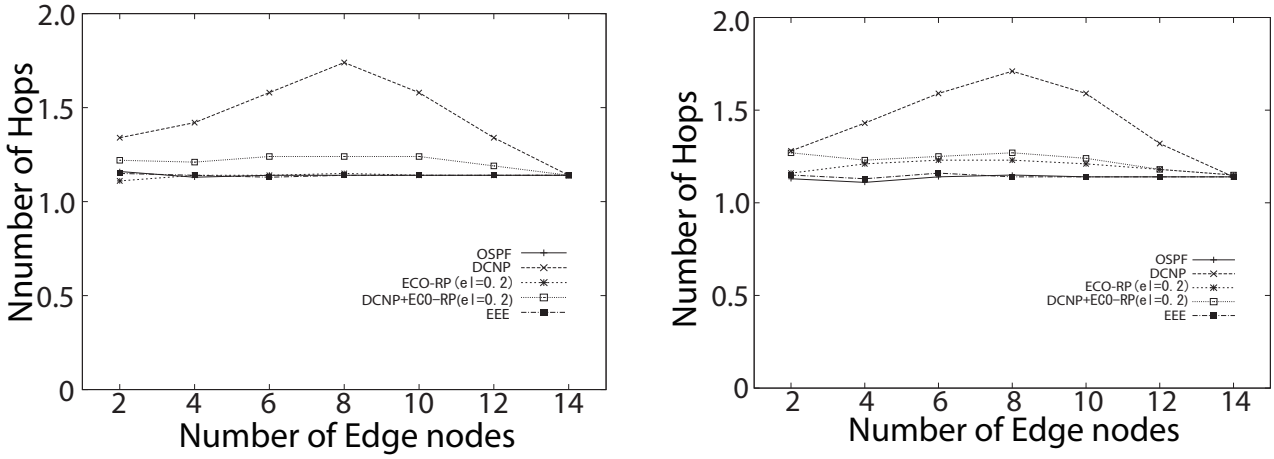


(a) トラフィックに正規分布を使用　　(b) トラフィックにコサイン波形を使用

図 5.6.2: NSFNET T1(乱数コスト使用時) の平均消費電力

図5.6.2は先と同様にNSFNET T1 トポロジを用いて実験した結果である．こちらは初期OSPFコストに1から100のランダムな数値を与えている．なお(a)(b)の条件も同じで，(a)は正規分布，(b)はコサイン波形をトラフィックモデルとして用いた結果になっている．まずECO-RPのグラフ全体を見てみると，どちらも削減率は低く，OSPFのグラフの上側を通っているが，(b)のほうが若干大きい値になっているように見える．このうち両者の差がもっとも大きくなったのは”6”のとき，(a)78.06パーセントにくらべ，(b)84.45パーセントと6.39パーセントの差である．均一コスト時と同じく，若干の差があるように見える．一方ECO-DCNPのグラフも(b)では若干の上昇がみられるものの，形状はほぼ同じでDCNP側に近い値を保っている．もっとも差が大きくなったのは”10”のとき，(a)74.09パーセント，(b)80.69パーセントと，差が6.6パーセントである．従来手法ECO-RPとDCNP-ECO-RPにおいて，もっとも消費電力削減率の差が大きくなる時の値を比較すると，(a)では”8”のときECO-RPが90.43パーセントで9.57パーセントの削減，DCNP ＋ ECO-RPは64.66パーセントで35.34であり，3.69倍の性能差でDCNP+ECO-RPが優れる．(b)では”6”のときECO-RPが84.45パーセントで15.55パーセントの削減，DCNP ＋ ECO-RPが59.38パーセントで40.62パーセントとなっており，2.61倍の性能差で，DCNP+ECO-RPが優れることがわかった．またDCNP+ECO-RPはどちらのグラフでもDCNPに続いて2番目に小さな平均消費電力パーセンテージとなっている．このことから乱数コストを用いて計算する際でも，どちらのトラフィックモデルを使用しても消費電力削減率を決めるパーセンテージの優劣に差はないことがわかった．

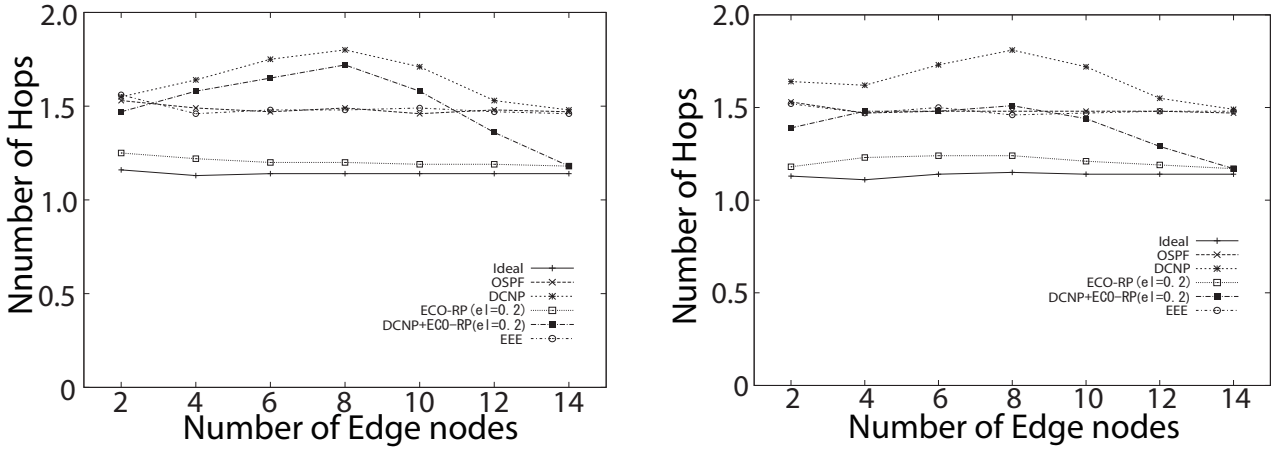
5.6.2　平均ホップ数の比較



(a) トラフィックに正規分布を使用　　(b) トラフィックにコサイン波形を使用

図 5.6.3: NSFNET T1(均一コスト使用時) の平均ホップ数

図 5.6.3 はネットワークに均一のコストを設定した際，最適化時に導出されるエッジノード間の平均ホップ数を示している．グラフ中 OSPF は常に最短経路を通る，理想のホップ数である．今 (a) に正規分布，(b) にコサイン波形のトラフィックを使用して導出したグラフを載せてあるが，このうち OSPF，DCNP，EEE はトラフィックによる経路変更がないためほとんど同じ値を通っている．ここでは ECO-RP と DCNP+ECO-RP を見ていくことになる．まず ECO-RP のグラフを見ていくが，(a)，(b) ともにほとんど変化していない．数値にすると全てが 1.2 付近に集中しており，0.05 の誤差内に収まっている．次に DCNP+ECO-RP を見てみる．こちらのグラフもほとんど重なっている．両者の誤差は同様に 0.05 以内に収まっておりほとんど一致しているといっても問題ない．両者の結果，コストが均一に設定されたネットワークでは，トラフィックモデルの違いによって起こる平均ホップ数の変動は，ほとんど確認できなかった．

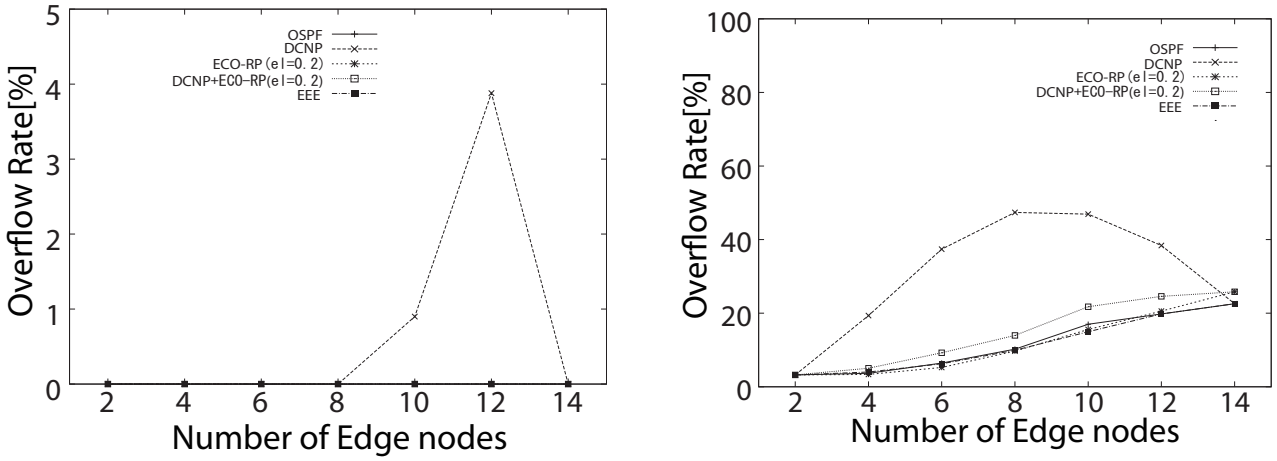


(a) トラフィックに正規分布を使用　　(b) トラフィックにコサイン波形を使用

図 5.6.4: NSFNET T1(乱数コスト使用時) の平均ホップ数

図 5.6.3 はネットワークに 1 から 100 の乱数コストを設定した際，最適化時に導出されるエッジノード間の平均ホップ数を示している．グラフにおける Ideal は確実に最短経路を通るグラフであり，もっとも小さいホップ数になっている．このグラフに最も近い手法が，優れたホップ数を持つ手法といえる．先と同様に (a) は正規分布，(b) はコサイン波形のトラフィックを用いている．今，トラフィックの影響を受けない OSPF，DCNP，EEE を除き，ECO-RP，DCNP+ECO=RP について考える．ECO-RP のグラフはどちらもほぼ同じ場所をなぞっている．もっとも差が大きかったのは”2”で，(a) のとき 1.31，(b) のとき 1.18 の 0.13 である．このほかの場所では誤差が 0.05 以下となり，ほとんど同一となった．一方 DCNP+ECO-RP のグラフには差が見られた．どちらも山なりのグラフを描いているが，もっとも差が大きかったのは”8”で (a) のとき 1.7，(b) のとき 1.51 と 0.19 の差で (b) が小さかった．その他の部分でも 0.1 ほどの差が開いた．(b) のトラフィックは (a) と TIME 数こそ同じだが，変化量と最大値が高い．このことからネットワークに乱数コストを設定し，なおかつ高低差が激しいトラフィックモデルの場合，DCNP と ECO-RP の連携時に平均ホップ数が低くなる以外は，変動がないことがわかった．

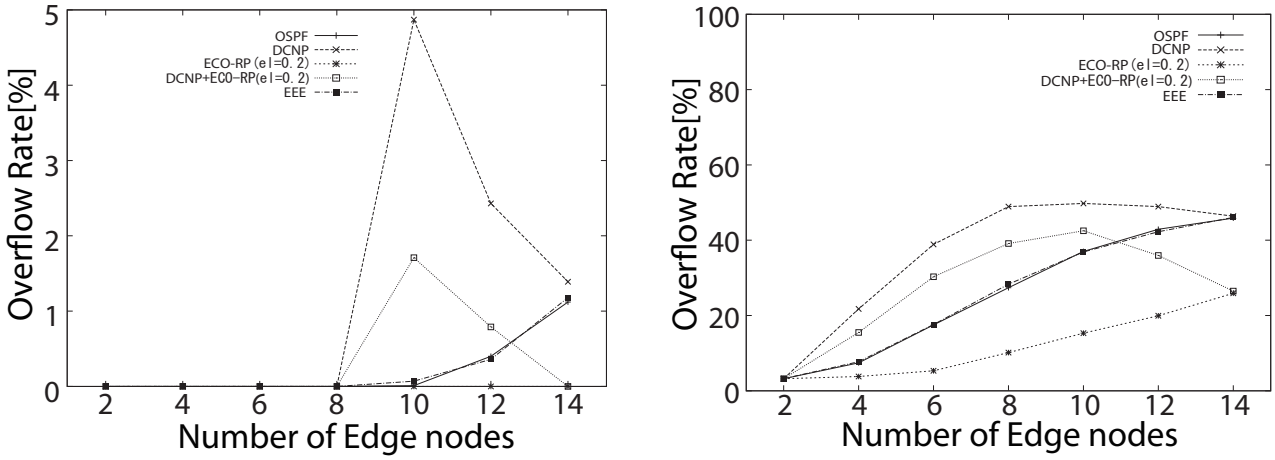
5.6.3　トラフィック溢れ発生率の比較



(a) トラフィックに正規分布を使用　　(b) トラフィックにコサイン波形を使用

図 5.6.5: NSFNET T1(均一コスト使用時) のトラフィック溢れ発生率

図 5.6.5 は NSFNET のトポロジに均一コストを設定して行った実験のトラフィック溢れ発生率を示している．(a) 正規分布のトラフィックモデルより，(b) のコサイン波形を用いたトラフィックモデルのほうが流量も変化も大きいため，よりトラフィック溢れ状態が発生しやすくなってしまう．そのため両者のグラフを同じ視点で測ることはできず，点による比較は難しい．しかしながらグラフ全体の傾向を見比べることはできる．(a) においては DCNP の発生率が単独で”10”から上昇している．この結果，DCNP は他手法に比べてもっとも発生率が高いということがいえるが，(b) でもそれは同様で，DCNP のグラフは山なりの線を描いており，他のグラフと 10 から 40 パーセント近い差がついている．このことから，ネットワークに均一なコストを設定した際の発生率は，どちらのグラフでも DCNP がもっとも高くなるという結果が得られた．



(a) トラフィックに正規分布を使用　　(b) トラフィックにコサイン波形を使用

図 5.6.6: NSFNET T1(乱数コスト使用時) のトラフィック溢れ発生率

図 5.6.6 は NSFNET ネットワークに乱数コストを設定して行った実験のトラフィック溢れ発生率を示している．ここでは (a) が正規分布，(b) がコサイン波形を用いたトラフィックモデルである．こちらも同様に比較を行ってみる．

どちらのグラフも DCNP および DCNP+ECO-RP は”10”を境に低下している．また (a) では 0 だった ECO-RP は (b) では最も低い発生率となっている．このことから大小関係を比較すると，どちらのグラフも DCNP が最大を記録し，次いで DCNP ＋ ECO-RP である．さらに (a) のグラフでは OSPF と EEE が”10”から上昇を開始しているのに対し，ECO-RP は全て 0 のままである．これは (b) における，もっとも小さい発生率を維持し続けた結果と同じである．この結果より，ネットワークコストに乱数を設定する実験において，トラフィックの大きさや変化量が変わっても，優劣の順位には差がなく，また DCNP は ECO-RP と連携することで，どちらのケースでもトラフィック溢れの発生率を抑えることができることがわかった．

5.7 まとめと考察

5.7.1 まとめ

本実験では各種法によるネットワーク消費電力削減を実装したシミュレータを利用し，提案手法同士，提案手法と ECO-RP の連携，DCNP と従来手法について比較してきた．まず提案手法同士の比較結果として，OSPF，DCP，DCN，DCNP の四手法を比較した．結果として消費電力平均パーセンテージは DCNP が最も低く，最も高い削減率となった．これは導出式コストを用いた場合，乱数コストを用いた場合に限らず，常に同じ結果であった．しかしながら DCNP を利用した初期 OSPF コスト変更は，効率の悪い機器を強制的に迂回させるという性能上，平均ホップ数とトラフィック溢れの発生率が他手法に劣るという結果が得られた．しかしながら全体的にみると平均ホップ数の差は小さく，運用にあたって障害となる実質的な問題点はトラフィック溢れの発生率のみである．よって提案手法同士の比較では，高い削減率を持つ DCNP が最も優れた結果を残したと結論付けた．

続いて，提案手法と ECO-RP との連係動作として，DCP+ECO-RP，DCN+ECO-RP，DCNP+ECO-RP のそれぞれを，通常の OSPF による経路削減と比較した．結果として，均一コストを用いた実験の平均ホップ数以外の条件において，OSPF の性能を下回った手法はなかった．また，電力削減効果が最も高かったのは全体を通して DCNP+ECO-RP であり，ホップ数については，ネットワーク形状において大きく値が分かれた．NSFNET T1 ネットワークのトポロジは DCNP による削減の結果，他の手法に比べてホップ数が増加する傾向にあったが，ランダム生成の場合はこの問題が表れていない．ランダム生成ネットワークの場合，均一のコストを用いた際は，OSPF を除くと DCN+ECO-RP がもっとも性能が良く，その他の手法もほぼ同じ値を示す結果となった．しかし乱数を用いて行った実験では，DCP+ECO-RP の性能は他の二種にくらべ大きく劣った．またトラフィック溢れ発生率の実験では，均一コストを用いた際の実験では全ての値が 0 となり変化がなく，乱数コストを用いた際も，ランダム生成ネットワークでは OSPF のみの上昇にとどまり，NSFNET T1 ネットワークを用いた実験でも，DCNP+ECO-RP，DCP ＋ ECO-RP が”10”以降にわずかに上昇した以外は，変化が見られなかった．そこで先に行った ECO-RP との連携実験の結果を踏まえ，従来手法との比較として，提案手法の中でも最も削減率が高かった DCNP，ECO-RP との連携で優れた結果を残した DCNP+ECO-RP を，従来手法である ECO-RP，EEE と比較した．

結果として，消費電力削減率の面では全てのケースにおいて DCNP が優れ，従来手法を上回る数値を残したが，NSFNET T1 ネットワークの最適化では平均ホップ数が高くなるという問題を残した．さらにトラフィック溢れの発生率では強すぎる経路集約性が問題となり，ネットワークを最適化する際に高い確率でトラフィック溢れを発生させてしまう結果になった．この問題点は規模が大きいネットワークかつ，回線の OSPF コストが異なる場合において，特に顕著である．しかしながら，DCNP と ECO-RP が連携して動作した場合には，消費電力の削減率は若干低下するものの従来手法よりも高く，平均ホップ数およびトラフィック溢れ発生率に大きな改善がみられた．また一般的にネットワークの規模が変わらない場合，EEE の削減率が低いケースが多いことから判断すると，EEE がもっとも性能が悪いということになるが，実際はそう結論付けることはできない．今，実際に図 5.5.1 で EEE による削減結果のグラフを見てみると，どの結果もエッジノード数の如何にかかわらずほぼ一定で推移，あるいは減少していることがわかる．また大きな特徴として，NSFNET T1 トポロジを用いて実験した際，すべて

のノードがエッジノードになった”14”のときでも，EEE は常に一定量の削減を行い続けている．他の手法が 100 パーセントを記録するのに対して，非常に安定した動作である．これは EEE がエッジノード数にほとんど影響されず，ネットワーク回線に発生している通信トラフィック量にのみ影響される電力削減手法であるためである．したがって EEE はきわめて安定した手法であるといえる．もし，全てのノードがエッジノードになる環境で使用されることが前提ならば，EEE は大きな効果を発揮するだろう．

最後にトラフィックの違いによる消費電力変動を確認するため，正規分布とコサイン波形のトラフィックモデルを用いて実験を行い，消費電力，ホップ数，トラフィック溢れ発生率の平均をそれぞれ比較した．結論としてほぼすべての値に変化がないあるいは変化が小さく，大きく増加したトラフィック溢れ発生率や，変化があった一部の項目に対しても，他のグラフとの優劣は変化がないという結果となった．OSPF コストの値，およびトラフィックモデルの如何にかかわらず，手法における優劣比較には差がないといえる．

5.7.2 考察

今，各手法の実験から算出された結果のうち，もっともすぐれた削減率を比較するため，表 5.6.1 にあるネットワークの規模と 1 パーセントあたりの消費電力を使用して，各々の手法が削減した電力を一つの例から算出してみることにした．

表 5.7.1: NSFNET T1 での最適削減値

	消費電力削減率[%]	削減電力量[W]	平均ホップ数	溢れ発生率[%]
OSPF	12.93	94.13〜396.93	1.49	0
DCNP	37.78	275.04〜1158.35	1.80	0
ECO-RP	8.56	62.32〜262.45	1.2	0
DCNP+ECO-RP	35.61	259.24〜1091.8	1.72	0
EEE	3.06	22.28〜93.82	1.48	0

NSFNET トポロジを用いた実験の平均消費電力では，エッジノード数が結果に大きく影響してくる．エッジ数が少なければ削減率が極めて良くなり，多ければ悪くなる．よって単純に消費電力が小さいから最適といえるわけではない．そこでこの考察では均一または乱数コストを用いた実験のうち，OSPF の値と DCNP の値の差がもっとも広くなるエッジノード数を，評価の指針とすることにした．OSPF は全ての手法の基準となる基礎的なアルゴリズムであり，提案手法である DCNP との差が広いということは，それだけ効果的に削減ができているということだからである．この基準に沿って選択した結果，図 5.5.1 のエッジノード数”8”が選択された．表 5.7.1 はこの時の状態を一覧化して表にしたものである．定義表には書かれていないが，全体ノード数 14 のとき，機器の種類によるネットワークの消費電力変動は，1 パーセントあたり 7.28 ∼ 30.66W である．このとき，最も優れた電力削減率となったのは DCNP の 37.78 パーセント，最悪は EEE の 4.06 パーセントである．最初に提案手法である DCNP の性

能について見ていく．DCNP は従来手法よりも優れた消費電力削減率を実現している．またエッジノード数が”8”のときはトラフィック溢れ発生率も他と同じ 0 である．このとき平均ホップ数が 1.8 であり，若干他よりも高くなっているが大きな差はついていない．また結果より NSFNETDCNP を用いた場合，OSPF の 2.92 倍，ECO-RP の 4.41 倍，EEE の 12.25 倍の削減率が期待できる．よって，DCNP は従来手法よりも優れた成果をもつ手法であるといえる．続いて DCNP+ECO-RP を見ていく．DCNP の初期コスト変更と，ECO-RP の OSPF コスト動的更新を合わせた手法では，削減率が 35.61 パーセントと，わずかに DCNP 単体を下回った．しかしながら DCNP よりも平均ホップ数がわずかに低く，トラフィック溢れ発生率も 0 パーセントである．従来手法と比較すると，DCNP+ECO-RP は OSPF の 2.75 倍，ECO-RP の 3.73 倍，EEE の 11.64 倍となり，DCNP には劣るものの，従来手法よりも優れることがわかった．このことから，DCNP を用いた初期コストは NSFNET の構造を効果的に最適化でき，消費電力を削減することができたといえる．

表 5.7.2: ランダム生成トポロジでの最適削減値

	消費電力削減率[%]	削減電力量[W]	平均ホップ数	溢れ発生率[%]
OSPF	41.49	1354.23〜5815.24	2.43	6.47
DCNP	66.66	2175.78〜9343.07	1.95	17.11
ECO-RP	22.46	733.09〜3147.99	0.87	0
DCNP+ECO-RP	54.91	1792.26〜7696.19	1.01	0
EEE	33.99	1101.28〜4764.04	2.43	5.73

表 5.7.2 はもっとも消費電力削減性能が高い状態として，DCNP と従来手法を比較した項の中にある図 5.5.2(b) の”64-32”を選択し，消費電力パーセンテージを計算したものである．削減電力の左側が最小削減電力，右が最大削減電力を示している．このうち最も優れた電力削減を行ったものは DCNP の 66.66 パーセントであり，最悪は ECO-RP の 22.46 パーセントであった．DCNP と従来手法の消費電力削減率を比較してみると，OSPF で 1.61 倍，ECO-RP では 2.97 倍，EEE とは 1.96 倍の性能差がついた．また ECO-RP との連携結果を見てみると ECO-RP 単独とは 2.44 倍，EEE とは 1.62 倍の差で勝ることが分かった．この結果より DCNP は従来手法と比べ，高い電力削減率を誇る優れた手法であるといえる．しかし ECO-RP は優れた平均ホップ数を持っており，トラフィック溢れの発生率が 0 パーセントという特徴も持っている．反面 DCNP はその強い経路集約性上，17.11 パーセントという高い確率を記録してしまった．しかしながら二番目に削減率の高い DCNP+ECO-RP は，平均ホップ数も 1.01 と低く，なおかつトラフィック溢れの発生率が 0 パーセントという，従来手法よりも優れた成果を残している．DCNP+ECO-RP の電力削減率を従来手法と比べると，OSPF では 1.32 倍，ECO-RP とは 2.45 倍，EEE とは 1.62 倍の効果を持つことがわかった．このことから DCNP が持つ高いトラフィック溢れの発生率は，ECO-RP と連携して動作することで補うことができ，なおかつ従来手法よりも高い電力削減率を維持することができるといえる．以上の結果より DCNP はランダム生成されたネットワークにおいても従来手法よりも効果的にネットワークを最適化し，消費電力を削減できることが確認された．

5.8 おわりに

本章では 4 章で提案したネットワーク構造を最適化する OSPF コスト設定手法を Dijkstra 法のシミュレーション上に実装し，従来手法である OSPF，ECO-RP，EEE を用いて，平均消費電力パーセンテージ，平均ホップ数，トラフィック溢れ発生率の平均値を比較した．NSFNET およびランダム生成ネットワークのどちらにおいても，DCNP は優れた消費電力削減率を記録したが，トラフィック溢れの発生率が他の提案手法，従来手法よりも高いという問題も明らかになった．この問題については ECO-RP との連携を行うことでトラフィック溢れを抑制することが可能だと結論付けた．また異なるトラフィックモデルを利用しても，各々の手法による優劣の順位は変わらず変化がほとんど見られないということもわかった．よって DCNP および DCNP と ECO-RP の連携手法は，他の従来手法と比較して優れた性能を発揮したと言える．

第6章　結論

近年，動画配信やクラウドコンピューティングによる通信トラフィックの増大にともない，それを処理するデータセンターの増加が続いている．中でもネットワーク機器の数は年々増え続けており，2006年末に出された経済産業省の予測調査では，ネットワーク機器の消費電力が現在から2025年までに5.2倍になるという問題提起がなされている．本研究では増え続けるネットワーク機器のうち，電力的に問題がある，あるいは不必要な機器を選択し，電源を停止することによってネットワークの消費電力を削減することを目的としており，省電力なネットワーク構造を導出するための手法を提案した．本稿で取り上げた従来研究には次のような問題点がある．

- 経路選択に機器の消費電力が考慮されない
- 経路選択に機器の性能や情報が考慮されない
- OSPF コストにより不適切な経路選択が発生する可能性

これらの問題は性能の異なる機器が無数に存在する現実のネットワーク環境においては，ネットワーク消費電力の削減率に直結する問題であり，早急に解決する必要がある問題である．本研究では経路選択に機器の消費電力，また隣接機器の性能を考慮し，異なる機器が偏在する環境においても効果的な削減を行うための手法を提案した．また実際のネットワーク機器を用いて消費電力モデルを作成し，シミュレーションを行った．本稿で提案したDCP，DCN，DCNPはいずれも消費電力情報あるいは隣接機器の情報を用いて回線毎に設定される初期OSPFコストを再定義し，経路情報を変更することでネットワークの構造を最適化する．これによって電力的に不適切，あるいは運用に不必要な機器を停止させ，ネットワーク全体の消費電力低下を実現している．このうち最も電力削減性能が良かったのはDCNPであり，従来手法と比較した結果，NSFNETトポロジではOSPFの2.92倍，ECO-RPの4.41倍，EEEの12.25倍の削減率を記録した．またランダム生成ネットワークを用いた最適化実験ではOSPFで1.61倍，ECO-RPでは2.97倍，EEEには1.96倍の差がついた．またネットワークの動的更新を実現するため，トラフィック情報を用いてOSPFコストを動的に更新する手法との連携を視野に入れた設計を行った．DCNP単体ではトラフィックに応じた動的な構造切り替えはできず，電力的に最適であっても，トラフィックやホップ数的に最適ではない状態が起こりえる．そうした欠点を補い，トラフィックに応じて適切な構造変更が可能になる手法として，ECO-RPとの連携動作を考慮した．正規分布トラフィック使用時の実験で最も高かったトラフィック溢れの発生率は，ランダム生成ネットワークにおける17.11パーセントであったが，ECO-RPと連携することで0パーセントにまで引き下げることに成功している．この時の性能はOSPFで1.32倍，ECO-RPとは2.45倍，EEEとは1.62倍の差となり，DCNPの性能には劣るものの，従来手法より優れた削減率を提供できたといえる．

しかしながらDCNPには未だ多くの課題が残されている．本研究においてはトラフィック溢れ状態を評価の指針とし，回線容量以上のトラフィックが集中した際と定義しているが，実際にはもっと低いトラフィックの段階で輻輳と呼ばれる状態が起こる．しかしながら輻輳の発生条件は完全に把握されておらず，明確な定義がされていないため，実際のネットワークで運用した際，どの程度輻輳の危険性があるのかが判断できないという問題がある．これは将来に解決しなければならない重要な問題である．

またDCNPはOSPFをベースとしているため，輻輳が発生した際にはコスト情報が更新され，ネットワーク経路が自動で再計算されてしまう．これをシミュレーションに反映することも重要である．さらに本研究は，プロバイダ等のASネットワークの電力削減を対象とする他，データセンター等に発生する過剰なコアルータを削減することも目的としている．サーバは一般的に要求が到着する前にも連続して動いており，サービスを提供していない間も同様の電力を必要とする．そこで，要求に見合った性能のサーバだけを動かしておき，残りは待機状態等にすることで，サーバの消費電力を削減しようという研究が進んでいる．これが実現されれば，サーバの接続されるルータはコアルータとエッジルータの状態を交互に遷移することになり，必ずしも経路に含まれなくてもよい，無駄なルータが出てくることもある．このとき，サーバの電源切り替えと合わせてDCNPの処理を行うことで，ネットワークの最適化を行うことができ，不必要なルータを停止させ，必要なルータを再起動させることができる．ルータの待機モードはネットワーク上で操作が可能な機種も少なからず存在するため，これを利用すれば動的切り替えは可能である．しかしながら，一旦停止した機器が，確実に再起動してくれるという保証はなく．そうなった場合は指定コアルータに障害が発生したものとして，OSPFのリストから削除したのち，ネットワークのシミュレーションから切り離す等の設定をする必要がある．また実際にネットワークを管理する場合には，全体のネットワーク構造を常に把握し続けるシミュレーションサーバを別途用意する必要がある．この結果として，シミュレーションサーバが故障した際に全てのシステムが停止する可能性が生まれてくるが，この問題については，デュプレックスシステムを用いて解決する，あるいはルータの中にP2P等でも利用される親ノードというものを定義して，複数台でネットワークの形状からシミュレーションまでを管理し合うなどの手法で対応できると考えている．

またDCNPを用いるためには，ネットワークに存在する全ての機器の電力情報を把握し，なおかつ正規化しておかなければならない．電力情報を収集するためには各ルータのMIB変数を参照すれば使用電力が得られるはずである．より正確な電力が知りたいのであれば，事前にルータの消費電力情報を登録しておくことで，SNMP等でのデータ取得が可能になる．データ取得時のSNMPによるトラフィック上昇はあまり考えなくてもよく，電力情報はルータが追加された時にのみ測定すれば，その後は離脱するまで更新なしで使い続けることができる．むしろシミュレーション結果として得られたOSPFコストの設定命令を出す際のトラフィックを考慮しなければならず，出来るだけ負荷を与えないため，ネットワーク再構成までの時間間隔を長くする等の処理で対処しなければならない．

DCNPはトラフィック流量を一切考慮せず，ECO-RPを用いてもトラフィック溢れの防止は確実にできるとは言えないほか，一定区間ごとの更新になるためその間どれだけトラフィックが変動しても補正がかからないという特徴がある．またネットワーク機器の処理性能の違いや，経路上重要なルータかそうでないかを区別することができないため，性能の低い機器にトラフィックが集中するほか，経路から外してはいけない機器を外してしまうということもおこりえる．この問題を解決し，最適性を高めるためにはトラフィックの予測や機器の処理性能，機器の重要度に応じた計算を行う必要があり，サーバの動的電源切り替えと連携するためには，サービス要求についても考慮しなければならない．これも重要な課題である．

参考文献

[1] 経済産業省，グリーンITイニシアティブ，http://www.meti.go.jp/press/20071207005/03_G_IT_ini.pdf，Dec 2007(URL available in Jan 2011).

[2] 経済産業省，経済産業省の情報政策について，http://home.jeita.or.jp/is/committee/infoterm/pdf/080604festival_meti.pdf，June 2008(URL available in Jan 2011).

[3] Wassal.A.G ,Hasan.M.A ,"Low-power system-level design of VLSI packet switching fabrics,"IEEE Transaction, Computer-Aided Design of Integrated Circuits and Systems, Vol.20, No.6, pp.723-738, June 2001.

[4] Gunaratne.C, Christensen.K, Nordman.B, Suen.S, "Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR),"IEEE Transaction, Computers, Vol.57, No.4, pp.448-461, April 2008.

[5] Maestro.J.A, Reviriego.P,"Energy Efficiency in Industrial Ethernet: The Case of Powerlink," IEEE Transaction, Industrial electronics, Vol.57, No.8, pp.2896-2903, Aug 2010.

[6] IEEE P802.3az Energy Efficient Ethernet Task Force, http://www.ieee802.org/3/az/, (URL available in Jan 2011).

[7] Christensen.K, Reviriego.P, Nordman.B, Bennett.M, Mostowfi.M, Maestro.J.A, "IEEE 802.3az: the road to energy efficient ethernet," IEEE Journal, Communications Magazine, Vol.48, No.11, pp.50-56, Nov 2010.

[8] Reviriego.P, Maestro.J.A, Herna.ndez.J.A, Larrabeiti.D, "Burst Transmission for Energy-Efficient Ethernet," IEEE Journal, Internet Computing, Vol.14, No.4, pp.50-57, July-Aug 2010.

[9] Chabarek.J, Sommers.J, Barford.P, Estan.C, Tsiang.D, Wright.S, "Power Awareness in Network Design and Routing," IEEE INFOCOM 2008, The 27th Conference on Computer Communications, pp.457-465, April 2008.

[10] Gangxiang.Shen, Tucker.R.S, "Energy-Minimized Design for IP Over WDM Networks," IEEE/OSA Journal, Optical Communications and Networking, Vol.1, No.1, pp.17-29, June 2009.

[11] 鯉渕道紘，大塚智宏，松谷宏紀，天野英晴，"マルチパスイーサネットにおける省電力On/Offリンクアクティベーション法，"情報通信学会　研究報告，計算機アーキテクチャ・ハイパフォーマンスコンピューティング研究会，Vol.14, No.21, pp.121-126, Feb 2009.

[12] 荒井大輔，堀賢治，吉原貴仁，"ネットワークの電力消費量を削減する省電力分散ルーティングプロトコル，"電子情報通信学会　信学技法，情報ネットワーク，Vol.109, No.411, pp.17-22, Feb 2010.

[13] Ling.Li, Somani.A.K, ”Dynamic wavelength routing using congestion and neighborhood information,” IEEE/ACM Transactions, Networking, Vol.7, No.5, pp.779-786, Oct 1999.

[14] John.T.Moy, トップスタジオ訳，小原泰弘監修，”詳解 OSPF，”翔泳社，2003.