

中心性指標を用いたネットワークの冗長化推奨箇所の 優先度付けシステムの提案

大西 玲弥† 井口 信和†‡
Reiya Onishi Nobukazu Iguchi

1. 序論

ネットワークは、企業や社会インフラを支える不可欠な基盤となっており、ネットワークを継続させる可用性は事業継続性の確保という観点から重要である。ネットワーク機器や回線の障害が発生した場合、サービス停止に直結し、収益の損失だけでなく、ブランドイメージの低下など多方面に深刻な影響が及ぶ。PagerDuty が実施した調査によると、1時間のシステムダウンタイムによる損失は約 4400 万円に達するという報告もある[1]。加えて、医療業界などネットワークの中断が許容できない分野もあり、ネットワーク障害によるリスクを低減し、高可用性を確保することが重要である。

このような背景から、可用性を高めるための一般的な手段として、ネットワーク機器や経路の冗長化が挙げられる。しかし、冗長化にはコストが伴うため、全てのネットワーク機器や経路を冗長化することは現実的ではなく、限られたリソースの中で、より優先度の高い箇所を冗長化する必要がある[2]。また、冗長化の判断には専門知識に加えて、対象となるネットワークへの深い理解が不可欠であり、すべての技術者が的確に判断できるとは限らない。

そこで本研究では、ネットワーク運用における冗長化の実施支援を目的に、グラフ理論で用いられる中心性という指標を用いて、冗長化推奨箇所の優先度を数値化し、提示可能なシステムを提案する。ここでの中心性とは、ネットワークの構造的な重要性を測る指標のことである。本システムでは、Web アプリケーション上でネットワークトポロジを登録し、無向グラフ化する。その後、無向グラフ化したトポロジに対して、2 種類の中心性を計算することで冗長化の優先度を数値化する。本システムでは、最短経路における中継回数に基づく媒介中心性と影響度の高いノードとのつながりに基づく固有ベクトル中心性を使用する[3]。

本システムを用いることで、利用者は客観的な指標を基に冗長化の要否を判断できる。さらに、ネットワークトポロジの全てのネットワーク機器や経路に対して計算するため、手作業で判断する場合と比較して、冗長化の優先度が高い箇所の見落としを防ぐことが期待できる。

2. 関連研究

関連研究として、George らの研究[4]では、複数の中心性指標が ISP ネットワークの構造的脆弱性の特定に適用できるかどうかを調査している。この研究では、媒介中心性や次数中心性といった 7 つの中心性指標で順位付けされた

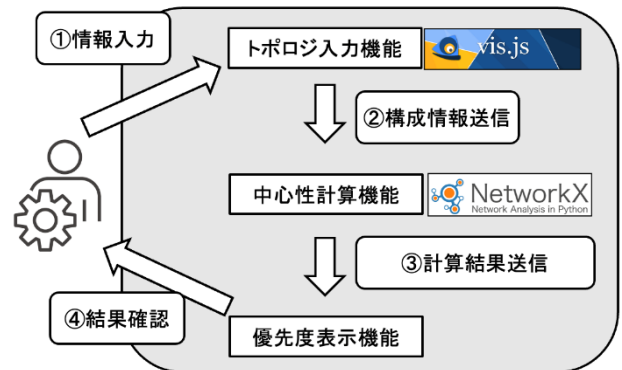


図1 システム構成図

重要ノードを意図的に削除し、その影響を評価するシミュレーションを実施している。中心性が高いと判断された ISP ルーターを削除した場合の影響を接続性、トラフィック処理容量から評価しており、一部の中心性指標を除いて、中心性の高い機器とネットワークへの影響度には強い相関があることが示された。これに対して本研究では、先行研究でネットワークへの影響度と強い相関があると示された中心性指標の中から、構造的なボトルネックを表す媒介中心性とネットワーク内で影響度の高いノードを表す固有ベクトル中心性を用いる。この 2 種類の中心性指標を用いることで、影響度の高いネットワーク機器を数値化し、冗長化の判断を支援する。

その他の中心性に関する関連研究として、Zelin らの研究[5]がある。この研究では、60 個の中心性指標を用いてグラフの巨大連結成分の大きさの変化に基づく、ネットワークの堅牢性を評価している。その結果、本研究で用いる媒介中心性が、ネットワークを分断させる上で最も影響力の強いノードを特定する場合に、有効であることが示された。また、Assemgul らの研究[6]でも同様の結果が示されている。この研究では、10 の人口密集都市の道路網を対象に、媒介中心性を含む複数の指標に基づいた攻撃を行い、ネットワークの堅牢性を評価している。その結果、媒介中心性に基づく攻撃が、ネットワークの分断や効率低下を引き起こす上で、最も効果的であることが示されている。本研究では、冗長化としてネットワークの分断を防止することを重要とし、媒介中心性を用いて優先度付けを実施する。

その他の関連研究として、田中らのネットワーク冗長性可視化システムの研究[7]がある。この研究ではネットワーク構成図上で冗長化地点を可視化し、単一障害点を自動で発見するシステムを提案している。手法として、ネットワーク内の全機器、経路に対して、障害をシミュレートし、通信が断絶する地点を非冗長化箇所として提示している。田中らの研究では、単一障害点となる非冗長化箇所を発見することを目的としている。これに対して本研究では、全

†近畿大学情報学部情報学科, Department of Informatics,
Faculty of Informatics, Kindai University

‡近畿大学情報学研究所, Cyber Informatics Research
Institute, Kindai University

での非冗長化箇所が、等しく重要ではないとして、中心性指標を用いて全てのノードと経路の冗長化優先度を数値化する。これにより、限られたリソース内で非冗長化箇所に対して、どこから冗長化を実施すれば良いのかという判断を支援する。

3. 研究内容

本システムの構成を図1に示す。本システムはWebアプリケーションとして動作し、ネットワークトポロジを登録するトポロジ入力機能、トポロジを無向グラフ化し、グラフ内の全てのノードに対して中心性を計算する中心性計算機能、計算した中心性のランキング結果から冗長化推奨箇所の優先度を可視化する優先度表示機能を持つ。

本システムは、JavaScript ライブラリの React をフロントエンドの実装に使用し、Python の Web フレームワークの Flask をバックエンドの実装に使用する。優先度の算出には、最短経路上の中継度を表す媒介中心性を主に使用し、同程度の媒介中心性を示すネットワーク機器に関しては、ネットワーク全体への重要度を表す固有ベクトル中心性を用いて算出する。また、実際のネットワークを再現した環境で中心性計算を実施することを目的に、各経路に通信量やルーティングプロトコルである OSPF で、ルーティングの決定に使用される OSPF コストや帯域幅などのパラメータを、重みとして登録できるようにする。このパラメータを基に経路に重み付けすることで、実際のネットワーク環境を考慮した中心性計算を実施する。

3. 1. 中心性計算の手法

本研究で用いる中心性計算の手法は、経路の重みを考慮しない重みなし中心性計算と、OSPF コストまたは通信量を考慮する重み付け中心性計算の2種類の計算手法がある。

3. 1. 1. 重みなし中心性計算

重みなし中心性計算での媒介中心性は、ある地点からある地点までの最短経路として、ノード数が最短となる経路を基に計算する。

重みなしの固有ベクトル中心性の計算では、全ての経路の重要性を等しいものとしてどれだけ重要なノードとつながっているかを基に計算する。

3. 1. 2. 重み付け中心性計算

重み付け媒介中心性計算では、各経路に設定されたパラメータの重みの合計値が最小となる経路を最短経路として決定し、計算する。ルーティングプロトコルである OSPF では、OSPF コストの合計値が少なくなる経路をルーティングする仕様であり、この重み付け媒介中心性の計算手法と同様の原理に基づいている。よって、OSPF プロトコルが動作しているネットワークにおいて、OSPF コストを重みとして計算することで、実際のネットワーク環境を考慮した媒介中心性の計算が可能となる。また、重み付けのパラメータとして通信量を逆数で設定する。これにより、多くの通信を担う経路やノードを重みの合計値が少ない最短経路とみなし、その経路上のノードを高く評価する計算が可能となる。

重み付け固有ベクトル中心性計算では、隣接するノードの重要度に、経路に設定したパラメータを重みとして掛け合わせ、計算する。固有ベクトル中心性は重みが大きいほど重要と判断するため、パラメータとして OSPF コストを逆数で重み付けすることで OSPF コストが小さな経路やそ

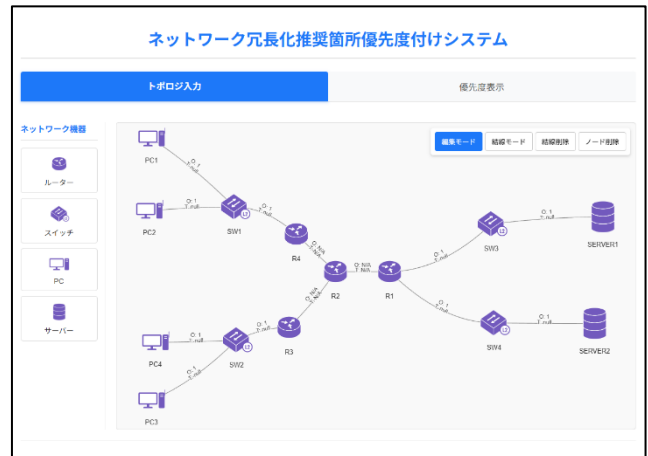


図2 トポロジ入力機能のグラフ操作部

図3 パラメータ設定部

図4 トポロジ入力機能の各種設定部

の経路と接続されているノードを重要とみなして計算する。また、通信量に関しては、そのままの値を重みとして設定することで通信が多いノードや経路を重要とみなして計算することが可能となる。

3. 2. トポロジ入力機能

本機能のGUIを図2に示す。本機能では、利用者が対象ネットワークの構成をWebブラウザ上で登録できるGUIを提供する。本機能はJavaScriptライブラリのReactで構成されており、グラフ操作部分はvis.jsを用いて実装している。

本機能により、ルーターやスイッチなどのネットワーク機器をドラッグ＆ドロップで配置し、マウス操作で結線することで視覚的にトポロジを作成できる。また、経路に対

して通信量や OSPF コストといったパラメータを設定できる。

本機能は、配置したノードを操作する「編集モード」、2 地点のノードを選択して結線する「結線モード」、結線を削除する「結線削除モード」、ノードを削除する「ノード

削除モード」の 4 種類の操作モードを備えており、画面右上のボタンから変更することが可能である。

初期状態では、「編集モード」に設定されている。このモードでは、各ノードをドラッグすることでグラフ操作部に配置したトポロジの位置を自由に変更することができる。

「結線モード」では、経路を作成したいノードを選択し、

経路を作成することができる。結線時には図 3 のように、

パラメータ登録画面が表示され、OSPF コストと通信量を

設定することができる。パラメータ入力画面は経路をダブル

クリックすることで、再度表示させることができ、パラメータ

の変更や追加を実行することができる。パラメータ

は未設定でも中心性計算を実施できるが、重みなし中心性

計算の実施のみに限られる。また、パラメータを使用した

中心性計算実施時に一部経路でパラメータが設定されてい

ない場合には、パラメータの設定されていない経路数ととも

に画面上部に警告が表示され、中心性計算が中止される。

OSPF コストの設定時には、OSPF コストを直接入力するか、

帯域幅を入力することができる。OSPF コストは基準となる

参照帯域幅を帯域幅で割ることで算出される値であり、

帯域幅を入力した場合には、OSPF コストに自動的に変換

され、経路に設定される。また、参照帯域幅は図 4 の画面

下部の設定欄から変更することができる。

「結線削除モード」では、GUI 上で任意の経路をクリック

することで、指定した経路をトポロジ上から削除すること

ができる。

「ノード削除モード」では、GUI 上で任意のノードをクリ

ックすることで、指定したノードとそのノードが接続し

ている経路をトポロジ上から削除することができる。

また、システム上で作成したトポロジデータのダウンロード、

システム上へのトポロジデータのアップロードを

JSON 形式のファイルを通して可能にした。JSON 形式のト

ポロジデータを本システムにアップロードすることで、経

路に設定したパラメータやノードが配置されていた地点の

情報から、保存したトポロジが即座に GUI 上に自動展開さ

れる。この機能により利用者は、本システムで作成したト

ポロジを再利用することが可能となる。

3.3. 中心性計算機能

本機能では、入力されたトポロジを無向グラフ化し、中心

性指標を計算する。本機能は、バックエンドの Flask サー

バ上で動作しており、中心性計算には Python ライブラリ

の NetworkX を用いて実装した。計算する中心性指標は、

媒介中心性と固有ベクトル中心性の 2 種類である。媒介中

心性の計算には NetworkX ライブラリの

betweenness_centrality 関数を、固有ベクトル中心性の計算

には同ライブラリの eigenvector_centrality 関数をそれぞれ用

いる¹⁾。

また、実際のネットワーク環境を考慮した計算ができる

ように、通信量や OSPF コストなどの重み付けを考慮した



図 5 優先度表示機能の GUI

表 1 性能評価実験に使用したサーバ

CPU	Memory	OS
Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz	32GB	Windows11

表 2 リング型トポロジの性能評価実験結果

ノード数	100 ノード	1000 ノード
平均計算時間	0.049 秒	3.65 秒

表 3 ツリー型トポロジの性能評価実験結果

ノード数	100 ノード	1000 ノード
平均計算時間	0.043 秒	3.42 秒

表 4 フルメッシュ型トポロジの性能評価実験結果

ノード数	100 ノード	1000 ノード
平均計算時間	0.2 秒	193.80 秒

中心性計算機能も実装している。重みなし中心性計算と各パラメータを使用した重み付け中心性計算は、トポロジ入力機能の各種設定部で変更することができる。

3.4. 優先度表示機能

本機能の GUI を図 5 に示す。本機能では、計算された中心性指標に基づく冗長化の優先度を GUI 上に表示する。算出された優先度に基づき、GUI 上のネットワーク機器や経路に対して、色や大きさを変更することで、結果を可視化する機能を実装した。優先度は 0 から 1 の間で正規化され、この値に基づき、優先度の値が高いほど暖色系の色で強調表示する。優先度が 0.8 以上の経路に関しては赤色、0.8 未満 0.6 以上の経路はオレンジ色、0.6 未満 0.4 以上の経路には黄色で表示される。また、算出された優先度のランキングと各ノード、各経路の媒介中心性と固有ベクトル中心性のランキングをログとしてトポロジ下部の GUI 上に表示し、各中心性の計算結果も表示する。

4. 実験

性能評価実験として、トポロジの種類やノード数を変化させた、複数のランダムなトポロジデータを用いて計算時間を計測した。性能評価実験は表 1 のサーバを用いて実施し、その結果を表 2 から表 4 に示す。

1) ¹ NetworkX: Centrality
<https://networkx.org/documentation/stable/reference/algorithms/centrality.html>(参照日 2025-06-20)

本実験では、全ノードをリング状に接続し、さらにノード間にランダムな数の経路を追加したリング型トポロジと、ランダムな接続先と追加経路を選択して階層化したツリー型トポロジ、全てのノードを互いに接続したフルメッシュ型トポロジの3種類をテストデータとして用いた。トポロジはそれぞれ10個ずつ作成し、実験を実施した。

性能評価実験の結果、100ノード規模のトポロジでは、リング型、ツリー型ともに0.1秒以下、最も経路数が多いフルメッシュ型でも約0.2秒と、いずれも高速に計算された。この結果により、100ノード規模であれば、トポロジ構造に依らず、本システムは問題なく動作することを確認した。

また、1000ノード規模のトポロジでは、リング型とツリー型はそれぞれ4秒以下で計算が完了した。フルメッシュ型では、平均して約194秒の計算時間となり、大幅に増大したものの、最も計算量の多いフルメッシュ型で5分以内と現実的な時間内で計算を完了できたことから、本システムは1000ノード規模のトポロジにも対応可能であることを確認した。

5. まとめ

本研究では、ネットワーク運用における冗長化の実施支援を目的に、中心性指標を用いた冗長化推奨箇所の優先度付けシステムを提案した。また、性能評価実験として、構造の異なる3種類のトポロジに対して、100ノード、1000ノードでの計算時間を計測し、1000ノード規模のトポロジに対しても、現実的な時間で計算可能であることを確認した。

今後は中心性計算機能の改良、優先度表示機能の改良とともに、本システムが冗長化の優先度を正しく示しているかを評価することを目的に、評価実験を実施する。評価実験として、優先度の高い地点のノードを削除した場合と、その他の地点のノードを削除した場合で到達不能なノードペアの総数の変化や、平均ホップ増加量、仮想的に設定した総損失トラフィック量の変化などの影響度を比較することを計画している。また、ランダムに作成したデータに加え、意図的に単一障害点と冗長化済みの地点を作成したトポロジで評価実験を実施する予定である。

また、媒介中心性が冗長な経路を高く評価する可能性が指摘されている[8]。そのため、実験の結果によって、さらに異なる中心性指標の使用の検討や優先度付けの手法の改良も実施する予定である。

6. 参考文献

- 1) PagerDuty: インシデントの平均修復時間 6 時間 12 分、システムダウンタイムのコストは 1 時間 4,440 万円。自動化で対応エンジニアの負担軽減へ、
<https://www.pagerduty.co.jp/blog/report_risk_incident>
(参照日 2025-04-22)
- 2) 松川達哉、松浦洋、中川雅弘、越地弘順、林理恵：ネットワークの強じん性を高める高信頼化設計技術、電子情報通信学会誌、Vol.108 No.5、p456-p463(2025)
- 3) Saxena Akarti, Sudarshan Iyengar, "Centrality Measures in Complex Networks: A survey." arXiv preprint arXiv:2011.07190 (2020).
- 4) George Nomikos, Panagiotis Pantazopoulos, Merkourios Karaliopoulos, Ioannis Stavrakakis "The multiple instances of node centrality and their implications on the vulnerability of ISP networks." arXiv preprint arXiv:1312.4707 (2013)
- 5) Zelin Wan, Yash Mahajan, Beom Woo Kang, Terrence J. Moore, Jin-hee Cho "A Survey on Centrality Metrics and Their Network Resilience Analysis" in IEEE Access, vol. 9, pp. 104773-104819, (2021)
- 6) Assemgul Kozhabek, Wei Koong Chai "Robustness assessment of urban road networks in densely populated cities." Appl Netw Sci 10, 29 (2025)
- 7) 田中啓基、井口信和: ネットワーク冗長性可視化システムにおける非冗長構成箇所自動発見機能の実装、第 84 回全国大会講演論文集、p. 453-454(2022)
- 8) Bitao Dai, Shuo Qin, Suoyi Tan, Chuchu Liu, Jianhong Mou, Hongzhong Deng, Fredrik Liljeros, Xin Lu, "Identifying influential nodes by leveraging redundant ties", Journal of Computational Science, vol.69, 2023