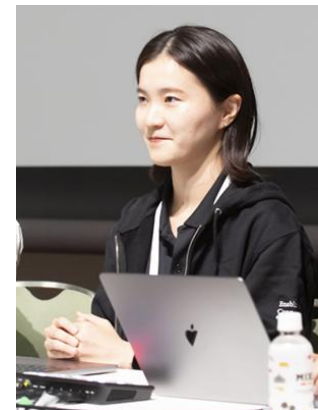


ネットワーク分析で「読むべきRFC」を探す

BBSakura Networks 株式会社 / BBIX株式会社
外山 結月

自己紹介

- 外山 結月（とやま ゆづき）
 - 所属：BBSakura Networks株式会社 / BBIX株式会社
- 経歴
 - 2024年4月～：BBSakura Networks株式会社 / BBIX株式会社
 - ソフトウェア・インフラエンジニア
 - 社内の契約管理システムのバックエンド開発（Go）
 - オンプレでのサーバー運用
 - RFC初心者
- JANOG参加歴
 - JANOG 56「トラフィックグラフはどう作られる？ ～NetFlowを使ってみる～」
 - JANOG 57
 - JANOG 58 ←今回



システム開発でこんな経験、ありませんか？



BBSakura
Networks

「MTAサーバたってるから、SMTPでそこに投げれば外部にメール送れるよ」

→SMTPとは？？MTAとは？？

「SNMPでMIB指定してVLANの情報取得して差分検出したい」

→SNMPとは？？MIBとは？？

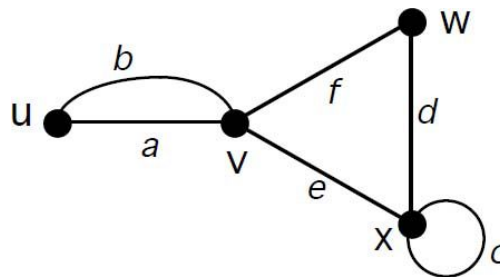
Google検索・LLMに質問・本の一節を読み、解決

が、本当はRFCとか気になるよね...

RFC気になるが...

- なにから読めば良いのかわからない
- どう読めば良いのかわからない

→ グラフ理論的なアプローチで、解決したい！



グラフ [1]

It has vertex-set $V = \{u, v, w, x\}$ and edge-set $E = \{a, b, c, d, e, f\}$.

The set $\{a, b\}$ is a multi-edge with endpoints u and v , and edge c is a self-loop.

解析の流れ



1. グラフの定義
2. PageRank
3. Recent In-Degree
4. Leiden algorithm

グラフの定義

対象データ：rfc-indexより過去のRFCを取得 [2]

各RFCのReferences節から引用を抽出

→引用関係のある 9743ノード / 105175エッジ を取得

例) RFC9110 (HTTP Semantics)
9110→9111,
9110→1950,
9110→1951, … の有向辺を定義

19. References

19.1. Normative References

[CACHING] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Caching", STD 98, RFC 9111, DOI 10.17487/RFC9111, June 2022, <<https://www.rfc-editor.org/info/rfc9111>>.

[RFC1950] Deutsch, P. and J-L. Gailly, "ZLIB Compressed Data Format Specification version 3.3", RFC 1950, DOI 10.17487/RFC1950, May 1996, <<https://www.rfc-editor.org/info/rfc1950>>.

[RFC1951] Deutsch, P., "DEFLATE Compressed Data Format Specification version 1.3", RFC 1951, DOI 10.17487/RFC1951, May 1996, <<https://www.rfc-editor.org/info/rfc1951>>.

RFC9110 References [3]

[2] RFC Editor. RFC Index | RFC Editor. [viewed 2026-07-13]. Available from: <https://www.rfc-editor.org/rfc-index/#top>

[3] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", RFC 9110, DOI 10.17487/RFC9110, June 2022, <<https://www.rfc-editor.org/info/rfc9110/>>.

PageRank

- 元はWebページのランキングアルゴリズム
- 被引用数 + 引用元の重要度 を合わせて各RFCの重要度を判定
 - 多く引用されているRFCほどスコアが高い
 - スコアが高いRFCから引用されているほどスコアが高い

ツール：

- NetworkX.pagerank ($\alpha=0.85$) [4]

[4] NetworkX Developers. pagerank — NetworkX 3.6.1 documentation. [viewed 2026-07-13]. Available from:

https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank.html

PageRank (結果)

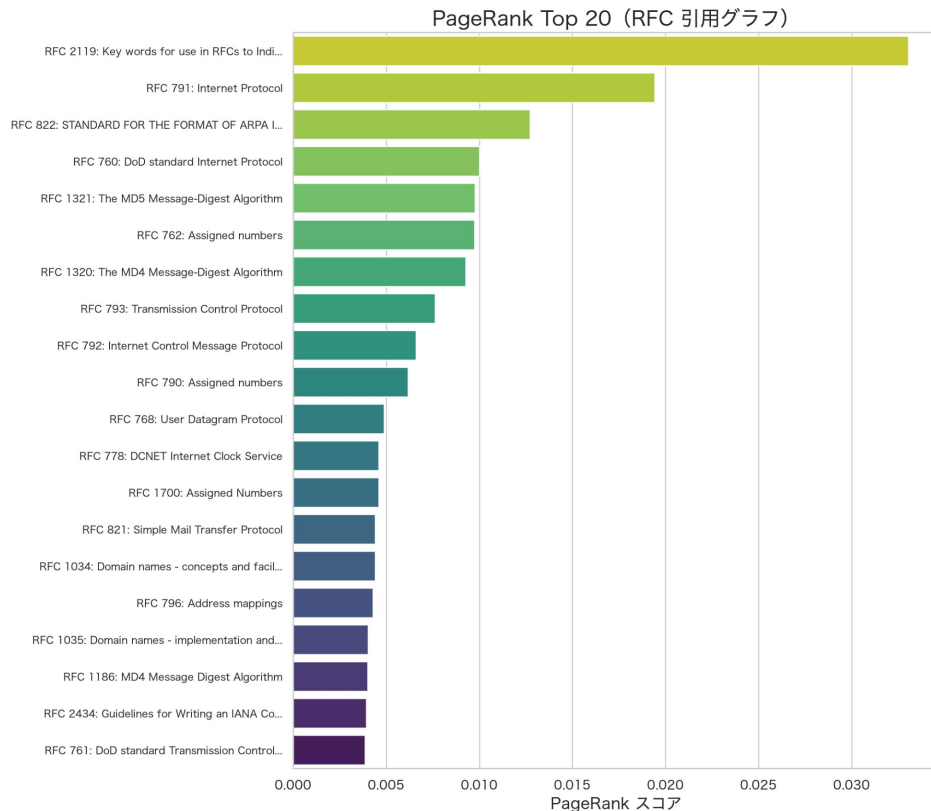
1位: RFC 2119

(MUST, SHOULD等の定義)

2位: RFC 791: Internet protocol

3位: RFC 822: ARPA

→古いRFCが強くなってしまう



Recent In-Degree

- 「直近の入次数」
(直近の期間における、他のRFCから自ノードに向けたエッジの数)
- 直近 = 2022年とした
- 2022年以降、他のRFCに引用された数でランキング

Recent In-Degree (結果)

1位: RFC2119

2位: RFC8174 (Updates RFC2119)

3位: RFC8126 (IANA Considerations)

4位: RFC8446 (TLS1.3)

5位: RFC5280 (X.509 PKI)

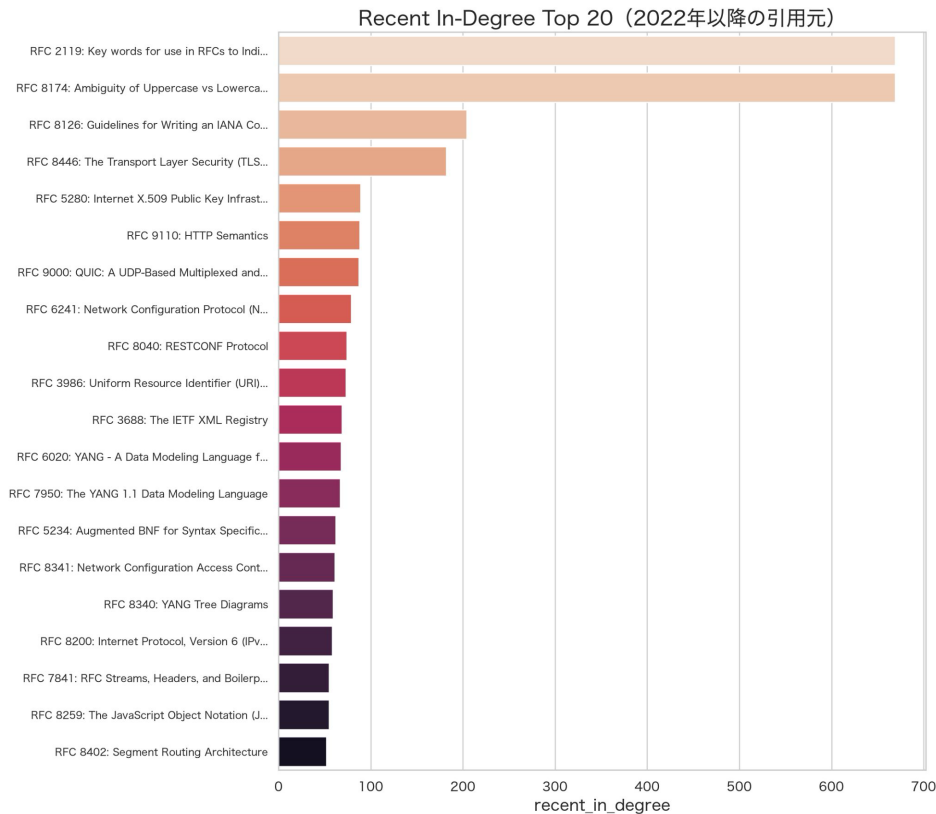
6位: RFC9110 (HTTP Semantics)

7位: RFC9000 (QUIC)

8位: RFC6241 (NETCONF)

9位: RFC8040 (RESTCONF)

10位: RFC3986 (URI)

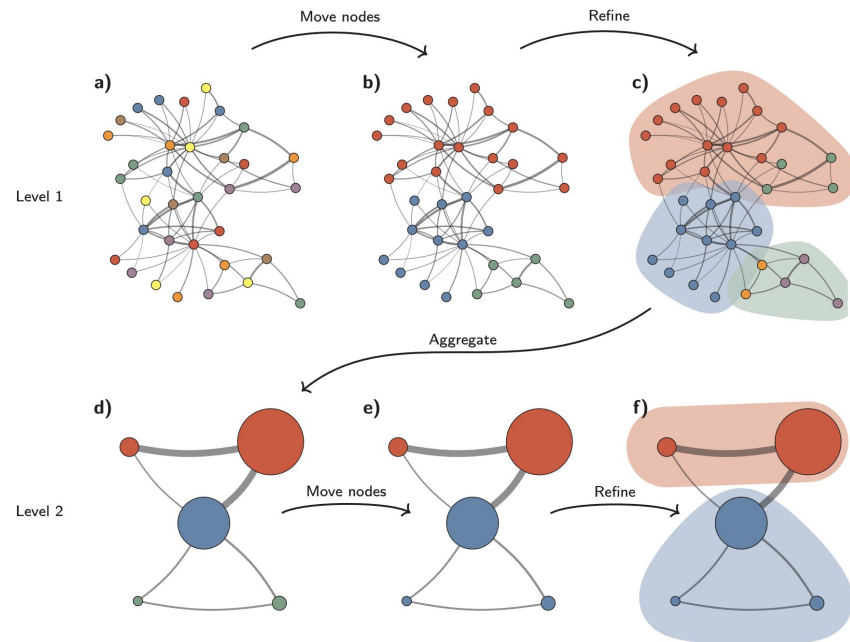


Leiden algorithm (コミュニティ検出) [5]

- 複雑なネットワーク (SNSのフォロー関係など) から、コミュニティを見つけ出すアルゴリズム
- Louvainアルゴリズムの改良版
- <https://github.com/vtraag/leidenalg>

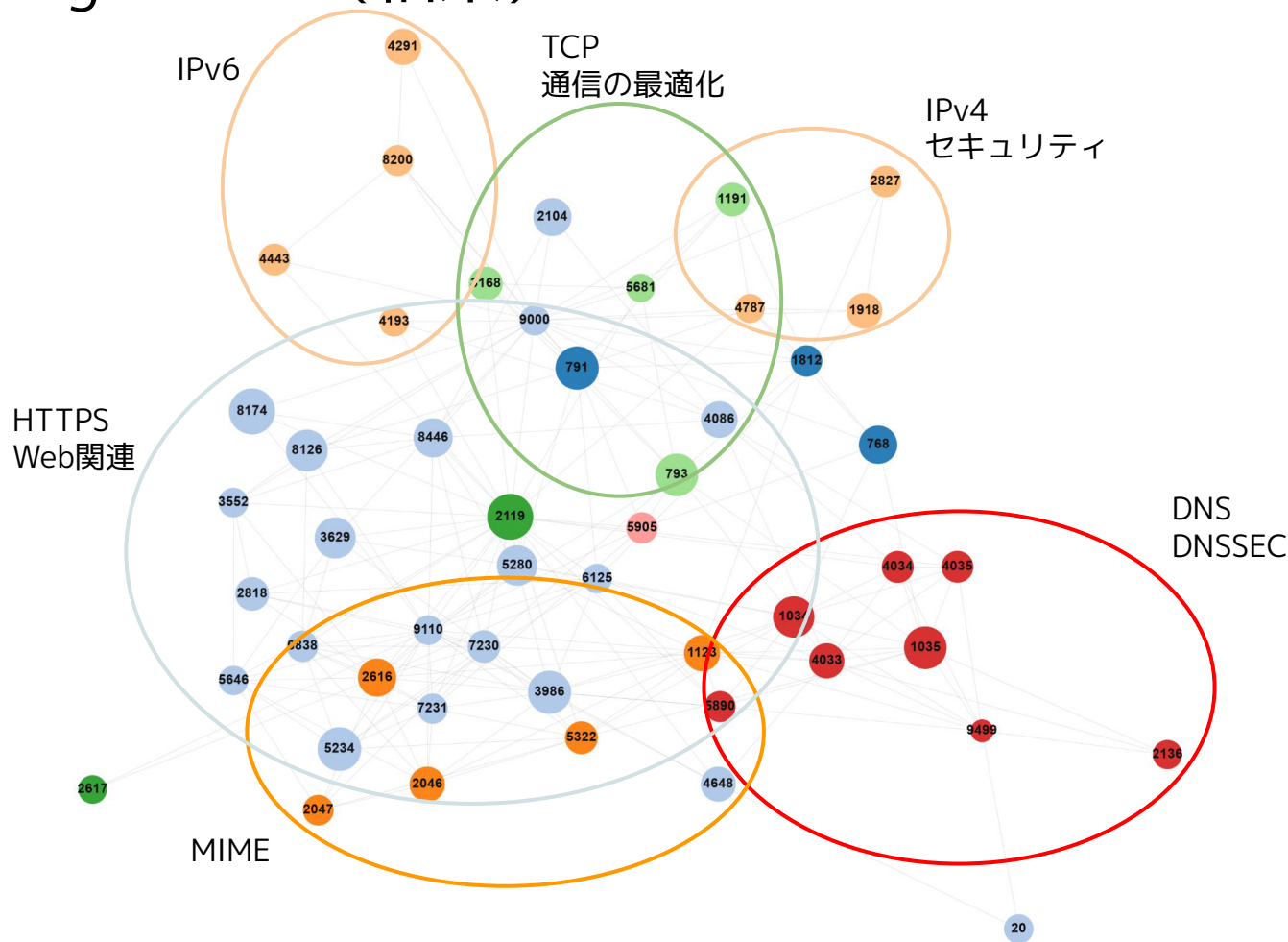
ステップ

1. ノードの移動・分割_(b)・Refine_(c)
2. 集約_(d)
3. 集約ネットワーク内でノードを移動_(e)・Refine_(f)
4. modularity指標の向上ができなくなるまで繰り返し



Leiden algorithm [4]

Leiden algorithm (結果)



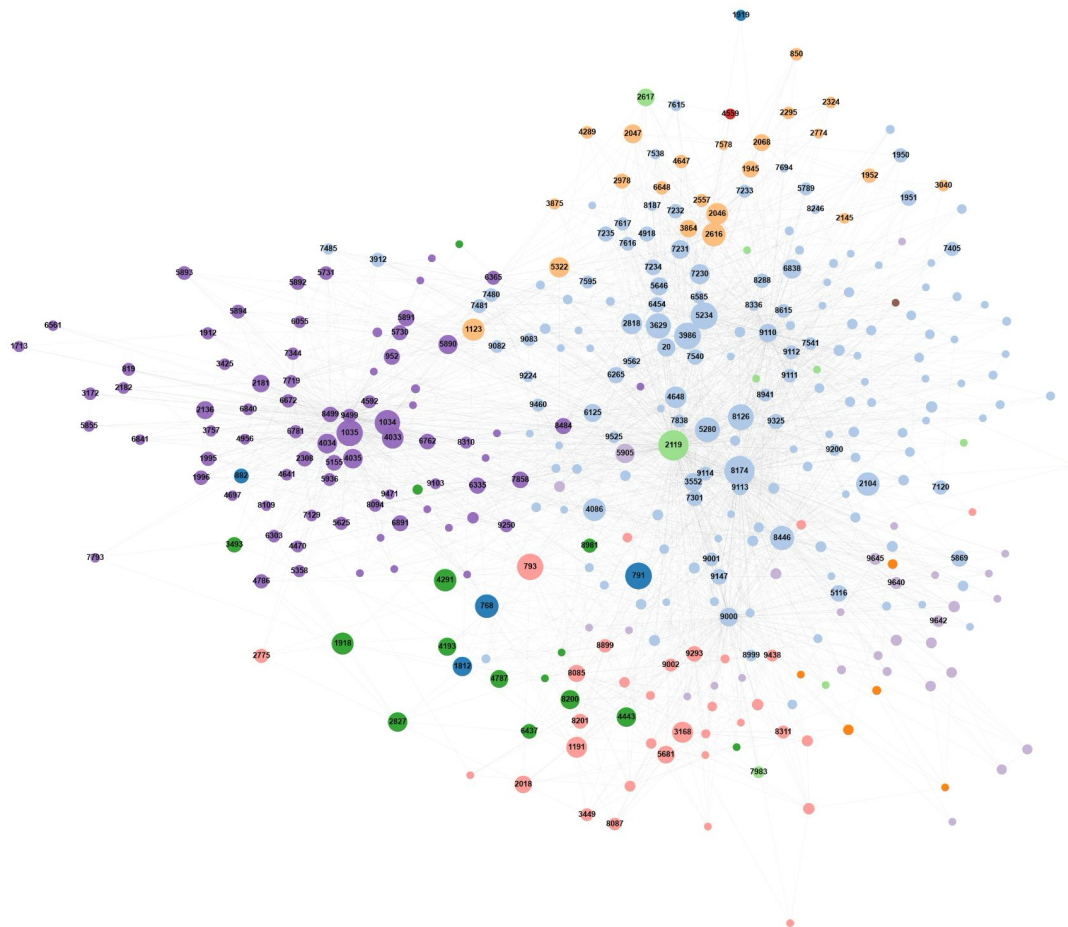
- RFC2119, 8174をまず読んでみる（それぞれ20分くらいあれば読める）
 - MUST, SHOULD, MAYなどの定義がわかる
 - Updateの入り方とか、セキュリティの章などRFC共通の箇所がわかってくる
- Recent In-Degreeの中で気になるTopicに関連して、コミュニティ近傍を攻めていくと良さそう
 - グラフを眺めると楽しい
 - RFCから有限オートマトンの自動抽出の研究^{[6][7]}とかもあるので、中身を理解する方法の検討に役立つかも

[6] Pacheco, Maria Leonor, et al. "Automated attack synthesis by extracting finite state machines from protocol specification documents." *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022.

[7] Feng, Xinguo, et al. "Detecting contradictions from IoT protocol specification documents based on neural generated knowledge graph." *ISA transactions* 141 (2023): 10-19.

Appendix

コミュニティ検出（結果詳細）



Enabling a Connected Future.



BBSakuraNetworks