動画配信サービスのための
軽量分散協調キャッシュ基盤の研究

中島　拓真

電気通信大学 大学院情報システム学研究科

博士（工学）の学位申請論文

2017年12月

動画配信サービスのための
軽量分散協調キャッシュ基盤の研究

博士論文審査委員会

主査　　吉永 努　　教授

委員　　加藤 聰彦　教授

委員　　大坐畠 智　准教授

委員　　田野 俊一　教授

委員　　本多 弘樹　教授

Light-weight Cooperative Caching Scheme for Video-on-Demand Services

Takuma Nakajima

Abstract

Internet traffic has rapid growth due to the popularization of the Video-on-Demand (VoD) services. While the VoD service providers offer users platforms that allow watching video contents without a time restrictions, it is estimated that the video traffic will increase three times in five years, reaching more than 80% of the whole Internet traffic in 2020. Although such enormous traffic could be transferred by introducing new network fabrics such as routers and switches, such solution is expensive and not scalable since the needed network fabrics are proportional with the data growing.

VoD service providers usually deliver contents via Content Delivery Network (CDNs) to offload the traffic. CDN providers construct large-scale cache networks by placing cache servers nearby users in the world. Cache servers copy the contents data when they pass the cache network, and reuse the data for the same incoming requests. Since the video contents rarely change once they are uploaded, cache servers efficiently reduce the internet traffic. Such behavior reduces the traffic by eliminating duplicated data transfers between cache servers and distant content servers. However, it is not realistic to store the whole video library in a cache server since it has an only limited capacity of storage while video contents are continually uploaded. Moreover, there are limited locations that CDN providers place their cache servers. This implicates that CDN providers still cannot reduce traffic on peering links to the contents servers as well as inside of the Internet Service Provider (ISP) networks, introducing congested links as well as data traffic overhead.

Several CDN providers are considering to reduce and distribute the traffic load by increasing effective cache capacity by sharing cached contents among cache servers. Such approach utilizes a traffic engineering technique and makes large cache capacity. However, it is difficult for CDN providers to apply an efficient traffic engineering since they usually have no global knowledge of the underlying networks. Several ISPs are considering to place cache servers in their backbone networks and create a cooperative cache networks. Such solution allows effective use of network links and cache servers since they are managed by the same owner, and it is called Telco-CDNs.

Several studies introduce cache management strategies for the Telco-CDNs to efficiently utilize the cache capacity. A major approach is to store different contents in different cache servers, which reduces traffic by increasing effective cache capacity. However, such strategy usually does not consider request popularities of contents, leading traffic concentration to a few cache servers that store popular contents. Another approach formulates an optimization problem to minimize traffic and tries to find sub-optimal content locations by solving the optimization problem. Although such content locations reduce much traffic, it takes a long calculation time overhead. Since the video traffic pattern changes 20-60% every hour, such long calculation time causes mismatches in content locations, leading a traffic overhead.

This thesis provides two approaches for the traffic reduction and combines them by considering the video traffic patterns. The first approach proposes a hybrid cache algorithm that combines different two cache algorithm in each server in the network, or split a cache storage and apply different two cache algorithm for each area. One cache area is managed by Least Frequently Used (LFU) based algorithm that store contents based on the request popularities. Such management efficiently reduce traffic under the static access patterns. The other cache area is managed by Least Recently Used (LRU) based algorithm that store contents based on the last accessed time. LRU based algorithm efficiently store contents that are newly inserted.

The second approach proposes a color-tag based cooperative caching strategy that efficiently distributes contents among the cache network. It utilizes LFU based storage of the hybrid cache algorithm as a cooperative caching area, which increases the effective cache capacity and reduces the traffic. The color-tags are applied to both contents and cache servers to store popular contents in many cache servers while unpopular contents are also cached in a few cache servers. Such behavior efficiently reduces both ISP-internal and origin server's traffic. The light-weight

tag management algorithms are also proposed to follow the video access patterns' changes. Furthermore, a color-tag based routing algorithm is also proposed to enhance cache utilization by fetching contents from nearby cache servers.

Evaluation results showed that the hybrid cache algorithm could follow rapid changes in access patterns. It maintains the high cache hit rate when several contents are inserted, while a single use of the LRU algorithm achieves low hit rate and the LFU algorithm drops the hit rate. Moreover, the color-tag based caching scheme could achieve a performance close to the sub-optimal one obtained with a genetic algorithm calculation, with only a few seconds of computational overhead. The color-based routing scheme could also reduce the traffic by up to 31.9% when compared with the shortest-path routing.

動画配信サービスのための
軽量分散協調キャッシュ基盤の研究

中島 拓真

概　要

時刻に関係なく動画を視聴できるオンデマンド動画配信 (Video-on-Demand; VoD) サービスの普及に伴い，インターネット通信量は急激に増大している．インターネット通信量は 5 年で 3 倍の速度で増加すると見込まれており，2020 年にはインターネット通信量の 80%以上を動画通信が占めると予測されている．増大する通信量はルータやスイッチなどの通信機材の増設や刷新で対応できるが，通信量の増加に伴い継続的な増設が求められるため，経済的ではない．

VoD サービス事業者は一般に，コンテンツ配信ネットワーク (Content Delivery Network; CDN) 事業者に大容量コンテンツの配信を委託している．CDN 事業者は，世界中のユーザから近い位置にキャッシュサーバを設置し，大規模なキャッシュネットワークを構築している．各キャッシュサーバはコンテンツがネットワークを通過する際にデータをコピーしておき，再利用することで，通信量を削減している．動画コンテンツは一度アップロードされると滅多に更新されることがないため，キャッシュサーバは重複する通信を削減できる．このようにすることで，キャッシュサーバは遠くの配信サーバとの通信回数を削減し，効率よくインターネット通信量を削減する．しかしながら，キャッシュ容量には上限があり，動画コンテンツは継続的に追加されるため， 1 台のキャッシュサーバにすべての動画コンテンツを保持させることは現実的ではない．また，CDN 事業者のキャッシュサーバは設置拠点が限られているため，キャッシュサーバの通信量や，キャッシュサーバまでの経路を提供するインターネットサービスプロバイダ (Internet Service Provider; ISP) 内の通信量を削減できない．その結果，通信路の混雑や通信量の増大を招いてしまう．

CDN 事業者は，複数のキャッシュサーバでキャッシュされたコンテンツを共有することで実効キャッシュ容量を拡大し，通信量削減と負荷分散を図ろうとしている．このような手段はデータ転送路を制御するトラフィックエンジニアリングをもとに実現されるが，CDN 事業者は ISP の物理ネットワーク形状やリンク帯域等に関する知識をもたないため，効率よくキャッシュサーバを協調させることが難しい．そのため，近年は ISP 事業者が自社のネットワーク中にキャッシュサーバを配置し，トラフィックエンジニアリングを駆使して協調動作させることで，分散協調キャッシュネットワークの構築を検討している．このような方法をとることで，ネットワークとキャッシュサーバの両方を同一の事業者が管理するため，効率の良い通信量削減が実現できる．ISP が管理する ISP 内のキャッシュネットワークは Telco-CDN と呼ばれる．

最近の研究では， Telco-CDN のキャッシュサーバを効率よく管理する方法が提案されている．典型的には，複数のキャッシュサーバで保持するコンテンツの規則を設定しておき，各キャッシュサーバで異なるコンテンツを保持させることで，実効キャッシュ容量を拡大し，通信量の削減が実現されている．しかしながら，このような方法は各コンテンツのアクセス頻度情報を考慮しないため，人気上位のコンテンツを保持する数台のキャッシュサーバに負荷が集中してしまう．また，効率の良いコンテンツ配置を求める最適化問題を設定して計算することで，通信量削減効果の高い分散協調キャッシュを実現する研究も行われている．しかしながら，最適化問題の計算には長時間の計算を要する一方で，VoD サービスの動画アクセスパターンは 1 時間で 20-40%程度変化してしまうため，計算が終了した時点で最適なコンテンツ配置との乖離が生まれ，通信量削減効果が低減してしまう．

本論文では，VoD サービスのアクセス傾向を効率よくキャッシュする 2 種類のキャッシュ制御アルゴリズムを提案し，組み合わせて利用することで，通信量の削減を図る．まず第 1 に， 2 種類の異なるキャッシュアルゴリズムを組み合わせたハイブリッドキャッシュアルゴリズムを提案する．このアルゴリズムは，異なるキャッシュアルゴリズムをネットワーク中に混在させたり， 1 台のキャッシュサーバのストレージ領域を分割してアルゴリズムを混合して利用することで，急激に変動する動画アクセスを効率よくキャッシュして，高い通信量削減効果を維持する．アクセス頻度の高いコンテンツを保持する Least Frequently Used (LFU) ベースのアルゴリズムで高い通信量削減効果を実現し，最近アクセスされたコンテンツを優先的に保持する Least Recently Used (LRU) ベースのアルゴリズムで急激なアクセス傾向の変化に追従する．

第 2 に，色タグ情報を用いた分散協調キャッシュ制御手法を提案し，キャッシュネットワーク中のコンテンツ配置を効率よく制御する．この方法は，コンテンツとキャッシュサーバの両方に色タグを設定し，色がマッチする場合にキャッシュするよう制御することで，コンテンツを分散配置し，実効キャッシュ容量を拡大する．具体的には，先に述べたハイブリッドキャッシュの LFU 領域に色タグを設定し，大容量な分散協調領域として利用するとともに，小容量な LRU 領域ではタグ情報にかかわらずコンテンツをキャッシュさせることで，動画アクセス傾向の変化に追従する．アクセス頻度の高いコンテンツほど多数の色を割り当てることでユーザからのホップ数を短縮し，コンテンツ配信サーバだけでなく ISP ネットワーク内部の通信量も効率よく削減する．色タグ情報の軽量管理手法と，色タグ情報を活用する経路制御アルゴルズムも合わせて提案し，軽量な計算オーバヘッドで高い通信量削減効果を実現する．

ハイブリッドキャッシュアルゴリズムの評価では，新規にアクセス頻度の高いコンテンツを追加しても，LFU ベースのキャッシュ領域で高い通信量削減効果を達成しつつ，LRU ベースのキャッシュ領域で通信量を維持できることが示された．また，色タグ情報に基づく分散協調キャッシュアルゴリズムは，遺伝的アルゴリズムで計算した準最適制御に近い通信量削減効果を実現し，その計算オーバヘッドも小さく抑えられることを確認した．また，色タグ情報を活用した経路制御アルゴリズムは，最短経路制御と比較して 31．9%の通信量削減効果を得られることを確認した．

目 次

図目次

表 目 次

第1章　序論

1.1　背景と目的

視聴者が時刻に関係なく好きな動画を視聴できるオンデマンド動画配信（Video-on-Demand; VoD）サービスが普及した結果，インターネットの通信量が急激に増大している．Cisco 社の調査によると，インターネット通信量は 5 年で 3 倍程度の速度で増大しており，2021 年には動画通信がインターネット通信量の 82%に達すると予測されている [1]. 最近では，4K，8K と呼ばれる高精細動画や，360 度好きな方向を視聴できる 360 度動画，リアルタイムに 3D オブジェクトを描画する Virtual Reality (VR) コンテンツが登場しており，インターネット通信量の増大に加担している．増大する通信量はルータやスイッチ等のネットワーク機器に負荷をかけ，混雑リンクの原因となる．混雑した通信路では通信の遅延やパケットロスが発生しやすくなるため，コンテンツの取得が再生に間に合わず途中で動画再生が停止したり，データの一部が不着になって再生画面が乱れるなどの問題の原因となる．また，動画配信以外の通信も混雑リンクの影響を受けて通信が不安定になるリスクがあるため，高速で安定した通信基盤の実現のためには，機器の負荷を低減する工夫が不可欠である．通信量の増大には，ルータやスイッチ，光多重化装置等の通信設備の増設や，高性能なモデルに刷新することで対応できるが，機器 1 台あたりの価格が高額なだけでなく，ネットワークの一部を高速化してもそれ以外の通信路がボトルネックとなり，全体で得られる効果は限定的であるため，通信設備の刷新には膨大な投資が必要である．

一般に，動画配信サービスは，リアルタイムに動画を配信する「ライブ配信」と，時間に関係なく動画を視聴できる「オンデマンド配信」の 2 つに分類できる．ライブ配信サービスは，テレビ番組と同様，不特定多数のユーザが同じ動画を同時刻に視聴するため，IP マルチキャスト技術を利用して複数ユーザへのデータ通信をまとめ，通信負荷を低減できる．一方，オンデマンド動画配信サービスでは，不特定多数のユーザがそれぞれ異なる動画を異なる時刻に視聴する．その結果，IP マルチキャストで通信をまとめられず，同じ動

画を異なる時刻，異なるユーザに何度も配信する必要があるため，オンデマンド配信はライブ配信と比べて通信量が増大しやすく，インターネット通信量の主たる増大要因である．実際，インターネットで配信される動画通信量の内訳は，オンデマンド動画が 83-94%程度を占める [1].

VoD サービスの通信量削減には，一度配信したデータを通信路上のサーバでコピーし再利用するキャッシュサーバが利用される．通常，動画は一度アップロードされると，その後内容が変更されることはほとんどないため，視聴者の近くにキャッシュサーバを配置して，過去に要求されたデータのコピー（キャッシュ）を保存して再利用することで，キャッシュサーバが配信サーバからデータを取得する回数を減らし，通信量を削減できる．現在は，コンテンツ配信ネットワーク（Content Delivery Network; CDN）と呼ばれる大規模キャッシュネットワークを利用して動画データが配信されている [2]．CDN は，各国に数拠点あるキャッシュサーバをネットワークで接続して構成されており，ユーザから配信サーバまでの距離がが遠くても，ユーザのリクエストは DNS によって近くに設置されたキャッシュサーバにリダイレクトされるため [3]，コンテンツ取得に要するホップ数が削減され，通信量の削減と取得遅延の短縮に利点がある．

しかし，キャッシュサーバのストレージ容量には制約があるため，継続的に動画が追加されるライブラリ全体を保存することは現実的ではない．動画ライブラリには，映像制作会社が制作した映画やドラマ，アニメ作品だけでなく，一般のユーザやグループが作成したオリジナル動画も継続的に追加されるため，キャッシュ領域は時間の経過とともに相対的に減少してしまう．また，最近は 4K や 8K と呼ばれる高画質動画も普及し始めており，動画 1 つあたりの容量も増大している [1]．動画の高画質化に伴いファイルサイズが大きくなると，1 秒間あたりに転送しなければならないデータ量（通信路に要求されるスループット）も比例して増加するため，動画データを転送する通信機器に高い負荷をかける．その結果，キャッシュサーバが多数配置されていないと，少数のキャッシュサーバに通信負荷が集中してしまうため，再生までに時間がかかったり，再生中の動画が停止するなど，円滑な動画配信の妨げとなる．CDN 事業者は，ISP ネットワーク内にキャッシュサーバを多数配置し，協調動作するようサーバ間のトラフィックエンジニアリングを駆使して通信負荷を分散することも理論上は可能であるが，CDN 事業者は ISP の物理ネットワーク構造を把握していないため，適切なトラフィックエンジニアリングは容易ではない [4]．CDN 事業者による不用意なトラフィックエンジニアリングは，ISP が実施するトラフィックエン

ジニアリングと競合し，特定の通信リンクに負荷をかけたり，上流 ISP と接続される従量課金リンクに大容量データを流してコスト増大の要因となる．このような懸念から，ISP は CDN 事業者がトラフィックエンジニアリングを実施することを許容できず，CDN 事業者が効率よく通信量を削減することは現実的ではなくなってきている．

以上の問題を踏まえて，最近では ISP 事業者が ISP 内部にキャッシュサーバを多数配置して利用する，ISP 内キャッシュネットワークの研究が実施されている [5, 6, 7, 8, 9, 10]．ISP 内キャッシュネットワークは，物理ネットワークとキャッシュサーバの両方を同一 ISP 事業者が管理する．そのため，物理ネットワーク構造を把握した効率の良いトラフィックエンジニアリングが可能である．また，ISP ネットワークは CDN よりもユーザに近い位置に展開されているため，データとユーザ間の距離を短縮した効率の良い通信基盤の実現が可能である．さらに，キャッシュサーバの通信を制御して別のキャッシュサーバに転送することで，複数のキャッシュサーバを協調動作させて実効キャッシュ容量を拡大し，CDN や上流ネットワークとの接続リンクの負荷を低減する分散協調キャッシュの研究も実施されている [11, 12, 7, 10]．これらの接続リンクは従量課金されることが多いため，ISP にとって金銭的コストの削減も期待できる．分散協調キャッシュでは，動画データをどのサーバでキャッシュするかによって，ユーザから動画データまでの平均ホップ数が決定される．通信量はこの平均ホップ数に比例するため，通信量削減効果が高くなるようにキャッシュ配置を工夫すること重要である．しかしながら，既存の取り組みでは，無数に存在するキャッシュ配置の組み合わせから通信量削減効果の高い配置を決定するための計算コストが大きく，長時間の計算を要する [7]．一方，ユーザは同じ動画を見続けるわけではなく，視聴される動画が時間経過とともに変化するため [13, 14]，キャッシュ配置計算に時間がかかると本来の最適配置との乖離が大きくなり，通信量削減効果が低下してしまう．

本論文では，短い計算時間で通信量削減効果の高いキャッシュ配置を計算し制御する，新しい分散協調キャッシュとその制御手法を提案し，現実的な評価環境でその効果を確認する．具体的には，以下の 3 つの課題を設定し，それぞれの解決策を組み合わせることで，効率的な分散協調キャッシュの技術を確立し，通信量削減効果の向上を図る．

(1) キャッシュサーバ単体で高いヒット率を維持できること

(2) 複数キャッシュサーバを協調させて実効キャッシュ容量を拡大すること

(3) キャッシュサーバ間のデータ共有に要する通信量を削減すること

具体的には，(1) の実現のため，異なる性質をもつキャッシュアルゴリズムを組み合わせたハイブリッドキャッシュアルゴリズムを提案し，急激にアクセス傾向が変化する状況下でもキャッシュヒット率を維持させる．次に，(2) の実現のため，コンテンツとキャッシュサーバをそれぞれグループに分けて，グループ ID に基づいてキャッシュ可否を決定することで，キャッシュサーバ間で異なるコンテンツが保持されるよう制御する．すると，(3) に示すように，キャッシュサーバ間でデータ共有通信が発生するため，アクセス頻度の高い一部のコンテンツは，複数のグループに所属するよう制御することで多数のキャッシュサーバで重複して保持されるよう調整し，キャッシュサーバ 1 台あたりのヒット率を向上させ，サーバ間通信を削減する．以上の仕組みを組み合わせて利用することで，軽量な計算で高い通信量削減効果を維持できる，ISP 内分散協調キャッシュ技術を確立する．日本の NTT ネットワークを模したネットワークトポロジと，代表的な VoD サービスである YouTube の動画アクセス傾向をもとにネットワークシミュレーションを実施し，数秒の計算時間で，準最適なキャッシュ配置との通信量削減効果の差を 3%程度に抑えるとともに，通常の制御では追従できないようなアクセス傾向の変化にも，ヒット率の下落幅を 2.3%程度にとどめ，通信量削減効果を維持できることを確認した．

1.2 論文の構成

本論文は，全 5 章で構成される．各章の構成を図 1.1 に示す．

第 2 章では，インターネット通信を削減するキャッシュネットワークに関する研究について述べ，既存の分散協調キャッシュの制御方法および，その問題点を述べる．また，動画を効率良くキャッシュする動画配信基盤を実現するため，動画配信のアクセス傾向についての調査論文について述べる．

第 3 章では，急激に傾向が変化する動画アクセスを効率よくキャッシュする，キャッシュアルゴリズムを提案する．複数キャッシュサーバ環境下でも協調動作できる仕組みを設計しておき，単一のキャッシュサーバで急激に変動する動画アクセスに追従できる方法を確立する．

第 4 章では，小さな計算コストで効率よく動画をキャッシュする，分散協調キャッシュの制御方法について述べる．第 2 章で述べるキャッシュ最適配置の特徴を考察し，コンテンツとサーバのグループ制御を行うことで，軽量な計算コストで高いヒット率を達成する

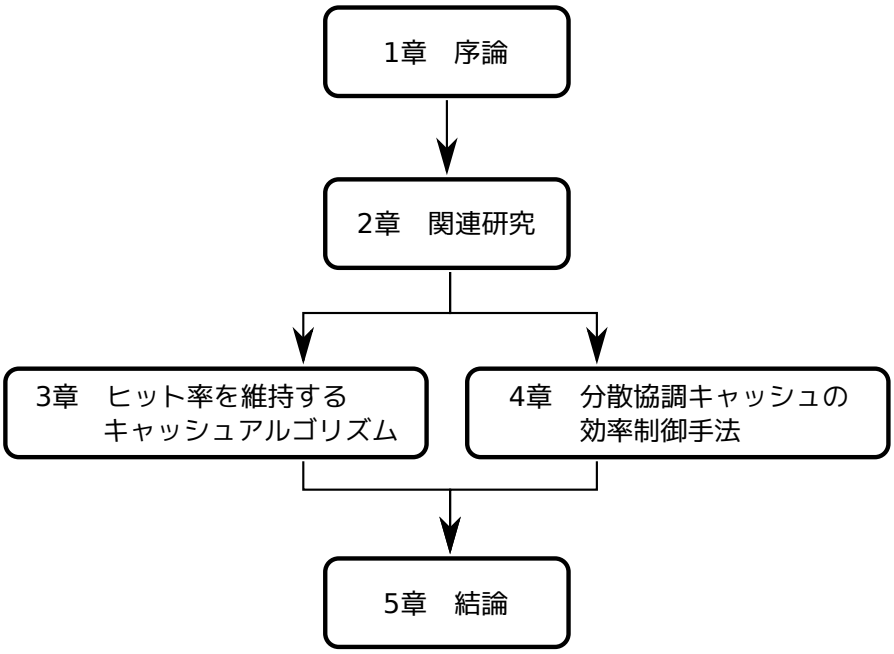


図 1.1: 論文の構成

分散協調キャッシュの制御方法を提案する．

第 5 章では，本研究の成果をまとめ，今後の展望を述べる．

第2章　関連研究

2.1　動画配信ネットワークの構成

動画配信ネットワークは，動画配信サーバを最上位とする階層ネットワークで構成されている（図 2.1）[15]．動画配信サーバは，映像制作会社や，映像制作会社から配信を委託された動画配信サービス事業者が管理するサーバで，配信対象の動画コンテンツを保持している．このサーバから直接ユーザに動画を配信すると，世界中の視聴者からのアクセスが集中し，高負荷でサーバが停止するおそれがあるだけでなく，遠隔地へのデータ転送が遅く，再生品質の低下につながってしまう．動画配信サーバは複数のデータセンタ拠点に分散して運用されることもあるが，配信拠点を増やすと運用コストが大きくなるため，通常は CDN 事業者に配信を任せた方がコストメリットが大きい．そのため，動画配信サーバは CDN 事業者に広域配信を委託し，サーバ負荷の低減と配信品質向上を図っている．Netflix のように動画配信事業者がキャッシュサーバを設置して CDN を構築 [16] している場合は，CDN の管理主体が動画配信事業者になる．ネットワークの構造は図 2.1 と共通である．

CDN 事業者は，多拠点にキャッシュサーバを展開して大規模なキャッシュネットワークを構築することで，限られた場所に位置する配信サーバが提供するコンテンツを広域配信する．世界最大規模の CDN を提供する Akamai は，102 カ国 1,300 以上のネットワークに合計 17 万台以上のキャッシュサーバを配置しており，全世界の Web 通信量の 15-30%の配信を担っている [17]．Amazon や Google も数十以上の拠点にキャッシュサーバを配置して CDN を展開しており [18, 19]，大容量動画データや OS のアップデートパッチを配信している．このような大容量コンテンツを配信する Web サイトは，Web ページを構成する HTML ファイル内に CDN 事業者に配置されたコンテンツへのリンクが記載されている．CDN に配置されたデータ（例:`http://cache.examplecdn.com/file.mp4`）をユーザが取得する際は，まず，DNS サーバを利用してホスト名を宛先 IP アドレスに解決する．ここ

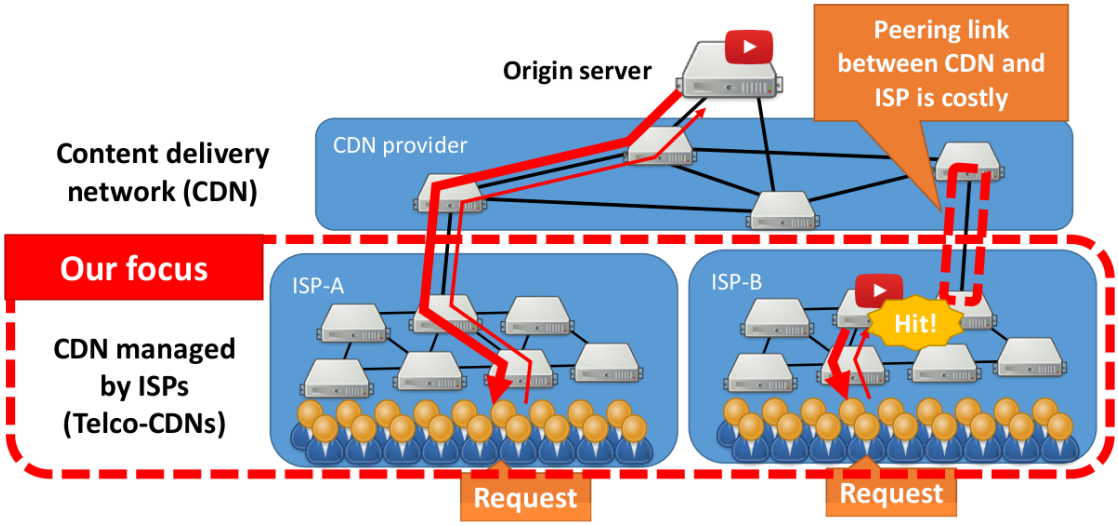


図 2.1: 動画配信ネットワークの構成

で，キャッシュサーバへの名前解決は CDN 事業者が管理する DNS サーバが利用され，アクセス元 IP アドレスに最も近いキャッシュサーバが DNS から返答される [17, 2]．このようにすることで，各地域のユーザは近くのキャッシュサーバを利用でき，通信量と取得遅延を削減できる．しかしながら，CDN 事業者が設置するキャッシュサーバは通常，各国に数拠点程度しかなく，キャッシュサーバと ISP との接続点に通信が集中しやすい構造にある [7]．また，これらのキャッシュサーバは人口密集地に設置されることが多いが，キャッシュサーバ間の通信帯域は必ずしも潤沢ではない．その結果，通信量削減を目的としてキャッシュサーバ間でデータを共有しようとすると，サーバ間を接続する通信経路が混雑するおそれがある．また，キャッシュサーバ間の通信経路がトランジット通信リンク（ISP 間の従量課金リンク）を含むと，ISP が高額な通信料金を請求されてしまう [7]．

CDN 事業者のキャッシュサーバは，ISP ネットワークを介して視聴者にデータを配信する．ユーザは ISP とインターネット回線を契約して利用するため，動画配信ネットワークにおいてユーザに最も近いのが ISP ネットワークである（図 2.1）．ISP ネットワークは国内〜地域規模のネットワークで，その規模や地形によってメッシュやリングのトポロジで構成されていることが多い．現在は ISP ネットワークにキャッシュサーバが多数配置されていることはほとんどないが，今後通信量が増大すると，ISP ネットワークに要求される転送性能や上流ネットワークに支払う通信料金も増加するため，将来的には ISP ネットワーク内にキャッシュサーバが設置されると言われている [7, 5]．階層ネットワーク構造

では，最下層にキャッシュサーバを配置することで上流の通信量を大幅に削減できるため [20]，ネットワーク構造の観点からも ISP ネットワークにキャッシュサーバを配置することは合理的である．

たとえば，総務省では，2020 年の東京オリンピックに向けて，4K 品質の動画配信基盤の構築を推進しているが [21]，4K 画質の高品質動画は圧縮技術を駆使しても配信に 85Mbps 程度の通信帯域が必要と言われており [22]，現在主流の SD 画質の動画と比較すると 10 倍程度の通信帯域を要する．東京都豊島区で 4K 動画を配信することを想定し，動画配信に必要な通信帯域を試算すると，H.264 形式で圧縮された 4K 高品質動画のビットレートを 85Mbps，豊島区の人口密度を 22,650 人/km^2，動画の同時視聴者数を人口の 5%として，1km^2 あたり約 100Gbps の配信性能が必要である（表 2.1）．ISP のインターネット接続点（他の ISP や Internet Exchange (IX) との接続点）の合計帯域は数百〜数千 Gbps 程度であるため [23]，日本全国で高品質動画を配信するためには，ネットワークとサーバの両方に莫大な投資が必要である．高速な学術情報ネットワークを提供する SINET4 の東京〜大阪間を接続するリンクの通信帯域が 40Gbps[24]，その後継の SINET5 の拠点間接続リンクの通信帯域が 100Gbps 程度 [25]（いずれも図 2.1 中の ISP ネットワークに相当する）であることを考えると，日本国内に数拠点しかない CDN キャッシュサーバだけで全国の視聴者に快適な動画視聴環境を提供することは容易ではなく，ISP 内部にキャッシュサーバを多数配置した通信量削減は極めて重要な課題である．

表 2.1: 同時視聴人口の割合と 1km^2 あたりの必要帯域

同時視聴人口の割合	視聴者数（人）	必要帯域（Gbps）
0.05	1,133	94
0.10	2.265	188
0.15	3,398	282
0.20	4,530	376
0.25	5,663	470

2.2 動画配信サービスのアクセス傾向

動画配信サービスの通信量を効率よく削減するためには，そのアクセス傾向を理解することが不可欠である．本節では，動画アクセスの偏りと，アクセス傾向の変化の 2 つの観点から動画配信サービスのアクセス傾向について述べる．

2.2.1 動画アクセスの偏り

通常，ユーザからの動画リクエストは人気のある動画に偏っており，それ以外動画はほとんどアクセスされない．そのため，アクセス頻度の高い動画を優先的にキャッシュすることで，効率よく通信量を削減できる．YouTube のアクセス傾向を調査した論文 [26] によると，動画のアクセスは一部の人気動画に集中しており，動画ライブラリ全体の 10%程度のコンテンツがアクセス全体の約 70%を占める．これは，ユーザの動画視聴傾向として，一度視聴した動画をその後何度も視聴されることは少なく [27]，新規に追加された動画や視聴数ランキング上位の動画がリコメンド機能を介してユーザに頻繁に提示されることに起因する．また，一般的な Web サイトのアクセス数がロングテールな Zipf 則でモデル化されることが多いのに対し，動画のアクセス数はより偏りが大きいガンマ分布でモデル化した方が，現実の VoD アクセスに近いことが示されている [26, 28].

図 2.2 に，コンテンツ総数を 1,000 種類として，Zipf 則とガンマ分布でモデル化した動画アクセスの累積確率分布を示す．ここで，動画アクセスの累積確率分布は，人気上位コンテンツをキャッシュしたときのヒット率の理論値と等しい．例えば，キャッシュサーバで上位 10%の動画を保持できた場合，Zipf 則では全体の約 35%のアクセスがキャッシュヒットするが，ガンマ分布ではキャッシュヒット率が 70%に達する．代表的な VoD サービス事業者の Netflix が各国に配置しているキャッシュサーバのストレージ容量は 1 台あたり 10%程度であるため [16]，このキャッシュサーバを 1 台導入すると，最大 70%程度の通信量が削減できる計算になる．すなわち，動画アクセスは偏りが大きいため，人気上位の動画のみをうまくキャッシュできれば，高い通信量削減効果を達成できる．一方で，人気のない動画をキャッシュしてストレージ容量を圧迫してしまうと，アクセス頻度の高い動画をキャッシュするためのストレージ容量が不足するため，通信量削減効果を高めるには，アクセス状況に合わせてアクセス傾向の高い動画を選択してキャッシュする仕組みが重要である．キャッシュサーバで利用されるキャッシュアルゴリズムについては後述する．

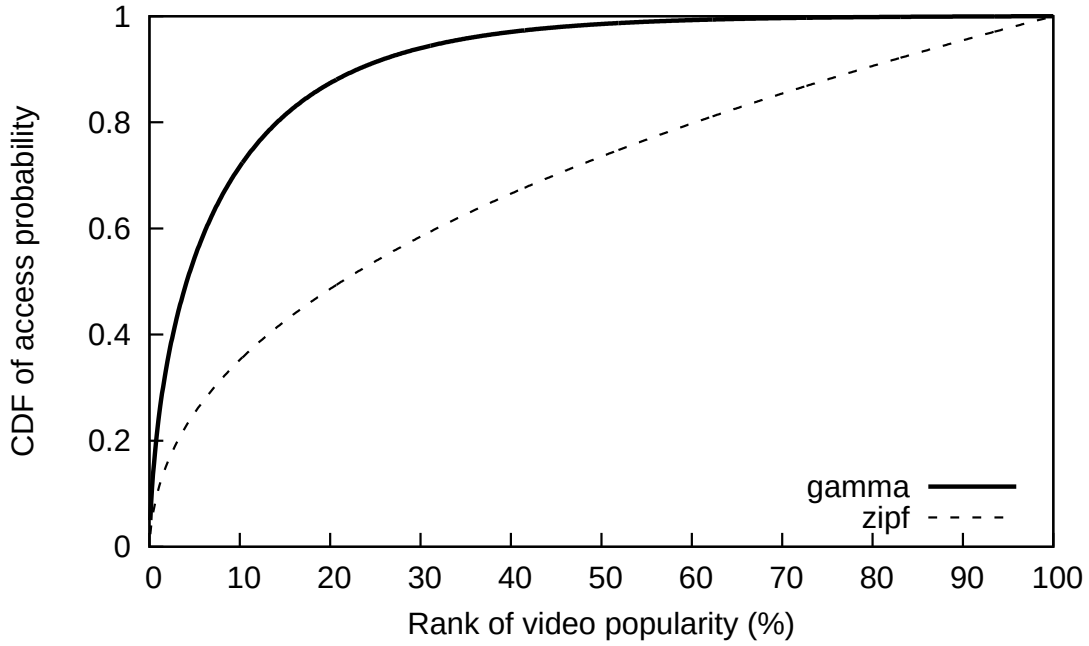


図 2.2: Zipf 則とガンマ分布による人気動画の偏り

2.2.2 動画アクセス傾向の変化

動画アクセス傾向が変化しない場合は，キャッシュサーバに最もアクセス頻度の高い動画を固定して保持させれば通信量削減効果を最大化できる．しかしながら，ユーザがアクセスする動画は時々刻々と変化するため，キャッシュサーバはアクセス傾向に合わせて保持する動画を入れ替えなければならない．中国の VoD サービスのアクセス傾向を調査した論文 [13] によると，同じユーザが同じ動画を何度も視聴することは少なく，時間帯によってユーザが視聴する動画の種類が変化するため，1 時間で動画アクセスの 40-60%が変動する．そのため，長時間同じ動画をキャッシュし続けると，キャッシュサーバを多数配置しても通信量削減効果が低下してしまうため，アクセスごと，または一定の時間間隔を設定し，キャッシュサーバで保持する動画の更新が必要である．一定間隔でキャッシュ内容を更新する場合は，更新間隔が長すぎると低いキャッシュヒット率を維持してしまうため，注意が必要である．

2008 年に開催された北京オリンピックの動画配信ログを解析した論文 [14] では，特定のシーンへのアクセスが 10 分程度で急激に上昇する状況が観測されている．このような急激な動画アクセスの変化に追従してキャッシュを制御できなければ，ヒット率が低下し

て通信量が増大するだけでなく，動画配信サーバにアクセスが集中して大きな負荷がかかり，サービスが停止する恐れもある．Twitter のハッシュタグの伝搬速度を解析した論文 [29] によると，タグ出現後 5 分で国内外に，2 時間で全世界に伝搬される様子が示されており，キャッシュサーバは俊敏にアクセス傾向の変化に追従しなければならない．

2.3 動画アクセス傾向に合わせたキャッシュアルゴリズム

2.3.1 アクセス頻度に着目したアルゴリズム

キャッシュ容量には制限があるため，アクセス頻度の低いコンテンツを優先的に追い出し，アクセス頻度の高いコンテンツをキャッシュに保持するよう制御しなければならない．動画アクセス傾向の変化速度が緩やかな場合は，アクセス履歴やリクエスト推移モデルを使用して将来のアクセス頻度を予測し，その結果に基づいたキャッシュ制御が有効である．たとえば，Least Frequently Used（LFU）アルゴリズム [30] は，アクセス頻度の低いコンテンツをキャッシュ領域から追い出すことで，ヒット率を高く維持するキャッシュアルゴリズムである．具体的には，キャッシュ領域に格納されたコンテンツそれぞれにアクセス頻度を格納する変数を用意しておき，新しくコンテンツを追加する際は，アクセス頻度が最も小さいコンテンツをキャッシュから追い出す．LFU は，アクセス傾向が変化しない状況下では通信量削減効果が理論上最大となることが知られているが，アクセスが発生するごとにキャッシュ上のコンテンツのアクセス頻度を再計算する必要があるため，計算負荷が大きく，実環境で LFU を採用する例は多くない．LFU の制御負荷を削減する TinyLFU アルゴリズム [31] も提案されている．TinyLFU は，LFU と同じくアクセス頻度が高い動画をキャッシュするが，一定時間ごとにアクセスログを解析して保持する動画を固定する．このようにすることで，毎回のアクセスでコンテンツのアクセス頻度を再計算する必要がなくなり，制御負荷が低いだけでなく，動画アクセス傾向が大きく変化しない状況では通信量削減効果も高い．

しかしながら，動画アクセス傾向は短時間で変化するだけでなく，状況によっては急激なアクセス集中が発生することもある．先に述べた LFU ベースのアルゴリズムでは，過去に頻繁にアクセスされた動画がキャッシュに残りやすく，新しい動画をキャッシュしても最初のうちはすぐに追い出されてしまい，アクセス傾向の変化に追従しにくい．また，アクセス傾向の変化が予測できる場合は LFU ベースのアルゴリズムと組み合わせた予測

キャッシュが有効であるが，予測結果が不正確であった場合や，事前予測が不可能な突然のニュース等に影響されると，ヒット率が大幅に下落してしまう．

2.3.2 直近のアクセスに着目したアルゴリズム

ニュースや SNS の動向を事前に予測することは容易ではないため，通常，CDN 事業者は，Least Recently Used（LRU）ベースのアルゴリズムを採用している．LRU は，直近アクセスされたコンテンツはすぐにまたアクセスされると考えて，最近最もアクセスされていないコンテンツを優先してキャッシュから追い出すアルゴリズムである．そのため，アクセス傾向が急激に変化しても新しいアクセス傾向に追従しやすく，特定の動画が急激に人気になっても効率よくその動画をキャッシュし，配信サーバへのアクセス集中を回避することができる．

しかしながら，純粋な LRU はアクセス頻度を考慮しないため，LFU と比較してヒット率が低く，通信量削減効果が不十分である．Li ら [10] は，ISP 内 CDN を対象とした LRU ベースのアルゴリズムを提案している．彼らが提案する Modified LRU アルゴリズムは，キャッシュ位置を調整するための jump パラメータを使用して，コンテンツの新規挿入位置およびキャッシュヒット時のキャッシュ位置の更新を LRU よりも追い出されやすい位置に変更する．jump パラメータは整数値が設定され，新規に挿入されるコンテンツは LRU リストの Most Recently Used (MRU) 側から jump だけ LRU 側に移動した場所に挿入される．また，キャッシュヒットしたコンテンツは，もとの位置から jump だけ LRU 側に移動される．このように制御することで，1 度しかアクセスされないようなコンテンツを追い出されやすい位置に保存し，ヒット率向上を図っているが，アクセス傾向が一定期間変化しない場合は，LFU の通信削減量には及ばない．

2.3.3 異なるキャッシュ制御を組み合わせたハイブリッドアルゴリズム

動画のアクセス頻度は，アップロードされると急激に上昇し，その後下落するよう推移する．このアクセス推移は動画のカテゴリによって異なり，例えば映画や音楽コンテンツはアクセス頻度が長期間持続しやすく，ニュースやスポーツに関する番組はアクセス頻度が下落しやすい．そこで，Zhou ら [32] は，LFU と FIFO を組み合わせることで，動画のアクセス頻度の推移が異なる動画を効率よくキャッシュするアルゴリズムを提案している．

このキャッシュアルゴリズムは，人気のあるコンテンツを LFU 領域で保持することでヒット率を高くし，アクセス頻度の計算に十分なアクセスがない最近追加された動画を FIFO 領域でキャッシュする．

HDD のバッファキャッシュを対象として，LFU と LFU を組み合わせた LRFU アルゴリズムも提案されている [33]．これは，過去のアクセス傾向と直近のアクセス傾向に重みを設定することで，LRU と LFU の両方の特徴を調整可能なキャッシュアルゴリズムである．LFU/FIFO ハイブリッドキャッシュがキャッシュ領域を 2 つに分割して制御できるのに対し，LRFU アルゴリズムはキャッシュ領域は 1 つのまま，追い出しに必要なコスト計算部分のみを変更している．

2.3.4 将来のアクセス傾向を予測するキャッシュ制御手法

動画のアクセス傾向がある程度予測できる場合は，将来の動画アクセス頻度をもとにキャッシュするコンテンツを選択する方法が提案されている [4, 27, 34]．これらの方法は，動画のアクセス数が何らかの数理モデルで表現されると仮定し，直近のアクセス数をもとに推移モデルのパラメータを決定することで，将来のアクセス数を推定する．具体的には，アクセス回数が増加傾向にある動画 A と減少傾向にある動画 B があるとき，動画 B をキャッシュから削除して動画 A をキャッシュすることでヒット率を改善する．しかしながら，これらの手法では動画アクセスの急激な人気変動を想定しておらず，新規動画の追加，視聴者に勧める動画リストの変更，SNS やニュース等の影響を受けた急激な人気変動への対応が十分に検討されていない．動画によってはアクセス推移が大きく異なる場合があるだけでなく，新規動画の追加や，突然のニュース，Twitter や Facebook などの SNS の影響を受けると，予期せず特定の動画にアクセスが突然集中することもある．そのため，キャッシュサーバは変化するアクセス傾向に合わせて，予測結果に頼らずヒット率を維持できる必要がある．

2.4 動画配信ネットワークに特化したキャッシュ制御手法

2.4.1 階層キャッシュネットワークおけるキャッシュ制御手法

キャッシュサーバを複数台並べて多段構成にすることで，実効キャッシュ領域を拡大してヒット率を向上できる [35]．しかし，下段のサーバでキャッシュされたコンテンツへのアクセスは上段のサーバに到達しないため，上段と下段で同じコンテンツを保持すると実効キャッシュ領域が減少し，意図したヒット率が得られない．そこで，階層キャッシュにより拡大したキャッシュ領域を有効活用するために，各階層で異なるコンテンツを保持する仕組みが提案されている [36, 37, 20, 38]．

[36] では，上段と下段で異なるキャッシュアルゴリズムを選択したヒット率改善を検討している．例えば，LRU はアクセス間隔の短いコンテンツをキャッシュするため，上段へ通過するアクセスは間隔が長くなる．アクセス間隔の長いコンテンツは LRU でキャッシュしにくいため，上段サーバのキャッシュアルゴリズムも LRU だとヒット率が大幅に下落する．一方で，上段で採用するアルゴリズムを，長期的なアクセス傾向を考慮した LFU や，容量あたりのヒット率が高いコンテンツを優先してキャッシュするアルゴリズムを置いておけば，ヒット率の下落幅は小さくなる．このように，各階層で適切なキャッシュアルゴリズムを選択することで階層キャッシュのヒット率を改善できる．

コンテンツをキャッシュするかどうかを一定の規則に基づいて決定し，サーバ間でキャッシュ重複を緩和する手法 [37] も提案されている．上段のサーバが保持しているコンテンツのみ下段でキャッシュするなど，一定の規則に基づいてキャッシュするかどうかを決定することで，必要なコンテンツが不要なコンテンツに追い出される確率を低くできる．[20] では，Web サーバが作成した人気順位をもとに各階層でキャッシュする動画を決定しておき，ユーザに近いサーバでは人気動画のみをキャッシュするように制限することで，ヒット率を向上している．

以上のようにキャッシュを階層的に並べ，各階層で異なるコンテンツを保持させることで，キャッシュネットワーク内でのヒット率を向上し，配信サーバの負荷を低減できる．しかしながら，急激な動画の人気変動はいずれも考慮されておらず，突発的なアクセスが発生すると配信サーバに負荷がかかる可能性がある．

2.4.2 複数キャッシュサーバを組み合わせる分散協調キャッシュ制御

最近の VoD サービスでは，多数のユーザが新しく動画を投稿するため動画総数が増大している [39]．また，撮影機器の高性能化によって動画が高画質化し，動画 1 つあたりの容量も大きくなっている [1]．その結果，将来的には，1 台のキャッシュサーバを効率よく制御しても，限定されたキャッシュ容量では十分な通信量削減効果が得られなくなってしまう．また，ネットワーク中にキャッシュサーバを多数配置しても，制御アルゴリズムが共通していると，各キャッシュサーバでほとんど同じコンテンツを保持してしまい，キャッシュ容量を効率よく利用することができない．そのため，複数のキャッシュサーバで異なるコンテンツを保持して共有することで実効キャッシュ容量を拡大する，分散協調キャッシュの仕組みが提案されている．分散協調キャッシュの実現にはトラフィックエンジニアリングにより通信経路の変更が必要となるため，ネットワークとキャッシュサーバが同一業者の管理下にある，ISP 内キャッシュネットワークで利用される [7]．

動画 ID のハッシュ値をもとにした分散協調アルゴリズム

キャッシュサーバと動画にそれぞれ整数の ID を割り当て，ある整数 k の剰余が等しい場合のみキャッシュするよう規則を追加することで，キャッシュ容量を k 倍に拡大するアルゴリズムが提案されている [11]．k の値をサーバ台数に設定すれば，キャッシュサーバ台数分だけキャッシュ容量を拡大できるため，外部ネットワークとの通信量を大幅に削減できる．一方で，ユーザが要求するコンテンツが数台のサーバでしか保持されず，キャッシュネットワーク内の平均ホップ数が増大し，混雑リンクの発生や取得遅延の原因となってしまう．そこで，この手法を拡張し，近くのサーバを事前にグループ化しておくことで，ユーザからコンテンツまでの最大ホップ数を制限する手法も提案されている [12]．

大手 CDN 事業者の Akamai 社では，拠点内のキャッシュサーバをクラスタリングする際に，コンテンツ ID のハッシュ値をサーバ ID と比較し，実効キャッシュ容量を拡大するとともに，クラスタ内でのキャッシュサーバの負荷分散を図っている [17]．しかしながら，同一拠点のサーバでしかキャッシュが協調動作しないため，実効キャッシュ容量が制限される．その結果，十分な通信量削減効果を得るためには，拠点あたりの投資額が増大してしまう．

一方，ネットワークを介した別拠点のキャッシュサーバを協調動作させることを目的と

して，Wang ら [11] はリングネットワークを，Li ら [12] はメッシュネットワークを対象として，分散協調キャッシュを提案している．これらの研究では，前述のようにハッシュ値をもとにして各拠点のサーバが管理されており，実効キャッシュ容量の拡大に大きく寄与するため，ISP ネットワーク外部との通信量を削減できる．一方，あるコンテンツを保持するキャッシュサーバがネットワーク中に数カ所しかなく，ユーザの近くに人気コンテンツがない際の通信量ペナルティが大きい．また，人気動画を保持する特定のキャッシュサーバにアクセスが集中し，サーバ負荷の増大や混雑リンクの発生につながってしまう．

最適化アルゴリズムをもとにした分散協調アルゴリズム

上述のように，動画 ID のハッシュ値をもとに分散保持させる簡単なアルゴリズムでは，特定のサーバへの負荷集中や，混雑リンクが発生して取得遅延が発生するおそれがある．そこで，コンテンツ取得帯域，遅延，ネットワーク転送にかかる電力等に制約を与え，ネットワーク内外の通信量を最小化するキャッシュ配置を求める研究も実施されている．これらの研究では，各サーバがキャッシュするコンテンツの組み合わせを多数用意し，遺伝的アルゴリズムで準最適解を求める方法がとられている [4, 40]．しかしながら，これらの研究で扱うキャッシュ配置問題は，容量制約付き施設配置問題（Capacitated Facility Location Problem; CFLP）に変形できる．CFLP は NP 完全であることが知られており [4]，このような多数の制約条件下で最適化問題を解くことは容易ではない．

Li ら [4] は，フランスのネットワークを対象に遺伝的アルゴリズムで最適キャッシュ配置を求めている．1 台で計算すると長時間の計算が必要となるため，10 台の PC で構成した PC クラスタで計算時間の短縮を図っている．しかし，そのような PC クラスタを利用してもなお 10 時間程度の計算時間を要している．遺伝的アルゴリズム中でのキャッシュ配置の評価には過去のアクセスログから生成したアクセスパターンが利用されるが，VoD のアクセス傾向は 1 時間で 40-60%程度変わってしまうため [13]，過去のアクセスログをもとに求めたキャッシュ配置はその時点での最適配置と乖離が大きく，計算終了時点で十分に通信量を削減できるキャッシュ配置ではなくなってしまう．

2.5 分散協調キャッシュの経路制御手法

2.5.1 コンテンツ ID のハッシュ値による経路制御

分散協調キャッシュの仕組みを実現する上で，ユーザからのリクエストを適切なキャッシュサーバへルーティングすることは通信量削減効果を大きく左右する．動画 ID のハッシュ値に基づくキャッシュ配置 [17, 11, 12] では，動画 ID から計算したハッシュ値で対応するキャッシュサーバのリストが求まるため，その中から最短ホップで到達できるサーバへリクエストを転送すればよい．具体的には，サーバ ID のハッシュ値とリクエスト転送先のネットワークインタフェースの組を記載した表を用意しておき，キャッシュサーバがコンテンツリクエストを受け取ったら，その ID をハッシュ関数に入力し，出力されたハッシュ値に対応するインタフェースからリクエストを創出する．動画 ID とサーバ ID は通常変更されないため，経路制御表は一度作成すれば長期間再利用できる．また，経路制御時に動画 ID のハッシュ値を計算するだけなので簡単に実現でき，経路制御の負荷が小さく抑えらる．ただし，キャッシュ配置自体がコンテンツのアクセス頻度を考慮していないため，特定のノード付近のリンクに負荷が集中してしまう．リンク負荷を分散するよう経路制御を工夫できたとしても，経路制御だけでは根本的解決に至らない．

2.5.2 コンテンツ配置場所に基づく経路制御

最適化アルゴリズムによって決定されたコンテンツ配置 [4, 40] は通信量削減効果が高い一方，どのコンテンツがどのキャッシュサーバに配置されているかの探索が難しい．特定の規則に基づいて配置が決定されているわけではないため，コンテンツ ID とキャッシュサーバが直接対応する表を保持しなければならない．また，アクセス傾向の変化に合わせてキャッシュ配置が再計算されると，それまで利用していた経路制御表と齟齬が発生してしまうため，経路制御表も合わせて更新しなければならない．しかし，継続的な動画の追加に合わせて経路制御表のエントリ数も増大するため，制御規則の更新データ量が増大する上，経路制御の探索負荷も大きく，現実的ではない．

第3章　アクセス傾向の変化に追従するハイブリッドキャッシュ制御

3.1　動画アクセス傾向とキャッシュアルゴリズムの考察

2 章で述べたように，動画のアクセスは (1) 人気上位コンテンツにアクセスが大きく偏っており，(2) 新規コンテンツの追加やニュース等の影響でアクセス傾向が急激に変化する．特に，(2) のアクセス傾向は，一定期間のアクセスログを必要とする既存の予測アルゴリズムでは対応が遅れ，通信量の増大を招く．また，キャッシュサーバ 1 台ではキャッシュ容量が小さく，通信量削減効果が小さいため，(3) 複数のキャッシュサーバを組み合わせて利用する分散協調キャッシュの併用が必要である．そこで，上記 (1)(2) の課題に合わせて適切なキャッシュアルゴリズムを選択し，組み合わせて利用することで，高いキャッシュヒット率を維持するハイブリッドキャッシュアルゴリズムを検討する．その後，複数キャッシュサーバ環境でのシミュレーション評価を通し，検討したハイブリッドキャッシュアルゴリズムが (3) 分散協調キャッシュで利用できることを示す．ただし，3 章では基本的性質を明らかにすることを目的とし，分散協調キャッシュ制御を組み合わせたキャッシュ制御方法については 4 章で述べる．

動画配信サービスで利用されるキャッシュアルゴリズムは，直近のアクセスに着目してキャッシュを制御する LRU ベースのアルゴリズムと，アクセス頻度に着目してキャッシュを制御する LFU ベースのアルゴリズムに分類できる．LFU は長期的な，LRU は短期的なアクセスログをもとにキャッシュを制御するため，それぞれのアルゴリズムでキャッシュが有効にはたらくアクセス傾向が異なる．上記 (1) で述べた偏りの大きいアクセスは，人気上位コンテンツをキャッシュすることで，高い通信量削減効果を達成できる．すなわち，過去のアクセスログを解析してアクセス頻度の高いコンテンツを優先して保持する，LFU をベースとしたキャッシュアルゴリズムの利用が有効である．実際，VoD サービスの動画アクセス傾向は，少数の人気動画に大きく偏っており [26, 28]，人気上位 10%の動画がア

クセス全体の 72%を占める．そのため，人気上位 10%の動画を保持できる LFU 容量を確保できれば，通信量を 72%削減可能である．一方，(2) で述べた急激なアクセスの変動は，過去のアクセス傾向の影響を受ける LFU ベースのアルゴリズムでは追従できない．そこで，直近のアクセス傾向をもとにキャッシュを制御する LRU アルゴリズムを組み合わせて利用することで，急激なアクセス傾向の変化に追従するハイブリッドキャッシュアルゴリズムを考える．また，実効キャッシュ容量を拡大するために，複数キャッシュサーバの LFU 領域を組み合わせて利用する．

3.2 異種キャッシュを混在させたキャッシュネットワーク

3.2.1 LRU と LFU を組み合わせたキャッシュネットワーク構造

アクセスネットワーク，メトロネットワーク，コアネットワークのように多段構成のネットワークを対象として，各階層にキャッシュを配置することを想定し，VoD 通信に適した階層キャッシュの仕組みを検討する．キャッシュを多段に並べることで，利用可能なキャッシュ領域を大きくして通信量を削減するとともに，階層間で異なるキャッシュアルゴリズムを採用することで，急激なアクセス傾向の変動に追従する．具体的には，大容量な LFU ベースのキャッシュ領域で高いヒット率を実現しつつ，急激な人気変動に備えて少量の LRU キャッシュ領域を組み合わせて利用することで，変動に強いキャッシュネットワークを構成する．

異種キャッシュの組み合わせ方は，図 3.1 に示す 2 種類を検討し，それぞれ効果を確かめる．実際の動画配信ネットワークのトポロジは単純な階層構造ではなく，メッシュやツリー，リングネットワークを複合した複雑な構造をしているが，3 章では異種キャッシュを組み合わせたキャッシュネットワークの基本的な性質を明らかにするため，複数のキャッシュサーバを多段に接続したネットワークのみを扱う．なお，各階層のキャッシュサーバにそれぞれユーザを接続すると，キャッシュサーバに到達するリクエストの性質が分かりにくくなるため，最下段のキャッシュサーバにのみユーザを接続し，リクエストを発生させる．3.2.1 節では，図 3.1 右の方針 1 に示すように，キャッシュサーバを階層的に接続した多段キャッシュネットワーク構成を検討する．キャッシュアルゴリズムにはヒット率の高い LFU ベースのアルゴリズムを採用し，3 台中 1 台を LRU で制御することで，高ヒット率を変動追従性を両立する．このように階層ごとに異なるキャッシュアルゴリズムを設

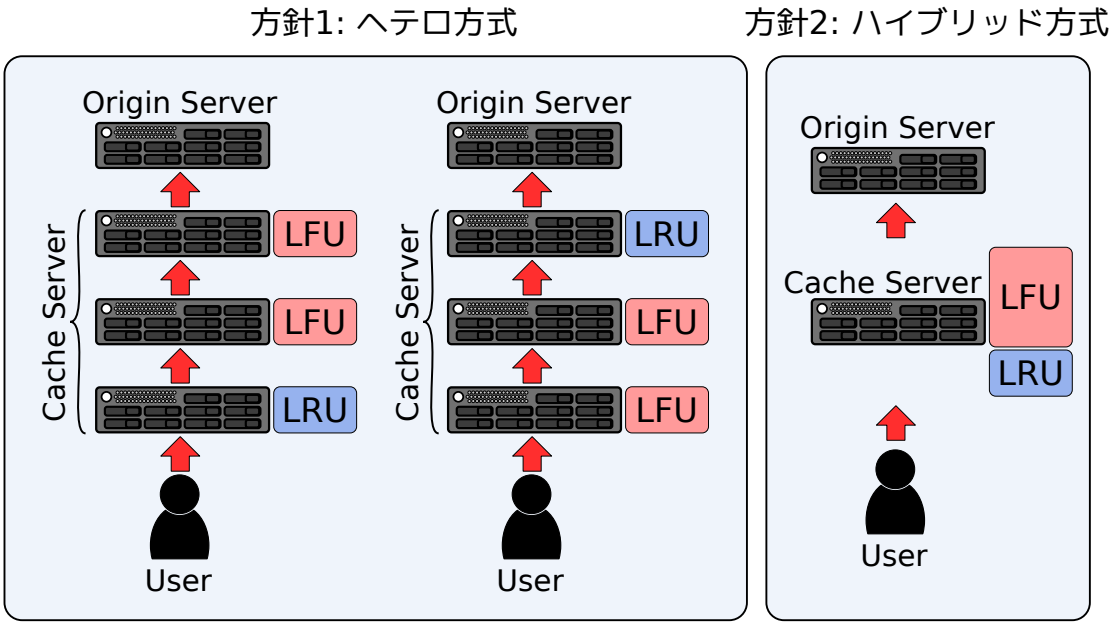


図 3.1: 異種キャッシュを利用する 2 つの方針

定する方法では，既存のキャッシュサーバの置き換えアルゴリズムを変更して多段に並べればよいため，サーバプログラムに大きな変更を加える必要がなく，実装コストが小さい．以下ではこの方針を「ヘテロ方式」と記載する．

なお，LFU 領域は，アクセスログから生成した人気コンテンツのテーブルを一定時間ごとに更新し，テーブルに存在するコンテンツのみをキャッシュする．LFU のキャッシュ制御を図 3.2 に示す．LFU は一定時間ごとにアクセスログを集計し，アクセス頻度の高いコンテンツから順に一定時間固定して保持する．キャッシュ対象のコンテンツは図 3.2 左に示すキャッシュテーブルに保存されており，テーブルに含まれるコンテンツのみがキャッシュされる．ただし，キャッシュテーブルの更新時にテーブルから削除されたコンテンツは，新規コンテンツが挿入されてキャッシュ容量が不足するまで削除しない．このため，テーブルから既に削除されたコンテンツでもキャッシュ領域に存在していればキャッシュヒットする．キャッシュの追い出しは，テーブルから削除されたコンテンツをキャッシュ領域からランダムに 1 つ選択して削除する．このように実装することで，小さな演算コストで LFU と同等のヒット率を達成できる [41].

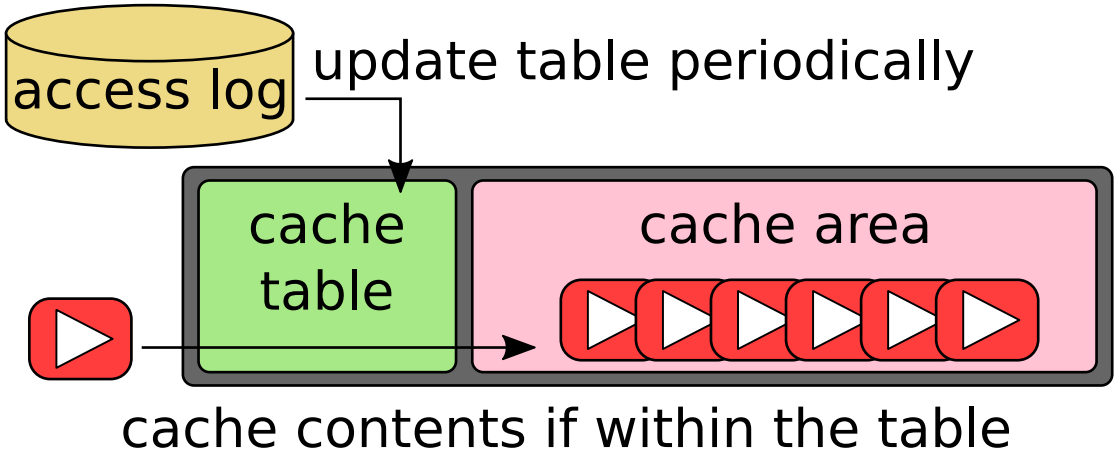


図 3.2: LFU アルゴリズムのキャッシュ制御

3.2.2 階層ネットワークにおけるキャッシュアルゴリズムの設置順序

階層ネットワークでは，アクセス頻度の高い動画をユーザから近い位置にキャッシュすることで，キャッシュヒットしたサーバよりも先のネットワークへの通信を大幅に削減できる．一方で，下層で採用するキャッシュアルゴリズムの効率が悪く，アクセス頻度の低い動画を多数キャッシュしてしまうと，キャッシュミスが頻発し，通信量の削減効率が大きく下がってしまう．また，ユーザから遠い位置に人気動画が配置されていると，動画配信サーバへのアクセスは削減されるが，ユーザとキャッシュサーバ間の通信量が増大してしまう．そのため，アクセス頻度の高い動画ほどユーザ近くでキャッシュするよう制御することが望ましい．

また，階層構造に並べられたキャッシュネットワークでは，ユーザがキャッシュサーバにリクエストを送信し，キャッシュヒットすればコンテンツが返答され，キャッシュミスすればさらに上段のキャッシュサーバにリクエストを転送する．その結果，上段へリクエストされる動画アクセス傾向は，もとのリクエスト傾向から下段でキャッシュヒットしたアクセスを除いたものになり，上段キャッシュサーバへ転送されるリクエストのアクセス傾向は最初と異なるものになる [36]．キャッシュヒット率は，キャッシュ容量，キャッシュアルゴリズム，アクセス傾向の組み合わせによって定まるため，異なるキャッシュアルゴリズムを採用したキャッシュサーバの並べ方によって，各サーバにおけるヒット率が変わる．すなわち，キャッシュアルゴリズムの並べ方で，キャッシュネットワークで生成される内部通信量と，動画配信サーバが送出するデータ量が異なるはずである．

たとえば，図 3.1 方針 1 の右側のように，下段と中段を LFU，上段を LRU にすると，下段と中段で人気のコンテンツをキャッシュしつつ，上段で急激な人気変動に対応できる．しかし，人気変動時には LFU の人気順位の更新が間に合わず，下・中段の 2 台のサーバを経由するため，通信量が増大してしまう．一方，図 3.1 方針 1 の左側のように，LRU を下段に設置し，中段と上段を LFU にすると，アクセス傾向が一定の場合にも中段以上から取得する動画の順位が一意に定まらないため，中段と上段の LFU のヒット率が下がる．しかし，人気変動時も人気動画が下段でキャッシュされるため，中段と上段の LFU キャッシュヒット率とホップ数を維持できる．

3.3 異種キャッシュを組み合わせたハイブリッドアルゴリズム

3.3.1 LRU と LFU を組み合わせたハイブリッドアルゴリズム

異なる種類のキャッシュを組み合わせて利用する第 2 の方針として，単一のキャッシュサーバのキャッシュ領域を 2 つに分割し，大容量な LFU 領域と小容量な LRU 領域で混合する方法を検討する（図 3.1 方針 2）．2 つの領域で同じコンテンツをキャッシュすると，ストレージ利用効率が低下してしまうため，LFU 領域で保持するコンテンツはアクセス頻度が高い人気上位コンテンツで固定しておき，それ以外のコンテンツを LRU 領域でキャッシュするよう制御する．以下ではこの方針を「ハイブリッド方式」と記載する．先述したヘテロ方式と比較するとキャッシュ1 台あたりの構成が複雑になるが，LRU と LFU の混合割合を各キャッシュサーバごとに設定できるので，ヒット率向上のための調整が可能になる．また，下段をハイブリッド方式，中段と上段を LFU の多段構成にすれば，キャッシュ容量全体に占める LFU の割合を大きくできるため，人気変動が小さい場合も高いヒット率を維持できる．

ハイブリッド方式のキャッシュサーバは，図 3.3 に示すように，キャッシュ領域を LRU と LFU の 2 つの領域に分割して制御する．LFU 領域は LFU アルゴリズムで制御する．(1) キャッシュサーバが動画要求を受けると，(2) その要求をアクセスログに追記する．その後，(3)LRU 領域と (4)LFU 領域でそれぞれ動画がキャッシュされていないか確認し，キャッシュヒットすればキャッシュ領域から動画を取り出し，要求元に返答する．キャッシュ領域に動画がなければ，(5) 上段のキャッシュまたは動画配信サーバに要求を転送する．上段から返答された動画はが LFU 領域のキャッシュテーブルにマッチすれば，LFU 領域に追加

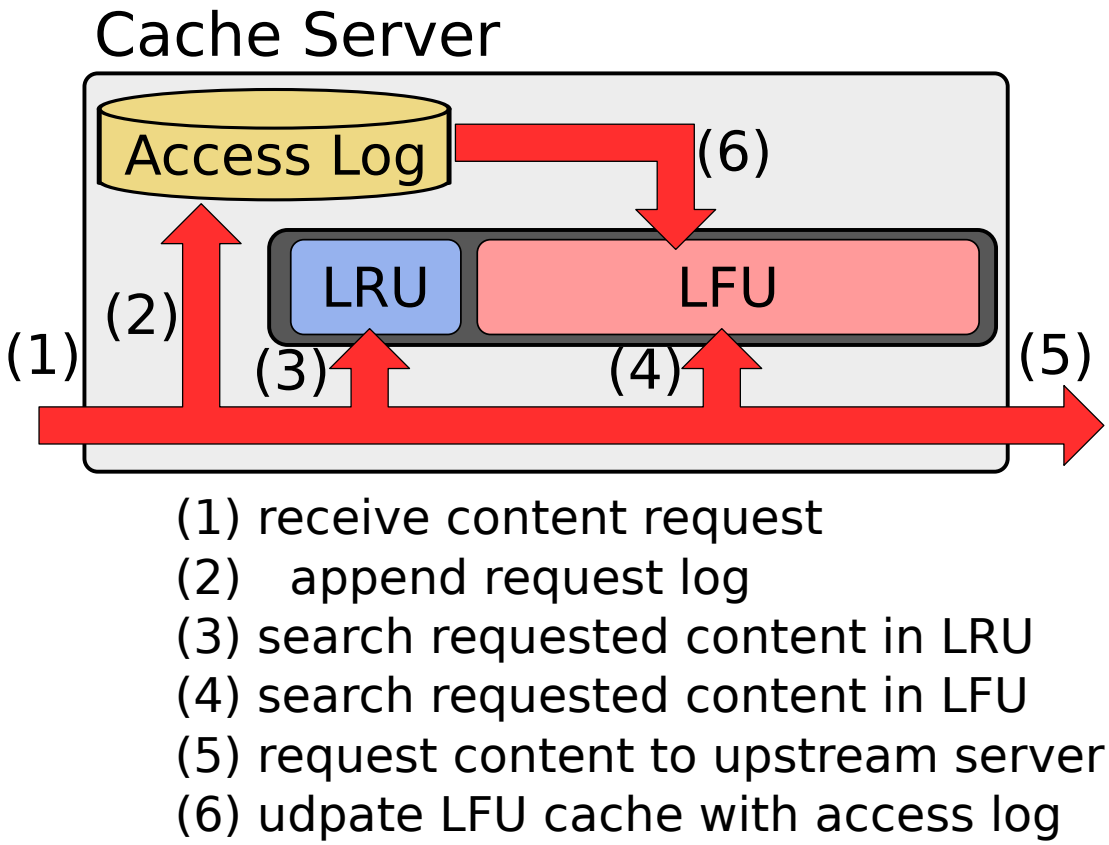


図 3.3: LRU/LFU ハイブリッド方式の動作

する．マッチしなかった場合は，LRU 領域でキャッシュする．また，(6) 一定時間ごとに，アクセスログを解析し，人気頻度の高い動画のみをキャッシュするよう LFU 領域のキャッシュテーブルを更新する．

3.3.2 ハイブリッドキャッシュ割合の設定方法

ハイブリッドキャッシュアルゴリズムの LRU/LFU 割合は，VoD サービスに追加される新規コンテンツの数によって設定する．新規コンテンツがほとんど追加されない状況下では，LFU 割合を多く設定することで高いヒット率を達成できるため，LRU 領域を小さくすることが望ましい．一方で，新規コンテンツが頻繁に追加される場合，LRU 領域が小さいと新規コンテンツをキャッシュできず，通信量が増大してしまう．そのため，VoD サービスのアクセスログ情報を解析したり，VoD サービス事業者が有する経験則に基づいて，新規コンテンツをキャッシュできる容量だけの LRU 割合を事前に設定することが望ましい．

3.4 評価

3.4.1 評価環境

評価に際して，LRU，LFU，ハイブリッド方式それぞれのキャッシュアルゴリズムを Ruby で実装し，ネットワークシミュレーションでヒット率とホップ数を評価した．LRU キャッシュは，最新のアクセス時刻が最も古いコンテンツを追い出して新しいコンテンツを挿入する，一般的な方法で実装した [42]．LFU キャッシュは，[41] をもとに一定時間ごとにアクセスログを解析し，キャッシュする動画を更新するよう実装した．

問題を簡単化するため，各動画アクセスは，直前の動画アクセスが完了するまで発生しないように設定し，すべての動画を同一ファイルサイズとした．また，大規模 VoD サービスの Netflix が展開しているキャッシュサーバの容量が 10%程度であるため [16]，ある時刻においてアクセスされる動画を 1000 種類，各キャッシュサーバは 100 種類の動画をキャッシュできるように容量を設定した．動画アクセスは文献 [26] に示されている YouTube のアクセスの偏りをもとにガンマ分布に従って生成した．ガンマ分布は，広く知られている Zipf 則にもとづくアクセス分布よりも，実際の動画アクセス傾向に近いアクセスを生成することができる．ガンマ分布の偏りパラメータは，文献 [26] と同じ値を設定し，$k = 0.475$, $\theta = 170.6067$ とした．LFU キャッシュのテーブルは過去 100,000 回のアクセスをもとに，100,000 アクセスごとに更新するよう設定した．

急激な人気変動時のヒット率と通信量を検討するため，ある時刻で新規コンテンツが追加される状況を設定する．まず，1000 種類の動画について 500,000 回のアクセスを発生させ，キャッシュ領域にコンテンツを保持させておく．その後，新規動画として 20 種類の動画を追加し，アクセス頻度を人気最上位と設定した．20 種類のコンテンツを後から追加した結果，動画総数が 1020 となるが，新規コンテンツ追加後は人気最下位の 20 動画はその後アクセスされないように設定した．すなわち，動画アクセスは常に 1000 種類の動画から生成され，そのアクセスの偏りはガンマ分布に従う．

階層キャッシュ構成の評価では，LRU キャッシュ，LFU キャッシュ，ハイブリッド方式のキャッシュサーバを直列に並べる．この際，キャッシュサーバ間でアクセスログの等の情報交換は行わない．そのため，各キャッシュサーバで保持される動画は，そのキャッシュサーバが受けた動画リクエストによってのみ決定される．

以降の評価において，ヒット率は，ユーザが発生した動画リクエストのうち，Web サー

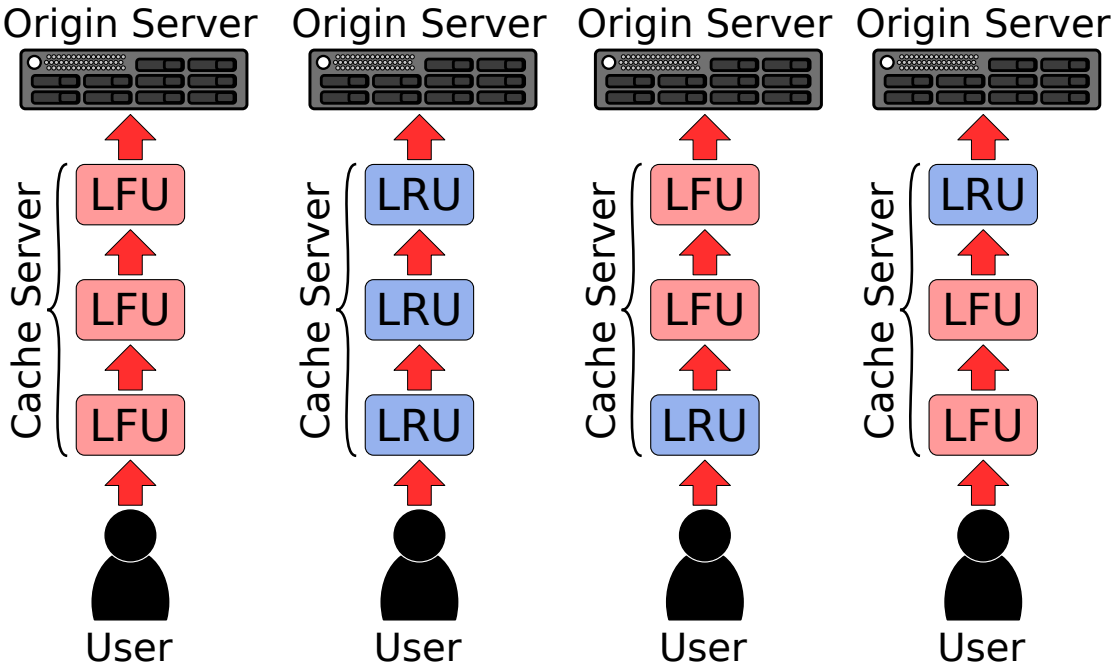


図 3.4: ヘテロ方式でのキャッシュアルゴリズムの並べ方

バに到達しなかった割合を示す．一方，通信量はホップ数に比例するため，1 つの動画を取得するのに必要な平均ホップ数を算出した．ホップ数は各階層のキャッシュヒット率をもとに算出し，アクセスが k 段目でヒットする確率に k 段目までのホップ数を乗じた値の総和を求めた．ユーザから直上のキャッシュサーバまで，キャッシュサーバ間，上段のキャッシュサーバから Web サーバまでのホップ数はそれぞれ 1 と設定した．

3.4.2　3 段構成のヘテロキャッシュネットワークにおける通信量

図 3.4 に示すようにキャッシュを直列に並べてアクセスを生成し，ヒット率をホップ数を評価した．ユーザから動画アクセスを発生させた際のヒット率とホップ数を図 3.5，3.6 にそれぞれ示す．凡例の括弧内は，左から順に下段，中段，上段のキャッシュアルゴリズムを示す．

人気変動がない状況下では LFU を 3 台並べた構成のヒット率が最も高いが，新規動画が追加されたタイミングでヒット率が下落する．この構成では人気上位 30%の動画を保持できるため，キャッシュヒット率の理論値は 94%である（図 2.2 を参照）．そのため，図 3.4 のアクセス変動前のヒット率は 92-94%程度を推移している． このとき，下段のキャッ

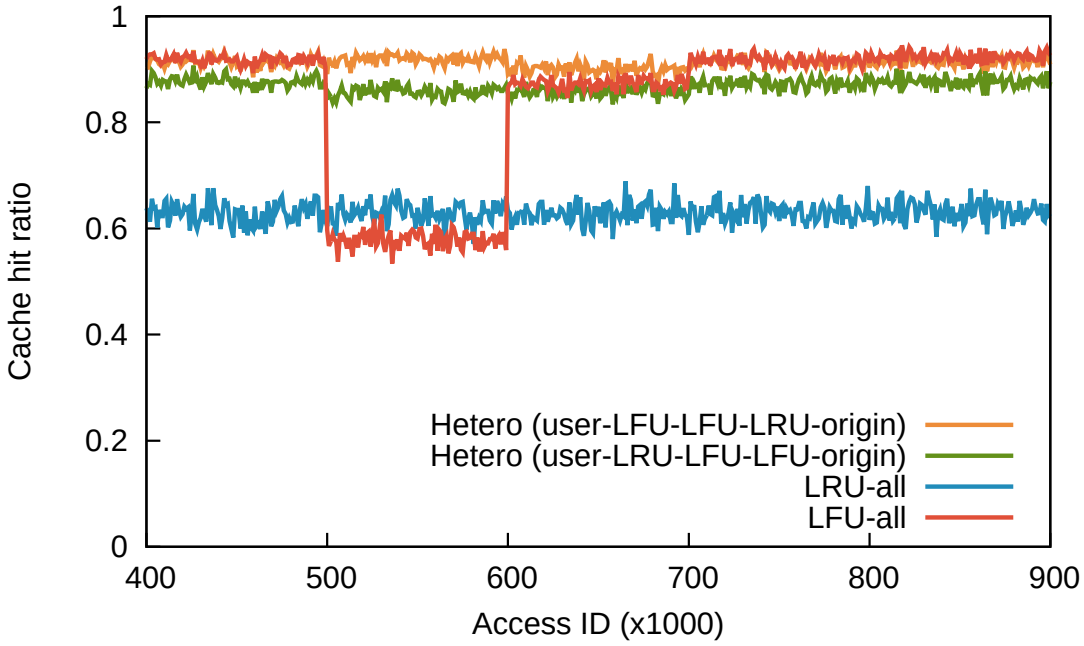


図 3.5: 3 段キャッシュ構成全体のヒット率

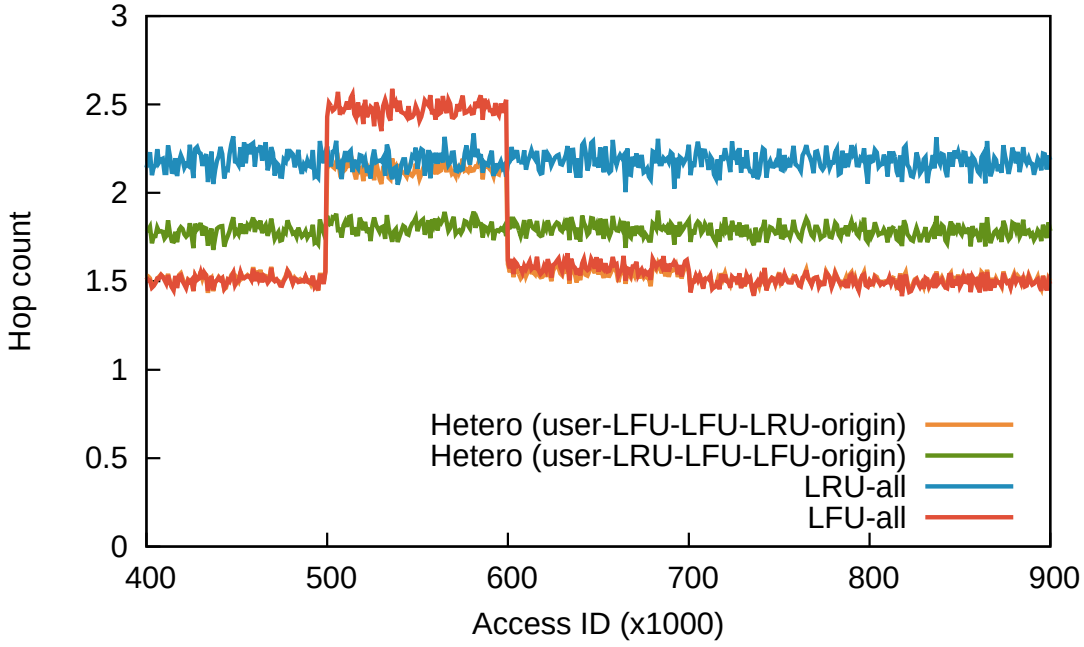


図 3.6: 3 段キャッシュ構成全体のホップ数

シュサーバほどアクセス頻度が高いコンテンツを保持しており，キャッシュサーバ間のコンテンツの重複は見られなかった．LFU アルゴリズムは過去のリクエストをもとに各コンテンツのアクセス頻度を算出し，アクセス頻度の高いものを保持するため，下段のキャッシュサーバはユーザからのリクエストをもとに上位 100 コンテンツを保持する．一方，下段のサーバでキャッシュヒットしたリクエストは中段のサーバに到達しないため，中段のサーバには 101-1000 位までのリクエストのみが到達し，そのうち上位 101-200 位のコンテンツが保持される．上段のサーバも同様に，下段と中段でキャッシュヒットしたリクエストを除く 201-1000 位までのリクエストが到達し，その中でアクセス頻度の高い 201-300 位までのコンテンツを保持する．その結果，各段のキャッシュサーバで保持するコンテンツが変わり，キャッシュネットワーク全体で上位 1-300 位までのコンテンツを保持することができるため，アクセス傾向が変化しない場合は LFU3 段構成のキャッシュヒット率が非常に高い．一方，新規に動画が追加されると，3 段のキャッシュネットワークで保持されている動画の順位がそれぞれ 20 位だけ下落するため，キャッシュされている動画は上位 2-32%となる．ガンマ分布では，これら動画はアクセス全体の 59.8%を占めるため，新規動画が追加された直後はヒット率が 58-60%程度に落ちている．その結果，ホップ数も急激に増加し，LFU が更新されるまでは LRU よりもホップ数が長くなるため，Web サーバとネットワークに大きな負荷をかけてしまう．アクセス変動後のヒット率とホップ数が階段状に改善しているのは，下段から順に正確な人気順位が LFU のキャッシュテーブルに適用されるためである．

LRU の 3 段構成では，新規動画の追加によるヒット率の変動がほとんどない．しかし，他の構成に比べてヒット率が低く，ホップ数も長いため，通信量削減効果が低い．これは，LRU がコンテンツの最終アクセス時刻に基づいて追い出し制御を行うことが原因である．すなわち，LRU アルゴリズムでは正確な動画のアクセス頻度を考慮しないため，人気下位の動画がアクセスされた際に，人気上位の動画を追い出してヒット率を落としてしまう．一方，このような制御ではキャッシュ領域で保持されるコンテンツが過去のアクセス情報に影響されにくいため，アクセスの変動に強く，ヒット率・ホップ数ともにほぼ一定の値を推移している．

また，キャッシュネットワーク中で異なるキャッシュアルゴリズムを混在させると，LRU を配置する位置によってヒット率とホップ数が変化した．LRU を上段にすると，変動時のヒット率が高い一方で，平均ホップ数が増大する．これは，新規動画が下段・中段を通過

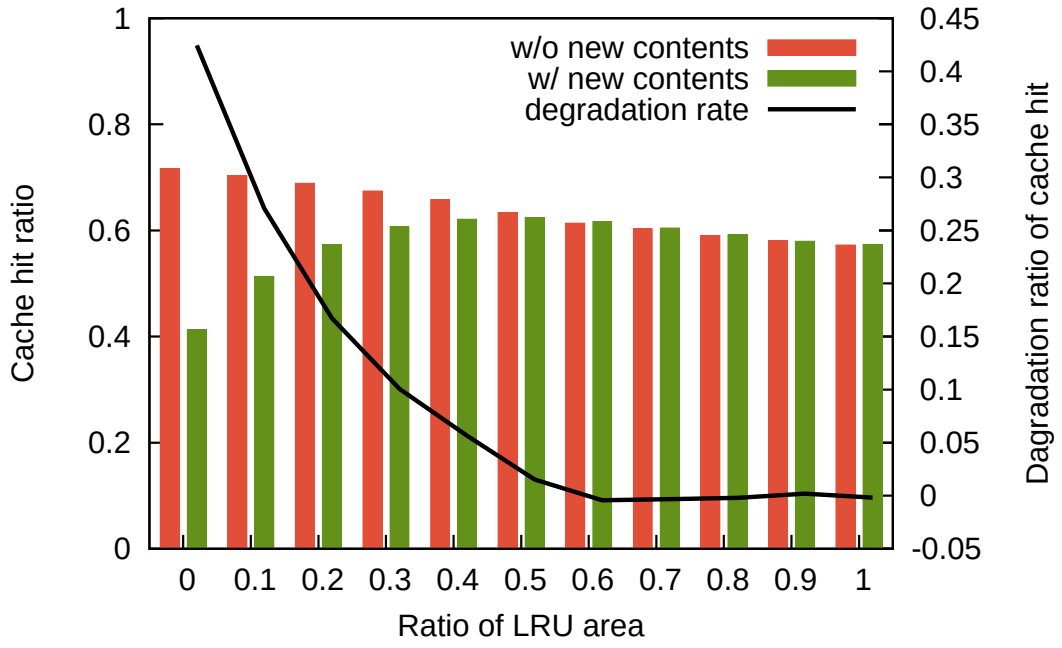


図 3.7: LRU の割合によるヒット率の変化

して上段でヒットするため，変動時には下位 2 段のホップ数が余計にかかってしまうためである．一方，LRU を下段にすると，ヒット率が 5%程度落ちるが，変動時も安定して短いホップ数を維持している．ヒット率が落ちるのは，下段の LRU が中段へ通過するアクセス傾向を変化させるために，LFU で正確な人気順位が算出できなくなることが原因である．また，変動時のホップ数が安定するのは，新規動画がすべて下段でキャッシュされ，中段以上へとアクセスが通過する確率が低くなるためである．これらのことから，下段を LRU にしたヘテロ構成とすることで，高いヒット率と短いホップ数をバランスよく達成できる．

3.4.3 単一ハイブリッドキャッシュの割合と通信量

ハイブリッド方式は，まず，キャッシュサーバ 1 台構成で評価した．LRU と LFU の割合とヒット率の関係を図 3.7 に示す．なお，キャッシュサーバ 1 台構成では，ホップ数のグラフは縦軸を逆転させた形状となる．

左側の棒グラフは LFU 領域が多く，右側は LRU 領域が大きい．人気変動がないときは，LRU 領域の拡大により，人気動画を確実にキャッシュする LFU 領域が減少して効率が悪

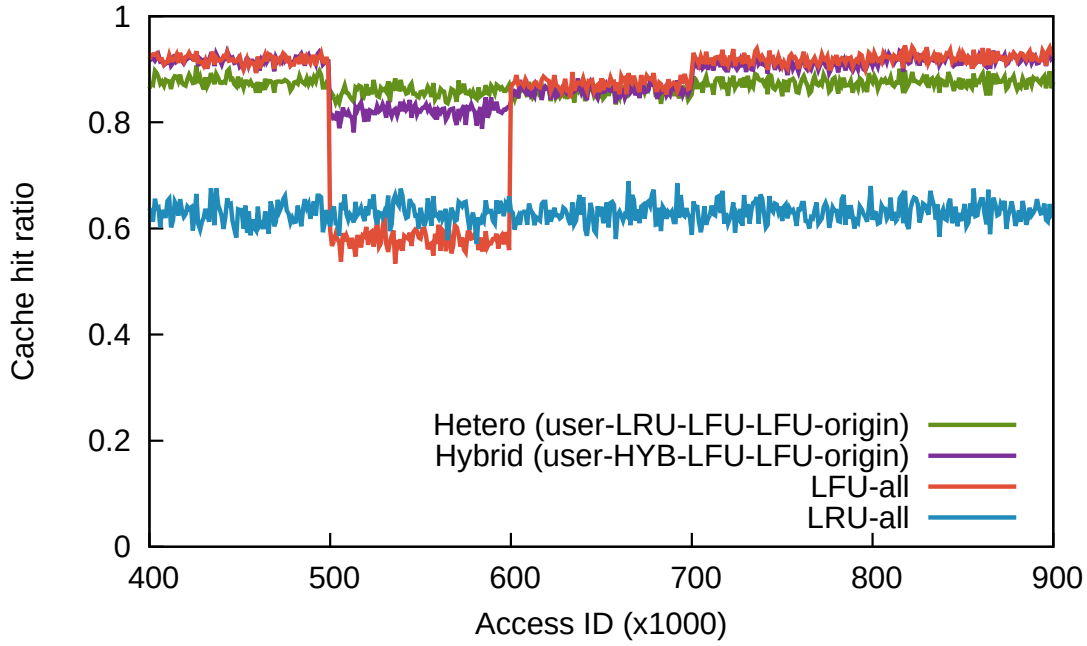


図 3.8: ハイブリッド構成と 3 段構成のヒット率の比較

化するため，ヒット率が緩やかに減少する．一方で，LRU 領域を少量追加すると，人気変動時のヒット率が大きく向上する．LFU 100%の構成では人気変動時のヒット率が 42%程度下落していたのに対し，キャッシュ領域全体の 30%を LRU 領域とすれば，変動時のヒット率の下落幅は 10%程度に抑えられる．LRU 領域と LFU 領域の割合は，要求するヒット率と許容する下落幅にしたがって，事前に設定すればよい．

3.4.4 階層ネットワークにおけるハイブリッドキャッシュの評価

ヘテロ方式の評価で，LRU は下段に置くとヒット率とホップ数をバランスよく改善できることが明らかになった．そこで，少量の LRU 領域を含むハイブリッド方式のキャッシュを下段に設定し，中段と上段に LFU キャッシュを並べた構成でヒット率とホップ数を評価した．LRU と LFU のキャッシュ混合割合は，LRU 30%，LFU 70%と設定した．ヒット率とホップ数の評価結果を図 3.8，3.9 にそれぞれ示す．

ハイブリッドキャッシュ1 台構成の実験では，新規動画の追加によりヒット率が 10%程度低下していた．LRU 領域はキャッシュ容量の 30%であるため，動画を 30 個分保持できる．一方，途中で追加される新規動画数は 20 個であるため，この LRU 領域は新規動画を

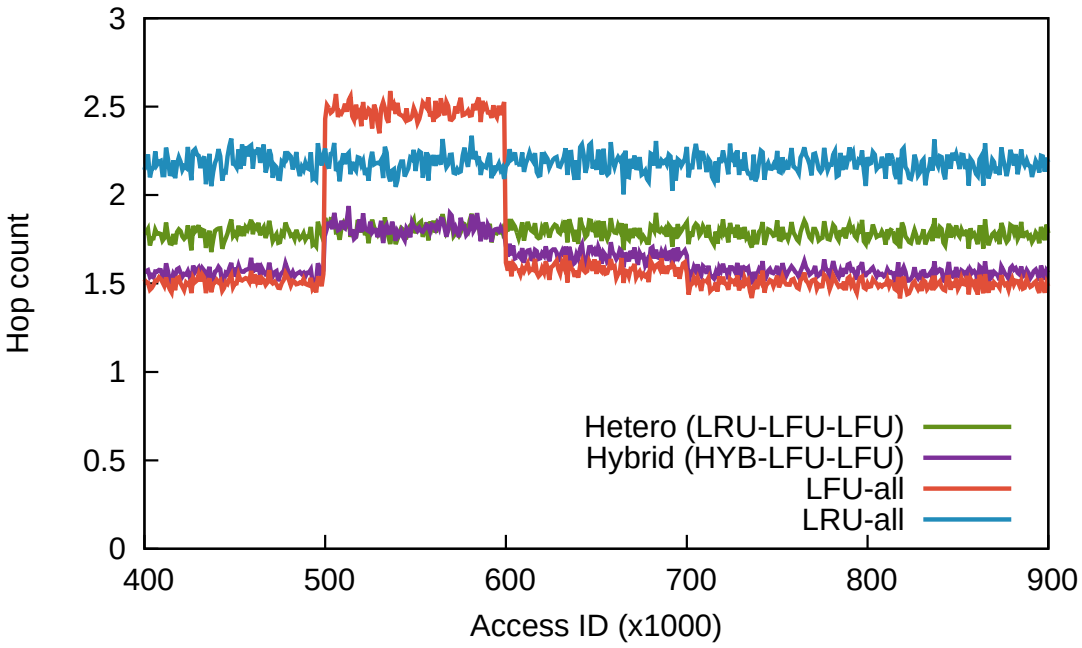


図 3.9: ハイブリッド構成と 3 段構成のホップ数の比較

すべてキャッシュできる容量である．しかし，ユーザからのアクセスは必ずしも人気上位のコンテンツに限定されない．そのため，人気下位のアクセスも発生すると，新規に追加された人気動画が LRU 領域から追い出されることがある．その結果，新規動画の追加によりヒット率が下がり，ホップ数が大きくなってしまう．一方，アクセス変動前のホップ数は LFU キャッシュ3 段構成のホップ数に近く，アクセス変動後もヘテロ方式と同等のホップ数を維持している．すなわち，ハイブリッドキャッシュの LRU と LFU の混合割合を適切に設定することで，人気変動がある場合・ない場合で共に良好なヒット率とホップ数を実現できる．

3.5 ハイブリッドキャッシュの分散協調方法の検討

階層ネットワークで LFU アルゴリズムを設置して高い通信量削減効果を実現できるのは，複数段に並べられたキャッシュサーバがそれぞれ異なるコンテンツを保持し，実効キャッシュ容量を拡大したためである．3 章では直列構成のキャッシュネットワークを対象にヒット率とホップ数の評価を実施したが，並列に設置されたキャッシュネットワークにおいて

も，各キャッシュサーバで異なるコンテンツを保持させることで，高い通信量削減効果を達成できることが示されている [43].

そこで，ハイブリッドキャッシュの LFU 領域で異なるコンテンツを保持させることで実効キャッシュ領域を拡大し，急激に変動する動画アクセスを LRU 領域で保持させることで，ハイブリッドキャッシュアルゴリズムと分散協調キャッシュ制御を組み合わせて利用できる．4 章では，軽量で効率的な分散協調キャッシュアルゴリズムを提案し，ハイブリッドキャッシュと組み合わせた構成での評価を通し，分散協調キャッシュにおけるハイブリッドキャッシュ利用の有用性を示す．

3.6 まとめ

LRU と LFU の 2 種類のキャッシュアルゴリズムを組み合わせ，急激な動画の人気変動に追従する階層キャッシュネットワークを検討し，ヒット率とホップ数を評価した．3 台のキャッシュサーバを直列に並べたキャッシュネットワーク構成で評価を行い，ヒット率の高い大容量 LFU と，人気変動に強い少量の LRU を混在させることで，急激なアクセス変動時も高いヒット率と短いホップ数を維持できることを確認した．また，1 台のキャッシュサーバのキャッシュ領域を LRU と LFU の 2 つに分割して制御するハイブリッドキャッシュアルゴリズムのヒット率を評価し，アクセス変動の大きさに応じて分割割合を設定することで，良好なヒット率とホップ数を達成できることを示した．また，複数キャッシュサーバのハイブリッドキャッシュ領域における LFU 領域で，それぞれ異なるコンテンツを保持させることで，実効キャッシュ容量を拡大して高い通信量削減効果を実現しつつ，アクセス変動に追従できる可能性を検討した．4 章では，3 章で提案したハイブリッドキャッシュアルゴリズムの LFU 領域を分散協調キャッシュ領域として制御することで，アクセス傾向の変化に追従しつつ，実効キャッシュ容量の拡大し，通信量を効率よく削減できる制御方法を提案する．

第4章　色タグ情報に基づく軽量分散協調キャッシュ制御

4.1　準最適なキャッシュ配置の考察

4 章では，効率よく通信量を削減する方策を検討するため，まず，理論上最小となる通信量を既存の最適化アルゴリズムで求める．その後，求めたキャッシュ配置の特徴をもとに規則性を見出し，簡単な規則で通信量削減効果の高いキャッシュ配置を求める方法を検討し，通信量削減効果を維持しつつ低コストな計算でキャッシュ配置を求める．Li[4] らは，通信量を最小化するキャッシュ配置を求めるため，キャッシュ配置問題を要領制約つき施設配置問題に変換し，遺伝的アルゴリズムで通信量を小さくするキャッシュ配置を求めている．彼らの研究では，通信帯域とレイテンシをネットワーク制約として最適化問題に導入しているが，計算コストが大きく，プログラムが複雑化してしまうため，帯域とレイテンシは制約外として通信量が小さくなるキャッシュ配置を計算した．

キャッシュ配置計算時のネットワークトポロジを図 4.1 に，パラメータを表 4.1 に示す．ネットワークは日本の NTT のバックボーンネットワークを模した構造をしており，全部で 55 ノードのキャッシュサーバで構成されている．各キャッシュサーバのキャッシュ容量は動画ライブラリ全体の 10%とし，動画リクエストはガンマ分布に従って生成し，4 章の評価と同じパラメータを使用して生成した．キャッシュサーバのストレージ容量は，大手 VoD サービス事業者の Netflix が各所に配置しているキャッシュサーバの容量と合わせた．また，動画リクエストは YouTube のアクセスを解析した論文 [26] からパラメータを流用した．ガンマ分布は，よく知られている Zipf や Weibul 分布と比較して，実際の動画のリクエストにより近いリクエスト生成モデルとして利用できる．キャッシュの最適配置計算は表 4.10 に示すスペックの PC1 台を使用し，ネットワーク帯域および取得遅延を考慮しないようにした以外は，Li らの方式と同じ方法を利用した．

図 4.2 に，遺伝的アルゴリズムの世代と，通信量の推移を示す．また，図 4.3 に，0 世

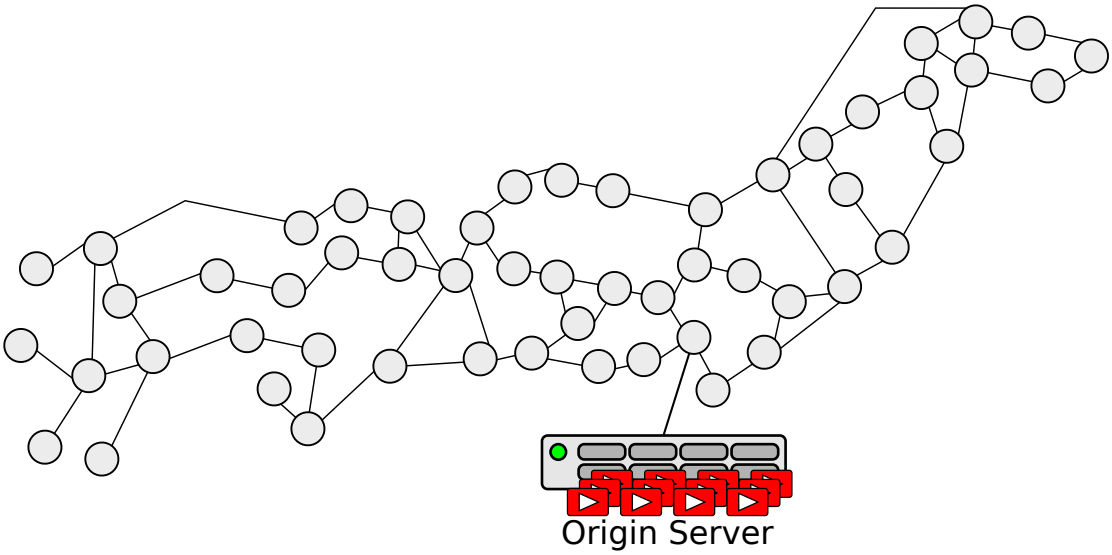


図 4.1: キャッシュ配置計算に使用したネットワークトポロジ

表 4.1: GA の計算に使用したパラメータ

Total content	1000
Cache capacity	100
Popularity distribution	Gamma distribution
Gamma parameter k	0.475
Gamma parameter θ	170.6067
Topology	NTT core network [44]
Number of servers	55

表 4.2: 計算ホストの構成

CPU	Intel Core i7-3930K @3.80GHz
CPU Core	6 physical cores / 12 logical cores
RAM	64GB

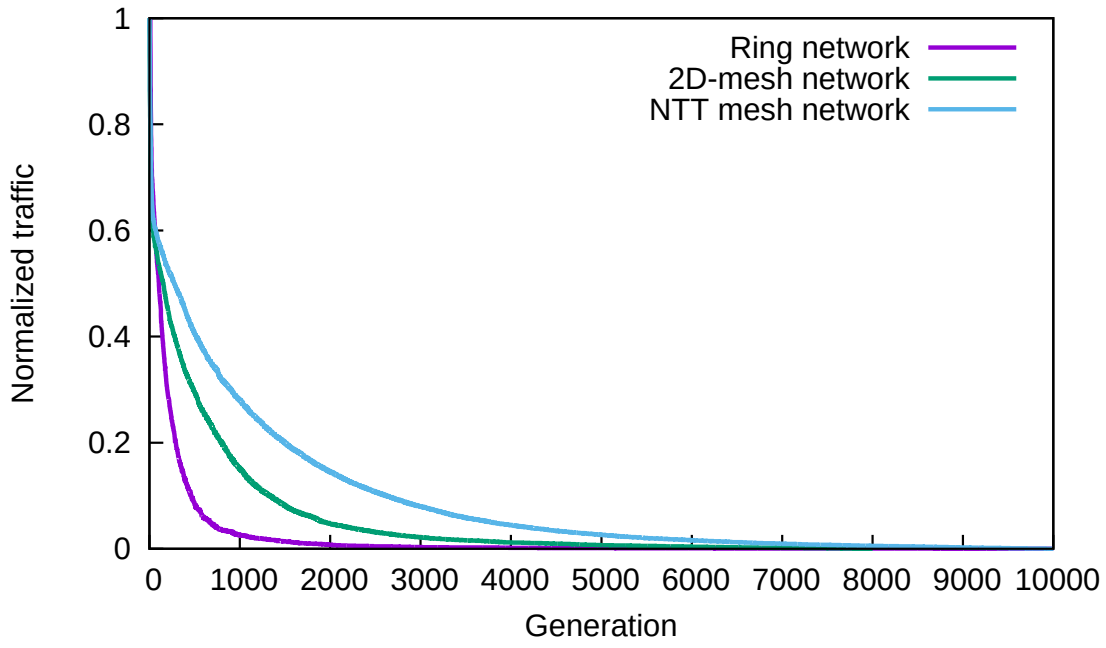


図 4.2: 遺伝的アルゴリズムの収束

代（初期状態），1000 世代，10000 世代目におけるキャッシュ配置から，ネットワーク中に存在するコンテンツ数を示す．通信削減量の改善幅は世代を重ねるにつれ小さくなっており，10000 世代まで計算すれば最適配置との差は大きくないと推測できる．ネットワーク中に保持されるコンテンツ数に着目すると，0 世代目はランダムに配置されたため，コンテンツが大きな偏りなく分布している．一方，1000 世代目，10000 世代目と計算が進むにつれ，上位コンテンツがネットワーク中で多く保持されるようになり，下位のコンテンツはキャッシュされなくなっている．特に，最上位コンテンツはネットワーク中のすべてのキャッシュサーバで保持されている．すなわち，上位コンテンツほどネットワーク中に多く保持し，そうでないコンテンツほど保持数が少なくなるようにキャッシュすれば，通信量削減効果を大きくできる．また，人気 100 位付近のコンテンツはネットワーク中の 20 箇所程度に保持されているが，ある地域に集中してキャッシュされていると，別の地域からのホップ数が長くなってしまうため，できるだけ分散して保持し，平均ホップ数を小さくすることが望ましい．

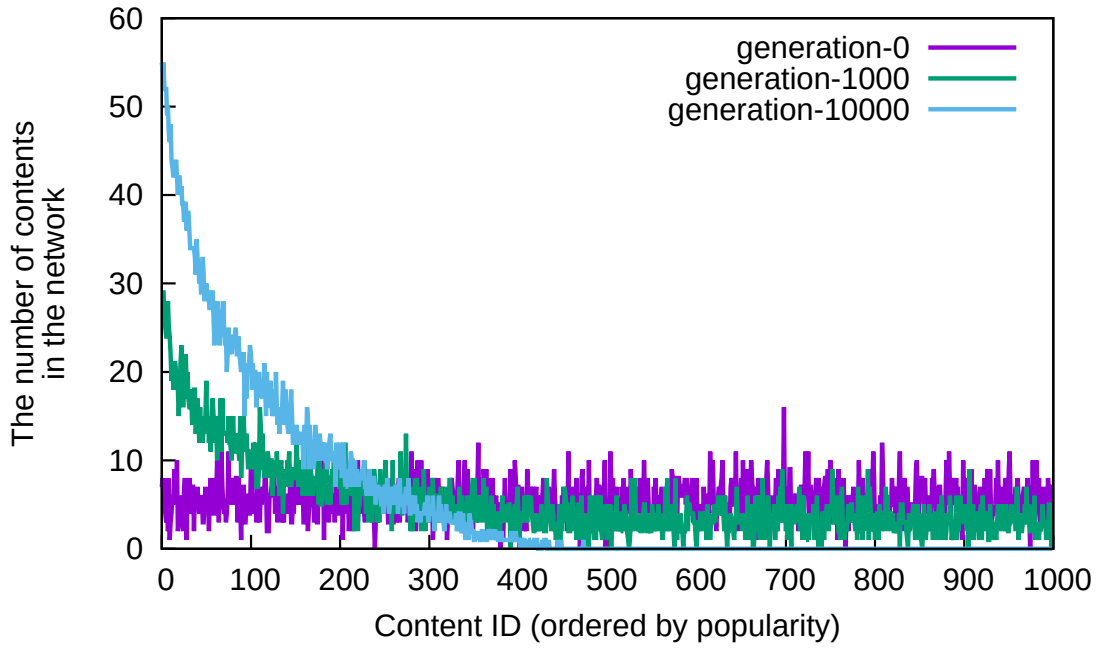


図 4.3: GA で計算した準最適キャッシュ配置における各コンテンツのネットワーク内配置数

4.2 色タグ情報に基づく分散協調キャッシュ

4.2.1 キャッシュ制御アルゴリズムの概要

コンテンツのアクセス頻度に応じてネットワーク中で保持するキャッシュサーバ数を決定するため，コンテンツを人気ごとにグループ化する．たとえば，コンテンツの人気を 4 つにグループ化すると，ネットワーク中で保持するキャッシュサーバ数は図 4.4 のようになる．このようにすると,「コンテンツをネットワーク中のキャッシュサーバ何ノードで保持するか」という問題を大幅に簡単化できる．コンテンツの人気順位でネットワーク中で保持する数を一位に定めることができるため，計算オーバヘッドを大幅に削減できる．

ネットワーク中で保持するコンテンツ数がもとまっても，どのキャッシュサーバで保持するかを決定しなければ，キャッシュ配置は求まらない．そこで，ネットワークをあらかじめ 4 色で塗り分けて，キャッシュサーバも同じくカテゴライズし，色のマッチングで各コンテンツのキャッシュ可否を決定する．ネットワーク彩色例を図 4.5 に示す．具体的には，図 4.6 に示すように，コンテンツとサーバに色情報を保持するタグを付与し，色がマッチする場合にキャッシュするよう制御する．人気上位のコンテンツはネットワーク中で多数

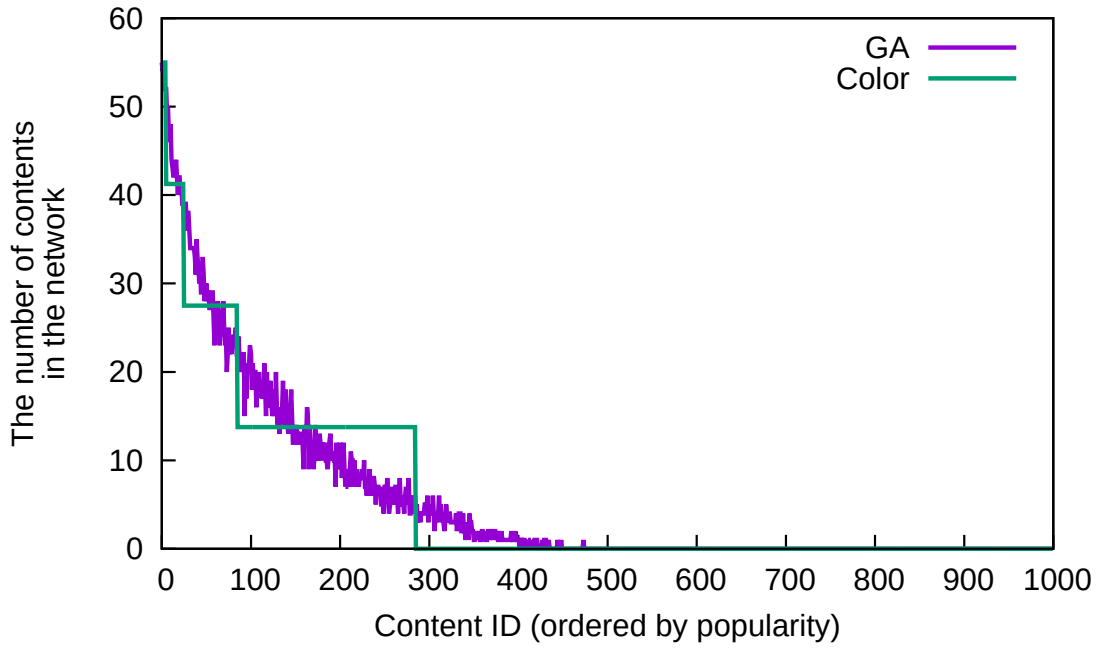


図 4.4: ネットワーク中に保持されるコンテンツ数

保持されるよう，色を多く設定することで，各キャッシュサーバのヒット率が向上し，ネットワーク内通信量の削減効果が向上する．また，人気下位のコンテンツは少数のサーバでキャッシュすればよいため，少ない色数を設定することで，各サーバで保持するコンテンツが変わって実効キャッシュ容量が拡大し，外部ネットワークとの通信量削減効果が向上する．このようにすると，人気上位のカテゴリにあるコンテンツは多数のサーバで，人気下位のカテゴリのコンテンツは少数のサーバでキャッシュするようにできるため，図 4.4 に示したようなコンテンツ数でキャッシュできる．

4 色定理によると，図 4.1 のようなネットワークでは，各ノードを同じ色が隣り合わない色で塗り分けるためには，4 色あれば十分である．同じ色が隣り合わなければ，隣接したサーバで同じコンテンツを保持しないため，ユーザからの平均ホップ数を小さくするように，効率よくコンテンツを分散保持できる．ネットワークを塗り分ける彩色パターンは無数にあるため，通信量削減効果が最大になるような塗り分けには長時間の計算が必要となる可能性がある．しかしながら，バックボーンネットワークのトポロジは頻繁に更新されないため，ネットワークの塗り分けに要する時間は隠蔽される．また，コンテンツのアクセス頻度は時間経過とともに変化するため，コンテンツに付与したタグを一定時間ごと

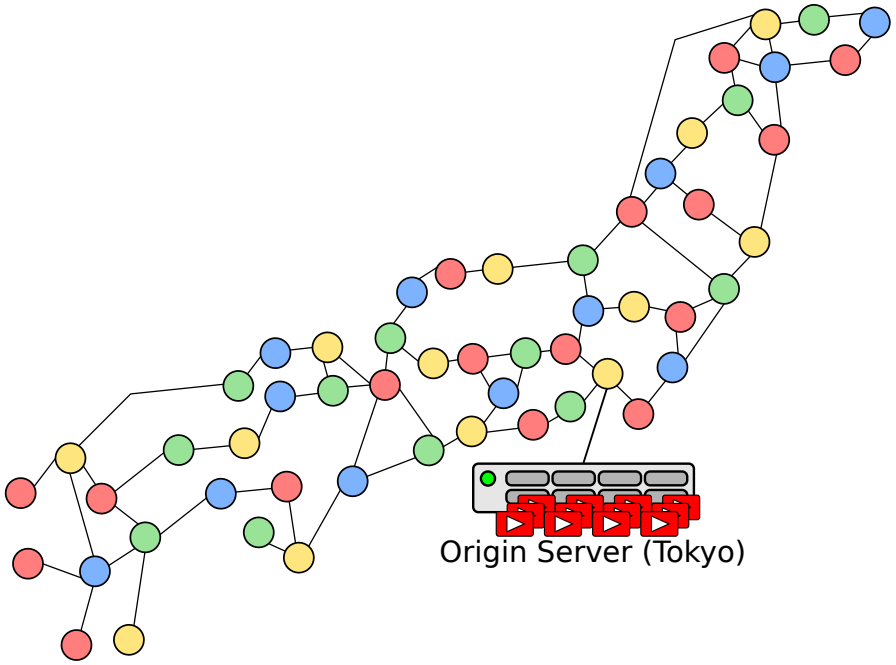


図 4.5: NTT コアネットワークを模したトポロジのサーバ彩色例

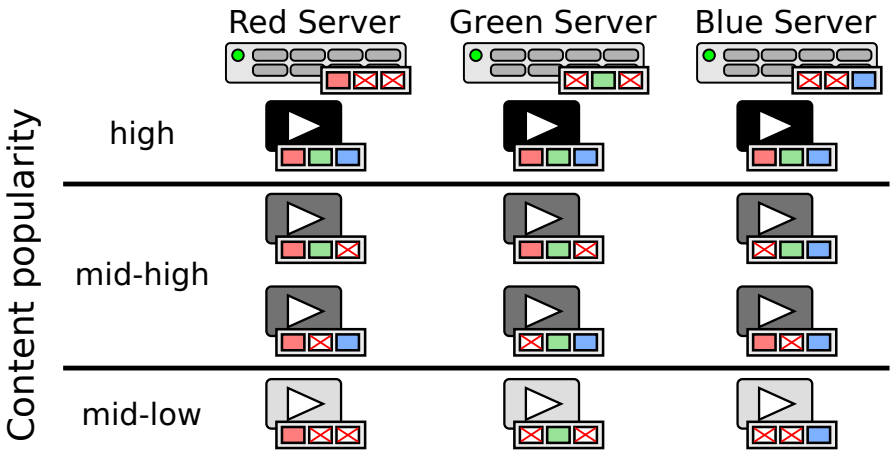


図 4.6: 色タグが付与されたコンテンツとサーバの対応

に更新することで，動画アクセス頻度の変化にも対応できる．

キャッシュサーバとコンテンツ彩色手順のフローチャートを図 4.7 に示す．まず，(1) 本節で述べたようにネットワーク中に配置されるキャッシュコンテンツ数を解析し，(2) 各ノードの色が隣り合わないようにネットワーク中のキャッシュサーバに色を設定する．その後，(3) 各キャッシュサーバからアクセスログを収集し，アクセスされたコンテンツのア

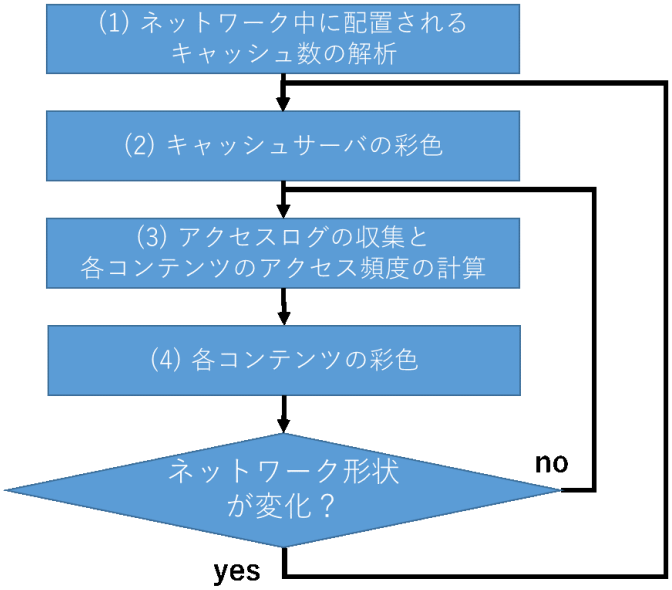


図 4.7: キャッシュサーバとコンテンツの彩色手順の概要

クセス頻度を計算し，(4) その情報をもとにコンテンツを彩色する．一定時間経過後，ネットワーク形状が変化していれば，(2) サーバ彩色からやり直し，ネットワーク形状に変化がなければ手順 (3) からやり直す．このようにすることで，アクセス傾向やネットワーク形状が変化してもキャッシュ制御方法を更新できる．

4.2.2 キャッシュサーバの分散彩色方法

ネットワーク中に存在するキャッシュサーバに付与される色数が一様に分布していれば，あるユーザからあるコンテンツまでの平均ホップ数の期待値を，コンテンツの人気クラスによって設定できる．たとえば，人気クラスが最上位のコンテンツはユーザ直上のキャッシュサーバに保持されており，そのキャッシュサーバと隣り合う 3 台のキャッシュサーバがすべて異なる色で彩色されていれば，人気クラスが低いコンテンツでも隣のキャッシュサーバに保持されているため，ユーザからコンテンツまでの平均ホップ数を小さくできる．

ネットワークを塗り分けるには，Welsh-Powell のアルゴリズム [45] が有名である．このアルゴリズムでは，次数の高いノードから順番に色を決定し，隣のノードと異なる色を順に付与することで，ネットワークノードを 4 色以内で彩色できる．また，このアルゴリズムは貪欲なため，短時間での彩色が可能である．しかし，ネットワーク形状によっては 2

Algorithm 1 Server colorization algorithm

Require: available colors $C = \{c_1 ... c_i\}$, Graph $G(V, E)$ where nodes $V = \{v_1 ... v_j\}$ and edges $E = \{e_1 ... e_k\}$

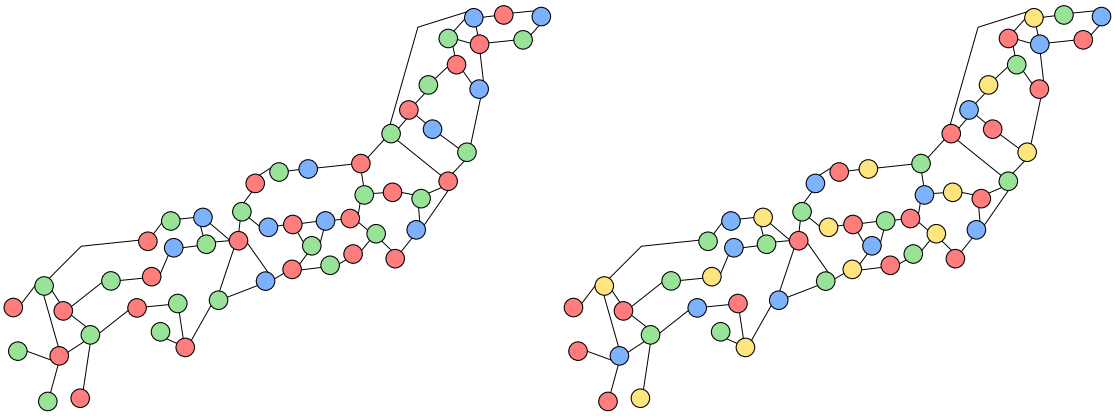
1: Initialization: sort V based on $degree(v_j)$ descendingly
2: **for** v in V **do**
3: $adjacent\_colors \leftarrow \emptyset$
4: **for** a in $adjacent\_nodes(v)$ **do**
5: $adjacent\_colors = adjacent\_colors \cup \{color(a)\}$
6: **end for**
7: $candidate\_colors \leftarrow C - adjacent\_colors$
8: sort $candidate\_colors$ based on the minimal distance to the same color descendingly
9: $color(v) = candidate\_colors[0]$
10: **end for**

色や 3 色でも着色可能なため（たとえば直線的なネットワーク），そのままのアルゴリズムでは 4 色に満たない色数で彩色されてしまう．

そこで，色間距離が遠くなる色を優先的に選択するようアルゴリズムを一部変更した．変更したアルゴリズムを Algorithm 1 に，彩色結果を図 4.8 に示す．Algorithm 1 は，オリジナルの Welsh-Powell アルゴリズムと同様に，次数の大きいノードから順に色を設定するが，隣り合わない色を選択（7 行目）した後で，色間距離が遠くなる順番でソート（8 行目）することで，グラフ全体に色が分散して彩色されるようになっている．NTT のバックボーンネットワークを模したネットワークを対象にノードを 4 色で彩色した結果，オリジナルの Welsh-Powell アルゴリズムでは 3 色しか利用されなかったのに対し（図 4.8 (a)），Algorithm 1 では 4 色利用され（図 4.8 (b)），ネットワーク中に色が分散していることが確認された．

4.2.3 アクセス頻度に基づくコンテンツ彩色手法

コンテンツごとにキャッシュすべきサーバを設定するためには長い計算時間を要するため，コンテンツをアクセス頻度をもとにいくつかの人気クラスに分類し，クラスごとに色情報を保持するタグを付与する．具体的には，サーバ彩色に使用した色数が N であるとき，コンテンツをアクセス頻度順に N+1 つのグループに分類する．k（1<k<=N）番目のグループのコンテンツは，全体の (N-k+1)/N のキャッシュサーバに保持すると考え，図 4.4 で求めたコンテンツ数となるべく近づくようにグループ内のコンテンツ数を設定する．例えば，キャッシュサーバ 1 台で 100 種類のコンテンツを保持できる場合，このようにグルー



(a) Original Welsh-Powell Algorithm　　(b) Select distant color with Algorithm 1

図 4.8: Welsh-Powell アルゴリズムと Algorithm 1 でネットワークを彩色した結果

プを設定すると，ネットワーク中で保持できるコンテンツは 300 種類程度まで増加する．すなわち，実効キャッシュ容量が増加し，より多くの通信要求をキャッシュから応答できるため，外部ネットワークとの通信量が削減される．

コンテンツをアクセス頻度順に N+1 つのグループに分割したら，表 4.3 のような表を作成し，アクセス頻度順にコンテンツを彩色する．この表は，各行がコンテンツに付与する N=4 ビットのタグ，各列が付与する色を示す．各タグは，タグに含まれる 1 のビット数で N+1 つのグループに分割される．例えば，アクセス頻度が最高のグループに含まれるコンテンツには，すべてのビットが 1 で構成される色タグを付与する．アクセス頻度が次に高いグループに含まれるコンテンツには，N-1 つのビットが 1 で構成される色タグを循環的に付与する．アクセス頻度順に循環的にタグを付与するだけなので，計算にかかる時間は非常に短い．たとえば，人気順位 35 番のコンテンツは人気クラス Middle に分類されるため，4 色中 2 色分の色が付与される．色タグは循環的に割り当てられるため，このコンテンツには “1001” のタグが付与される．

各コンテンツのアクセス頻度は一定時間ごとに再計算され，最新のアクセス頻度をもとにコンテンツのタグが更新される．具体的には，配信サーバまたは別途サーバを導入し，キャッシュサーバからアクセスログを収集し，一定期間内の各コンテンツのアクセス数を集計する．

4.2.4 コンテンツグループ分割の自動化方法

表 4.3 に示した人気グループは，アクセスの偏りによって異なる．コンテンツアクセスの偏りが大きく，多数のアクセスが少数のコンテンツに集中してる場合はどのキャッシュサーバでも人気コンテンツを多数重複して保持したほうが通信量削減効果が高い．一方で，コンテンツアクセスが分散していると，キャッシュサーバ間でコンテンツ重複をできるだけ排除し，実効キャッシュ容量の拡大を図ることが望ましい．このように，コンテンツアクセスが人気上位コンテンツにどれだけ偏っているかによって，通信量削減効果の高い人

表 4.3: 4 色で彩色する際のコンテンツの人気クラスと付与する色タグ

			Tag's bit patterns			
Popularity class	content popularity ranking	# of colors	R	G	B	Y
high	1-22	4	1	1	1	1
mid-high	23-26	3	1	1	1	0
			1	1	0	1
			1	0	1	1
			0	1	1	1
middle	27-47	2	1	1	0	0
			1	0	1	0
			1	0	0	1
			0	1	1	0
			0	1	0	1
			0	0	1	1
mid-low	48-305	1	1	0	0	0
			0	1	0	0
			0	0	1	0
			0	0	0	1
low	306-	0	0	0	0	0

気クラスの設定が異なる．

そこで，動画アクセスはガンマ分布に近い偏りを示すと仮定し，現在の動画アクセス傾向からガンマ分布の偏りパラメータ k を求め，適切な人気グループを設定したい．具体的には，各人気クラスの最下位（表 4.3 中では 22，26，47，305）をセパレータランクとして，適切なセパレータランクの組を設定することで，人気の偏りに応じたコンテンツ彩色を表現する．最終的には，表 4.4 に示すように，偏りパラメータ k とセパレータランクの組のセットを用意しておき，現在のアクセス傾向から偏りパラメータ k が分かると，最も近い k に対応する色分散パターンを指定できるようにする．このようにすると，動画アクセスの偏りが変化しても，適切なコンテンツ重複・分散度を設定できるため，通信量削減効果が維持できる．

コンテンツ重複度は，偏りパラメータ k ごとに，遺伝的アルゴリズムで準最適配置を計算しておき，目視で設定することもできるが，目視で設定したコンテンツ重複度が最適とは限らないだけでなく，遺伝的アルゴリズムの計算にも長時間を要する．そこで，コンテンツ重複度の各セパレータランクを順番に上下させるイテレーション計算を実施することで，計算時間の短縮を図る．具体的なアルゴリズムを Algorithm 2 に示す．たとえば，動画コンテンツが全部で 1000 種類あり，キャッシュサーバ 1 台で 100 種類の動画をキャッシュ

表 4.4: ガンマパラメータごとに用意されたセパレータランク（4 色の場合）

bias parameter k	separator ranks
0.1	[38, 43, 58, 265]
0.2	[34, 39, 54, 277]
0.3	[30, 34, 55, 285]
0.4	[24, 28, 49, 303]
0.5	[21, 22, 43, 318]
0.6	[15, 15, 42, 332]
0.7	[10, 11, 32, 351]
0.8	[4, 5, 26, 369]
0.9	[0, 0, 10, 392]

できる状況を想定する．まず，セパレータランクを [0, 0, 0, 400] に設定し，人気上位 400 種類の動画コンテンツに 1 色ずつ色を付与し，通信量削減効果を推定する．ここから，1 つ目のセパレータランクを 0 から 100 まで推移させ，通信量削減効果が最大となる値を見つける．1 つ目のセパレータランクを 0 から 100 まで推移させると，通信量は図 4.9 に示すように推移する（Algorithm 2 中の 6–14 行目に対応）．図 4.9 の縦軸に示す通信量は，Algorithm 2 中の T_{est} に対応する．このグラフを見ると，1 つ目のセパレータランクを 30 程度に設定すると，通信量を最小化できると読み取れる．1 つ目のセパレータランクが求まったら，2 つ目のセパレータランクで同じように計算し，通信量が最小化される値に設定する．ここで注意すべきは，2 つ目のセパレータランクが決定されると，1 つ目のセパレータランクを設定した時とはコンテンツの分散重複の仕方が変わってしまうため，1 つ目のセパレータランクの再計算が必要ということである．そのため，各セパレータランクを 1 度計算したら，各セパレータランクを再計算するイテレーションを繰り返し，通信量が下がらなくなったセパレータランクの組を，偏りパラメータ k に対応するセパレータランクの組として扱い，表 4.3 に登録する．なお，セパレータランクは事前に計算しておくことができ，ネットワーク形状とキャッシュ容量が変化しなければ再利用できる．コアネットワークの形状は頻繁に変更されず，キャッシュサーバも一定期間同じものを利用するため，計算に多少時間がかかっても大きな問題は生じない．

表 4.3 を事前に求めておけば，オリジンサーバは軽量な計算でセパレータを決定できる．具体的には，一定期間のアクセスログを集計し，各コンテンツのアクセス頻度を計算し，最小二乗法である偏りパラメータ k でプロットしたガンマ分布との二乗誤差を計算し，誤差が最小となる k を見つける．ここで，横軸に k，縦軸に二乗誤差をとったグラフをプロットすると，下に凸の単峰性グラフができる．このようなグラフでは，黄金分割探索アルゴリズムを使用することで，軽量な計算で極値を求められることが知られている [46].

4.2.5 階層ネットワーク対応の検討

現実のネットワークはいくつかの階層に分かれているため，複数の階層にまたがってキャッシュサーバを配置し，協調動作を可能とすることで，さらなる通信量削減効果の向上を期待できる．特に，今後動画が高精細化してファイルサイズが増大すると，キャッシュサーバ 1 台あたりに要求されるデータ転送帯域も大きくなるため，より下層のネットワー

Algorithm 2 Iterative calculation for separator ranks

```
1: Initialization: S ← {0, 0, ..., 0}, S_prev ← {0, 0, ..., 0}, T_min ← ∞
2: S[N − 1] ← N × C
3: while S ≠ S_prev do
4:     S_prev ← S
5:     for i ← 0 to N − 2 do
6:         for v ← Max{0, S[i − 1]} to Min{S[i + 1], N × C} do
7:             S_tmp ← S
8:             S_tmp[i] ← v
9:             T_est ← estimate_traffic(S_tmp)
10:            if T_est < T_min then
11:                T_min ← T_est
12:                S ← S_tmp
13:            end if
14:        end for
15:    end for
16: end while
17: return S
```

S, S_{prev}, S_{best}, S_{tmp}	array of separator ranks, with N elements
T_{min}	minimum traffic in the whole calculation
N	the number of colors
C	the number of contents a cache server can store
T_{est}	traffic size estimated by estimate_traffic()

クにキャッシュサーバを多数配置することで，キャッシュサーバ 1 台あたりに求められる転送性能を抑えるとともに，各階層のキャッシュサーバで異なるコンテンツを保持することで，実効キャッシュ容量を拡大し，さらなる通信量の削減を見込める．

図 2.1 に示したインターネット構造の最下層にある ISP ネットワークは，図 4.10 のように階層構造になっている．単一階層の（特に上流の）ネットワークにキャッシュサーバを配置するだけでは，前述のように通信帯域の確保が困難であったり，十分なキャッシュ容量を確保できない可能性がある．一般に，キャッシュサーバは下層のネットワークに配置すると通信量削減効果を大きくできるが，下層のネットワークには強力なサーバを配置しにくく，キャッシュ容量も不足しがちになる．たとえば，データセンタ拠点にキャッシュサーバを多数設置すると十分なキャッシュ容量が確保できるが，ユーザからデータセンタまでのネットワークに大きな負荷がかかる．一方，各家庭のルータや回線集約機等にキャッシュを配置すれば通信量を効率よく削減できるが，キャッシュ1 台あたりの容量が小さく，通信量削減効果が限定的になってしまう．そこで，各階層にキャッシュネットワークを構

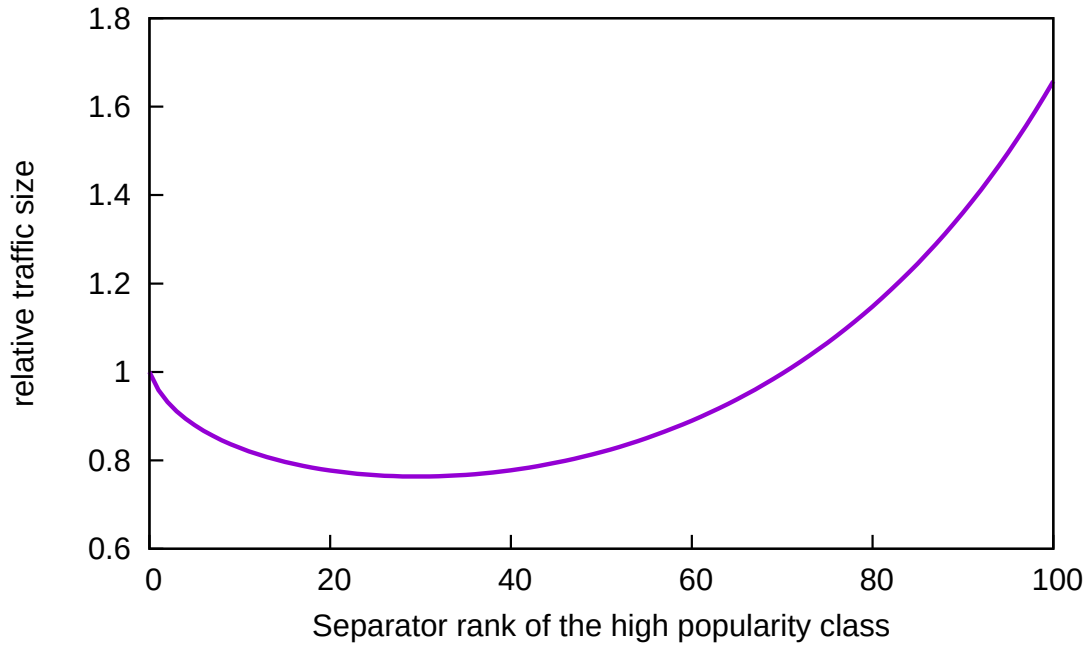


図 4.9: グループ分割位置と通信量の推移

築し，効率よく協調動作させるための仕組みを検討する．

具体的には，図 4.7 に示すコンテンツ彩色方法のうち「(4) 各コンテンツの彩色」を拡張し，階層ネットワークで同技術を利用する．具体的には，各コンテンツのアクセス確率によってコンテンツをどの階層に配置するかを事前に割り当てておき，階層ごとに Algorithm 1-3 の彩色方法を適用する．その際，コンテンツのタグはグループごとに循環的に付するだけで済むように，各階層の色タグを統合して扱えるよう工夫する（図 4.10）．その結果，階層ネットワークでも既存の方法と同じ手順でコンテンツが彩色でき，各階層のキャッシュネットワークを組み合わせて利用し，通信量削減効果を向上できる．

サーバの彩色方法

階層ネットワークで Algorithm 1 と同じ方法でサーバを彩色すると，異なる階層のキャッシュサーバに同じ色が付されることがある．その結果，階層間でキャッシュされるコンテンツが重複するため，キャッシュ容量の利用が非効率的である．そのため，各階層で利用する色を異なるものに設定し，階層ごとに Algorithm 1 を利用してサーバを彩色し，階層間での色の重複を排除してキャッシュ領域を効率的に利用し，通信量削減効果の向上を図

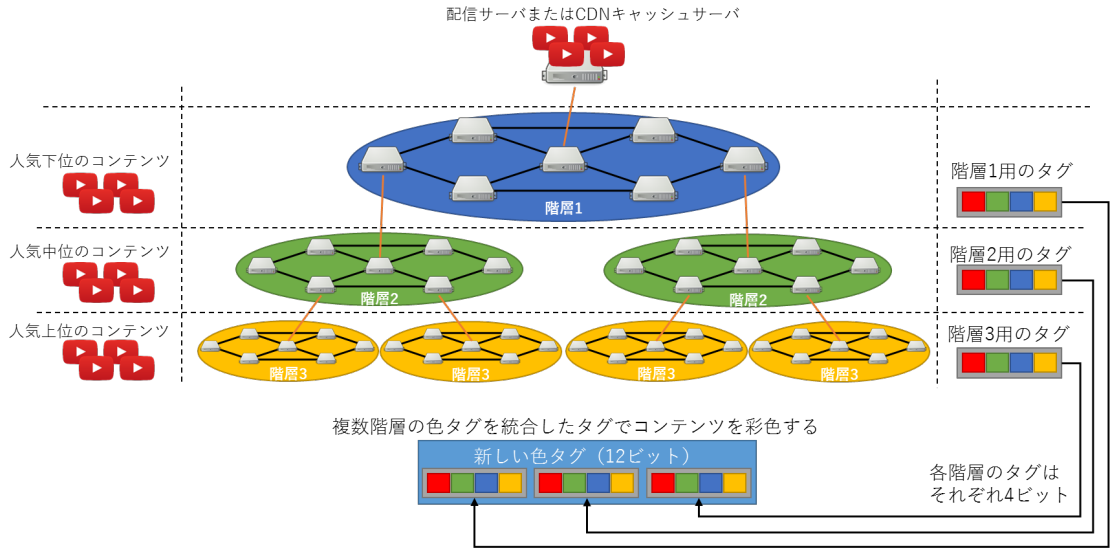


図 4.10: 3 階層ネットワークを例にした階層統合型の色タグ構成

る． 具体的には，N 階層のネットワークを，それぞれ C1，C2，…CN 色で彩色する場合，使用するタグのビット数 $M = \sum_{i=1}^{N} C_i$ で設定する．彩色対象の階層番号が L のとき，最左 $\sum_{i=1}^{L-1} C_i$ ビットと最右 $\sum_{i=L+1}^{N} Ci$ ビットは 0 で埋められる．たとえば，3 階層のネットワークをそれぞれ 4 色で彩色するとき，第 2 階層のキャッシュサーバに付与する色タグは次の 4 通りのビットベクトルを組み合わせて論理和を計算した値である．

- 0000 1000 0000
- 0000 0100 0000
- 0000 0010 0000
- 0000 0001 0000

コンテンツの階層割り当てと階層統合型の色タグの生成方法

階層ネットワークにおいて，通信量削減効果を最大化するためには，アクセスされやすいコンテンツを下層に，そうでないコンテンツを上層に配置するとよい．そこで，まず，各コンテンツのアクセスログを収集し，コンテンツのアクセス確率を算出し，アクセス確率順にソートしておく．その後，アクセス確率の高いコンテンツほど下層になるよう，コ

ンテンツの人気順位に階層 ID を割り当て，対応する色タグを付与する．具体的には，表 2 に示すような表を作成し，人気順位に対応する色タグを循環的に付与する．表 4.5 は，図 4.10 に示すような 3 階層のネットワークにおいて，各階層で 4 色利用することを想定して作成したものである．最左列の，人気順位いくつまでのコンテンツに何色割り当てるかは，アクセスログから計算されたアクセスの偏りや，各キャッシュサーバの容量をもとに事前に与えられる．色タグは，各階層のタグを統合して生成された 12 ビットのビットベクトルで，対応する階層以外のビットはすべて 0 で埋められている．たとえば，表 4.5 中の人気順位 801 1200 位のコンテンツ（階層 2 に対応）は，以下 6 種類のタグを循環的に付与する．

- 0000 1100 0000
- 0000 1010 0000
- 0000 1001 0000
- 0000 0110 0000
- 0000 0101 0000
- 0000 0011 0000

このようにすると，最左の 4 ビット（階層 1 に対応）と，最右の 4 ビット（階層 3 に対応）には 1 のビットがないため，階層 1 と 3 のキャッシュサーバは自分の階層でキャッシュ不要と判断できる．キャッシュサーバは自分の階層に対応するビットだけを確認すれば，自分の階層内にキャッシュがあるかどうかわかるため，ユーザからのリクエスト転送時のオーバヘッドも小さく抑えられる．

コンテンツのタグ付け方法

コンテンツのタグ付けは，表 4.5 をもとに，人気最上位コンテンツから順に行う．具体的には，<開始順位, 終了順位, [色タグのリスト]>のタプルを用意しておき，開始順位から終了順位までのコンテンツに，リスト中のタグを順番に付与していき，リストの最後のエントリになったら，リスト先頭のタグに戻って繰り返しタグを割り当てる．このタグ付け

表 4.5: 3 階層ネットワークを例にした階層統合型の色タグ構成

人気順位	階層	色数	色タグ											
			R1	G1	B1	Y1	R2	G2	B2	Y2	R3	G3	B3	Y3
～ 20	3	4	0	0	0	0	0	0	0	0	1	1	1	1
～ 100		3	0	0	0	0	0	0	0	0	1	1	1	0
			0	0	0	0	0	0	0	0	1	1	0	1
			0	0	0	0	0	0	0	0	1	0	1	1
			0	0	0	0	0	0	0	0	0	1	1	1
～ 250		2	0	0	0	0	0	0	0	0	1	1	0	0
			0	0	0	0	0	0	0	0	1	0	1	0
			0	0	0	0	0	0	0	0	1	0	0	1
			0	0	0	0	0	0	0	0	0	1	1	0
			0	0	0	0	0	0	0	0	0	1	0	1
			0	0	0	0	0	0	0	0	0	0	1	1
～ 500		1	0	0	0	0	0	0	0	0	1	0	0	0
			0	0	0	0	0	0	0	0	0	1	0	0
			0	0	0	0	0	0	0	0	0	0	1	0
			0	0	0	0	0	0	0	0	0	0	0	1
～ 550	2	4	0	0	0	0	1	1	1	1	0	0	0	0
～ 800		3	0	0	0	0	1	1	1	0	0	0	0	0
			0	0	0	0	1	1	0	1	0	0	0	0
			0	0	0	0	1	0	1	1	0	0	0	0
			0	0	0	0	0	1	1	1	0	0	0	0
～ 1200		2	0	0	0	0	1	1	0	0	0	0	0	0
			0	0	0	0	1	0	1	0	0	0	0	0
			0	0	0	0	1	0	0	1	0	0	0	0
			0	0	0	0	0	1	1	0	0	0	0	0
			0	0	0	0	0	1	0	1	0	0	0	0
			0	0	0	0	0	0	1	1	0	0	0	0
～ 1800		1	0	0	0	0	1	0	0	0	0	0	0	0
			0	0	0	0	0	1	0	0	0	0	0	0
			0	0	0	0	0	0	1	0	0	0	0	0
			0	0	0	0	0	0	0	1	0	0	0	0
～ 2000	1	4	1	1	1	1	0	0	0	0	0	0	0	0
～ 2800		3	1	1	1	0	0	0	0	0	0	0	0	0
			1	1	0	1	0	0	0	0	0	0	0	0
			1	0	1	1	0	0	0	0	0	0	0	0
			0	1	1	1	0	0	0	0	0	0	0	0
～ 3600		2	1	1	0	0	0	0	0	0	0	0	0	0
			1	0	1	0	0	0	0	0	0	0	0	0
			1	0	0	1	0	0	0	0	0	0	0	0
			0	1	1	0	0	0	0	0	0	0	0	0
			0	1	0	1	0	0	0	0	0	0	0	0
			0	0	1	1	0	0	0	0	0	0	0	0
～ 4500		1	1	0	0	0	0	0	0	0	0	0	0	0
			0	1	0	0	0	0	0	0	0	0	0	0
			0	0	1	0	0	0	0	0	0	0	0	0
			0	0	0	1	0	0	0	0	0	0	0	0
4501 ～	割り当てなし	0	0	0	0	0	0	0	0	0	0	0	0	0

方法は，第 4 章で提示される方法と同じである．すなわち，上記で示したように階層統合型のタグを利用すれば，コンテンツのタグ付け方法は変更不要である．

4.2.6 分散協調可能なハイブリッドキャッシュ制御

色タグベースのキャッシュ制御単体ではコンテンツに付与する色タグの更新が一定時間ごとに行われるため，更新直後に急激にアクセス傾向が変化するとヒット率が下落してしまう．そこで，提案する色タグベースの分散協調キャッシュでは，異なる種類のキャッシュアルゴリズムを組み合わせたハイブリッドキャッシュを採用する．図 4.11 に提案するハイブリッドキャッシュの構造を，ハイブリッドキャッシュの制御アルゴリズムを Algorithm 4 にそれぞれ示す．各キャッシュサーバはそれぞれのストレージ領域を大容量の色タグ制御 LFU 領域と，小容量の Modified LRU[10] 領域に分割する．既存のハイブリッドキャッシュと異なり，各キャッシュサーバはコンテンツに付与された色タグを最初に確認し，サーバとコンテンツの色タグがマッチした場合にキャッシュすることで（8–16 行目），色タグベース LFU 領域は分散協調制御され，ネットワーク全体の通信量削減に貢献する．一方，Modified LRU 領域は，色タグがマッチしなかったアクセス頻度の高いコンテンツをキャッシュする領域で，急激にアクセス傾向が変化してコンテンツに付与された色タグの更新が追いつかない場合でもキャッシュが可能である．このようにすることで，色タグベースの LFU 領域では効率よく分散協調キャッシュを構成し，Modified LRU 領域では急激なアクセス傾向の変化に追従できる．さらに，Modified LRU 領域は，色がマッチしなかったアクセス頻度の高いコンテンツを保持することで，サーバ単体でもヒット率向上効果がある．また，純粋な LRU アルゴリズムではなく，よりヒット率の高い Modified LRU アルゴリズムを採用することで，アクセス傾向の変化に追従しつつ，ヒット率を改善することができる．

4.2.7 色タグ情報に基づく経路制御手法

ユーザからキャッシュサーバまでのリクエスト転送

ユーザからのコンテンツ要求は，多数存在するキャッシュサーバの中から 1 台を選択し，最も近いキャッシュサーバに転送してホップ数の短縮を図ることが望ましい．URL は CDN のホスト名，要求するコンテンツのパス，リクエストパラメータで構成されており，リ

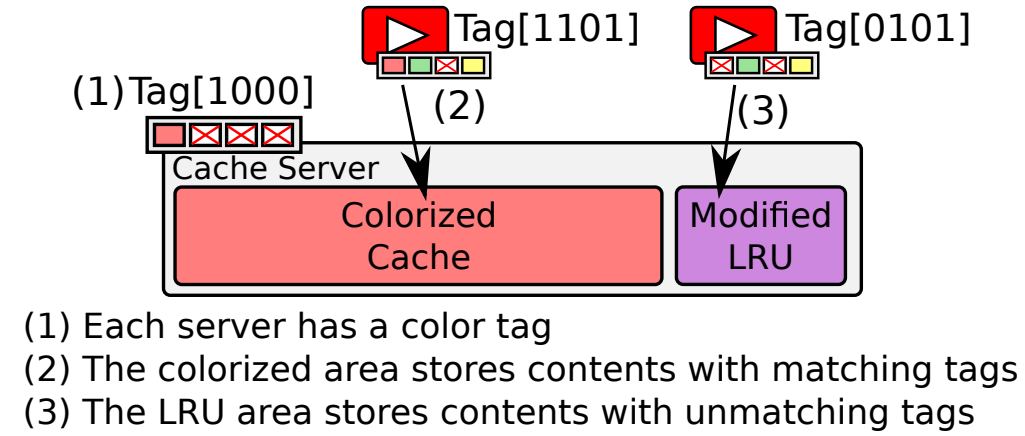


図 4.11: ハイブリッドキャッシュ領域

Algorithm 4 Request handling with hybrid caching scheme

```
1:  if requested content exists in the LFU area then
2:      content ← fetchFromCache(key, LFU)
3:  else if requested content exists in the LRU area then
4:      updateRank(key, LRU)
5:      content ← fetchFromCache(key, LRU)
6:  else
7:      content ← fetchFromOrigin(key)
8:      if (colorbit(content) & colorbit(server)) ≠ 0 then
9:          while availableSize(LFU) < sizeof(content) do
10:             if LFU has a content without the same color then
11:                 evict it from LFU area
12:             else
13:                 evict the least popular content from LFU area
14:             end if
15:         end while
16:         insert(content, LFU)
17:     else
18:         while availableSize(LRU) < sizeof(content) do
19:             evict the oldest content from LRU area
20:         end while
21:         insert(content, LRU)
22:     end if
23:     return content
24: end if
```

クエスト転送時にはまず DNS を用いてホスト名が解決されるため，DNS を組み合わせてユーザが近接するキャッシュサーバへリクエストを転送する．具体的には，地域ごとに異

なるレコードを返答する GeoDNS[2, 3, 17] を組み合わせて，ユーザと近接するキャッシュサーバにリクエストを転送する．GeoDNS では，リクエスト元 IP アドレスからアクセス元の位置情報を推定し，位置情報ごとに返答する DNS レコードを設定することができる．ユーザは通常，ISP によって管理されている DNS サーバを参照しているため，DNS サーバはリクエスト元 IP アドレスと契約場所を紐付けることができる．そのため，ユーザは DNS サーバで名前解決することで，最も近いキャッシュサーバを利用できる．具体的には，ユーザがコンテンツを取得する際，`http://cdn.example.com/video01.mp4?tag=0011` のような URL を指定したリクエストが生成される．この際，GeoDNS 機能を利用して名前解決することで，ユーザから最も近いキャッシュサーバの IP が返答される．その結果，コンテンツリクエストはユーザから近接するキャッシュサーバへ転送される．

キャッシュサーバ間のリクエスト転送

色タグベースのキャッシュ制御は，既存の最短経路制御でも高い通信量削減効果を達成する．これは，最短経路中に複数の異なる色が割り当てられたサーバが複数存在するため，ユーザが生成したリクエストがオリジンサーバまでルーティングされる途中でキャッシュヒットする確率が高いためである．しかしながら，ユーザから最も近いコンテンツは最短経路とは別の経路上に存在していることがあるため，色タグ情報をもとにした効率の良い経路制御アルゴリズムを提案し，さらなる通信量削減を図る．

図 4.12 に，色タグベースの経路制御の概要を示す．図中の円はキャッシュサーバを表しており，キャッシュサーバ間は論理的に接続されている．物理的な構成については後述する．ネットワーク中には図示されていない管理サーバが存在しており，一定時間ごとにコンテンツの色タグを更新したり，キャッシュサーバ間のリクエスト転送規則を制御する．ユーザは色タグ情報が付与されたコンテンツリクエストを生成し，GeoDNS を利用して近接するキャッシュサーバに送信する．色タグ情報はオリジンサーバから受け取った HTML ファイル内の URL パラメータとして記載されている．たとえば，`http://cdn.example.com/video01.mp4?tag=0011` のように，末尾に付与される． ユーザはオリジンサーバから取得した HTML ファイルをブラウザで表示し，リンクをクリックしてコンテンツを取得する．すなわち，HTML ファイル内に記載されたコンテンツ URL のリクエストパラメータとして色情報を含めておけば，ユーザがリンクをクリックした際に，色情報を含むリクエストが生成され

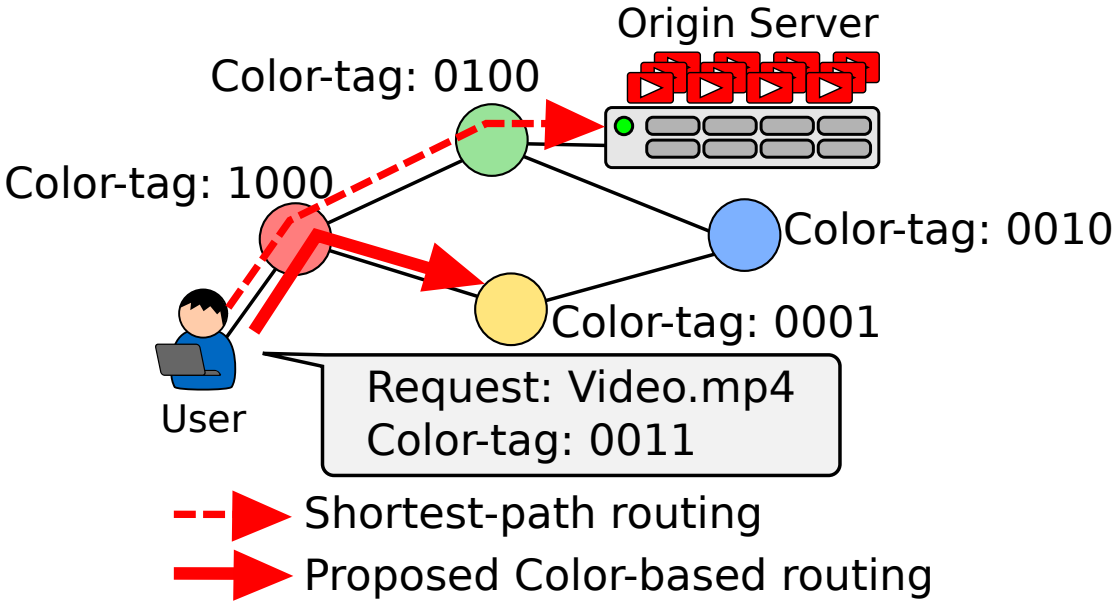


図 4.12: 色タグベースの経路制御の概要

る． キャッシュサーバは，近接するキャッシュサーバの IP アドレスと付されている色タグのリストを保持している．キャッシュサーバがリクエストを受け取ると，まず，要求されたコンテンツがキャッシュヒットするかを確認する．キャッシュミスした場合は，リクエストに付与された色タグ情報を抽出し，色がマッチするキャッシュサーバの中から最も近いサーバへリクエストを転送する． このようにすることで，キャッシュサーバは巨大な経路制御表を持つことなく，軽量な仕組みで経路制御が可能である．

図 4.13 に，色タグベース経路制御が有効なキャッシュサーバの構成を示す．また，色タグベース経路制御あるとリズムを Algorithm 3 に示す．図 4.13 中に記載された括弧内の数字は，Algorithm 3 中の処理に対応している．各キャッシュサーバはリクエスト経路表とレスポンス経路表の 2 つを供えており，LFU/LRU のハイブリッドキャッシュ領域と，経路制御エージェント，およびネットワークインタフェースで構成される． キャッシュサーバは一般的な x86 サーバを想定しており，生成したパケットはサーバと接続されているルータを介して宛先のサーバまで転送される．すなわち，キャッシュサーバ間では論理ネットワークが構成されており，リクエストの宛先 IP アドレスをソフトウェアで書き換えることで，柔軟な経路制御を行う．

(1) キャッシュサーバがリクエストを受け取ると，まず，リクエストに付与された色情報

を抽出し，(2) 対応するコンテンツをキャッシュ領域から取得を試みる．この際，色情報が自分のサーバとマッチしていればLFU 領域を，マッチ指定なければLRU 領域を探索する．目的のコンテンツがキャッシュされていれば，(3) キャッシュサーバはレスポンス経路表を参照して返答用のネットワークインタフェースを決定し，(4) コンテンツを返送する．表4.6 にレスポンス経路表の例を示す．これは一般的な経路制御表と同じように，ネットワークアドレスと対応する送出ネットワークインタフェースの組が記載されている． ただし，キャッシュサーバから転送されたリクエストに対する返答は，直接ユーザ端末へ返答せず，リクエスト元キャッシュサーバに転送する．このようにすることで，リクエストが経由したキャッシュサーバそれぞれでコンテンツをキャッシュすることができ，次回以降のリクエストに対して高速にキャッシュから返答することができる．また，コンテンツの色タグが更新された際も，経由したサーバでコンテンツがキャッシュされるため，新規コンテンツの取得オーバヘッドを小さく抑えることができる． コンテンツがキャッシュされていなければ，(5) キャッシュサーバはリクエストを転送するネットワークインタフェースを検索し，(6) リクエスト経路表をもとに決定したポートからリクエストを転送する．表4.7 にリクエスト経路表の例を示す．この表は色ビットと対応するネットワークインタフェースのID の組が記載されている．色マスクは色がマッチする各サーバまでのホップ数が昇順になるようにソートされており，上から検索して最初にマッチしたインタフェースからリクエストを転送する．もし，色がマッチするインタフェースが見つからなければ，デフォルトのインタフェースからリクエストを転送する．デフォルトインタフェースはオリジンサーバまで最短経路で通信できるよう設定されている．(7) リクエストを転送したコンテンツを，オリジンサーバまたは隣接するキャッシュサーバから受け取ると，(8) そのコンテンツをキャッシュし，(4) レスポンス経路表をもとにコンテンツを返答する．

基本的には上記のような手順でリクエストを転送するが，サーバにキャッシュされているコンテンツの状況によってはループが発生する可能性がある．そのため，過去に自分が転送したリクエストをまた受け取った場合は，オリジンサーバへの最短経路制御に切り替えることで，ループを回避する．なお，パケット単位でのルーティングは下層の物理ネットワークにまかせ，キャッシュサーバではパケット単位でのルーティングは行わない．

また，リクエスト経路表はネットワーク中に別途存在する管理サーバによって生成される．管理サーバはネットワーク中に存在するキャッシュサーバと付された色情報，およびネットワーク構造を把握しており，各キャッシュサーバを起点として，各色のキャッシュ

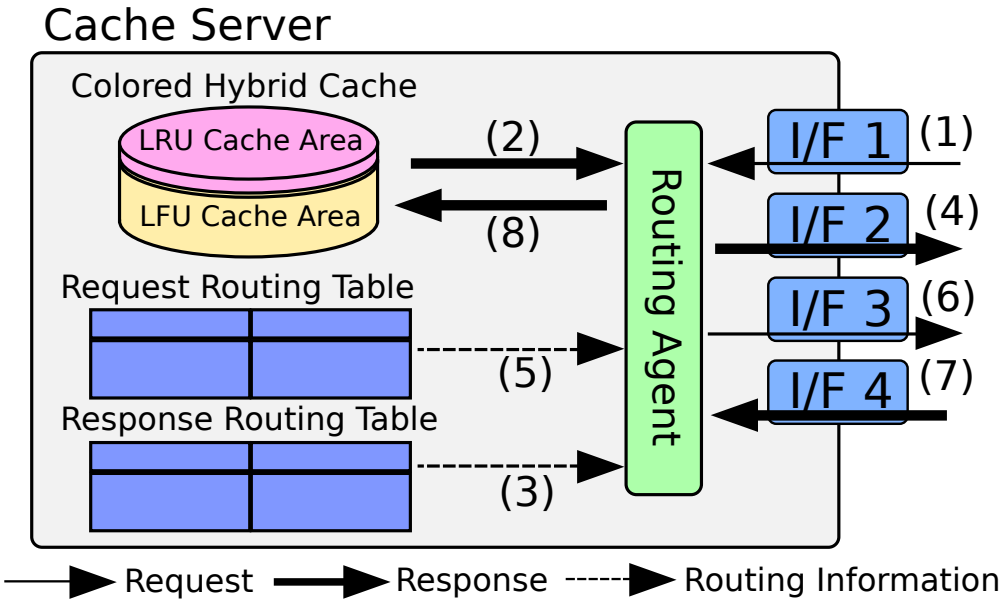


図 4.13: キャッシュサーバを構成するモジュールと接続

サーバまでの最短経路をそれぞれ計算してリクエスト経路表を生成する．具体的には，まず，各キャッシュサーバごとに Dijkstra 法で周囲のキャッシュサーバまでのコストを算出する．次に，サーバに付された色ごとに最短コストで到達できるキャッシュサーバを選定し，そのサーバをリクエストの色と関連付けられた中継サーバとして選択する．このようにすることで，リクエスト色情報をもとに，色がマッチするキャッシュサーバへの最短経路を設定することができる．

色タグベースの経路制御アルゴリズムは，2 種類の色タグベースの経路表を追加する必要があるが，大きな経路制御オーバヘッドは生じない．実際のところ，リクエスト経路表の大きさは最大で色数+1 である．これは，オリジンサーバが保持する数万〜百万種類のコンテンツごとに経路を設定する場合と比較して明らかにエントリ数が少ない．また，コンテンツ指向ネットワーク（CCN）はコンテンツのカテゴリごとに転送先をまとめているが [47]，提案する色タグベースの経路制御表は 8〜16 色程度で十分な通信量削減効果が得られる．

Algorithm 3 Color based routing algorithm

```
1: if received a request then                                              ▷ (1)
2:     if desired content is cached then
3:         content ←fetchFromCache(request)                                ▷ (2)
4:         ipaddr ←sourceAddress(request)
5:         interface ←findResponseInterface(ipaddr)                        ▷ (3)
6:         sendContent(content, interface)                                 ▷ (4)
7:     else
8:         tag ←colorTagOf(request)
9:         interface ←findRequestInterface(tag)                            ▷ (5)
10:        sendRequest(request, interface)                                 ▷ (6)
11:    end if
12: end if
13: if received a response then                                            ▷ (7)
14:    content ← contentOf(response)
15:    if colorTagOf(response) & server_color > 0 then
16:        cacheContent(content)                                           ▷ (8)
17:    end if
18:    ipaddr ←destinationAddress(response)
19:    interface ←findResponseInterface(ipaddr)                            ▷ (5)
20:    sendContent(content, interface)                                     ▷ (4)
21: end if
```

request	content request message from end-user, containing content ID, color tag, and source IP address
response	response message from the origin or a cache server, containing content data, color tag, and destination IP
interface	network interface to transfer data
server_color	color tag bit of the server like 0010

表 4.6: レスポンス経路表の例

Destination	Output I/F
10.0.0.0/8	3
20.1.0.0/16	4
20.2.0.0/16	3
30.0.0.0/8	1
default	2

4.3 評価

4.3.1 想定するネットワーク

提案する分散協調キャッシュの仕組みを評価するため，単方向リングネットワーク，2次元メッシュネットワーク，およびNTTのバックボーンネットワークを模したメッシュネットワーク [44] において，通信量削減効果を計算する．図4.14，4.15，4.26に，評価で使用したリングネットワーク，2次元メッシュネットワーク，およびNTTライクなトポロジと，サーバの彩色結果をそれぞれ示す．リングネットワークと2次元メッシュネットワークは構造が簡単であるため，手作業で着色した．NTTライクなネットワークは構造が複雑であるため，Algorithm 1に示した改変版Welsh-Powellアルゴリズムを使用した．各ネットワーク内に円で示されるノードの下にユーザが接続されていると仮定し，コンテンツリクエストを発生させる．

コンテンツのアクセス頻度はガンマ分布をもとに生成した．ガンマ分布は，ZipfやWeibullなどWebアクセスでよく使用されるリクエストモデルよりも高精度に動画のアクセスをモデル化できる確率分布である [26]．評価に使用したコンテンツ総数，各キャッシュサーバのキャッシュ容量，およびガンマ分布のパラメータを表4.8に示す．

各人気クラスに属するコンテンツ数は，Algorithm 2に示す人気クラス決定アルゴリズムを使用して設定した．GAをもとに目視で設定した結果と比較したグラフを図4.4に示すまた，理論上通信量が最小となるキャッシュ配置を，遺伝的アルゴリズムをもとに計算した．計算に使用したホストの構成を表4.10に示す．目視で設定した人気クラスは遺伝的ア

表4.7: リクエスト経路表の例

Color mask	Output I/F
0100	3
0001	4
0010	2
1000	1
default	2

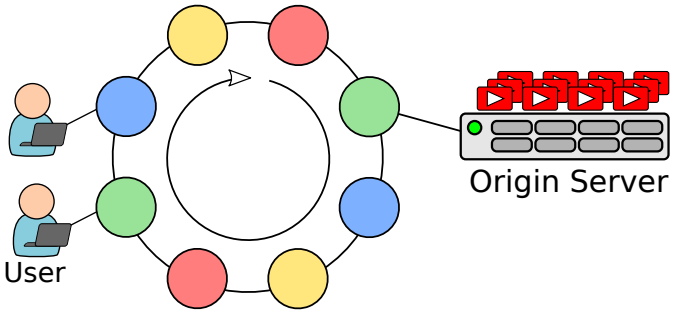


図 4.14: リングネットワークトポロジとサーバ彩色

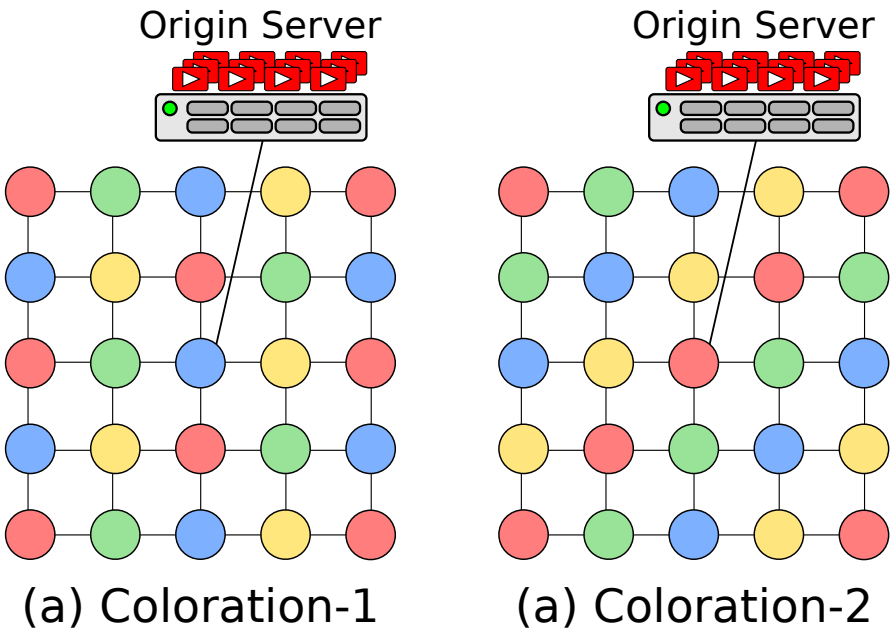


図 4.15: 2 次元メッシュネットワークトポロジとサーバ彩色

ルゴリズムの計算が収束した後に決定した．具体的には，リングネットワーク，2 次元メッシュネットワーク，NTT ライクなトポロジにおいて，3000 世代目，8000 世代目，10000 世代目の最も評価が良かった個体を選択した．

4.3.2 キャッシュ制御方法と通信量の関係

表 4.9 は，各人気クラスのコンテンツ数を示す．たとえば，4 色でコンテンツを彩色したとき，最も人気のある上位 5 つのコンテンツには 1111 の色タグが付与されるため，NTT

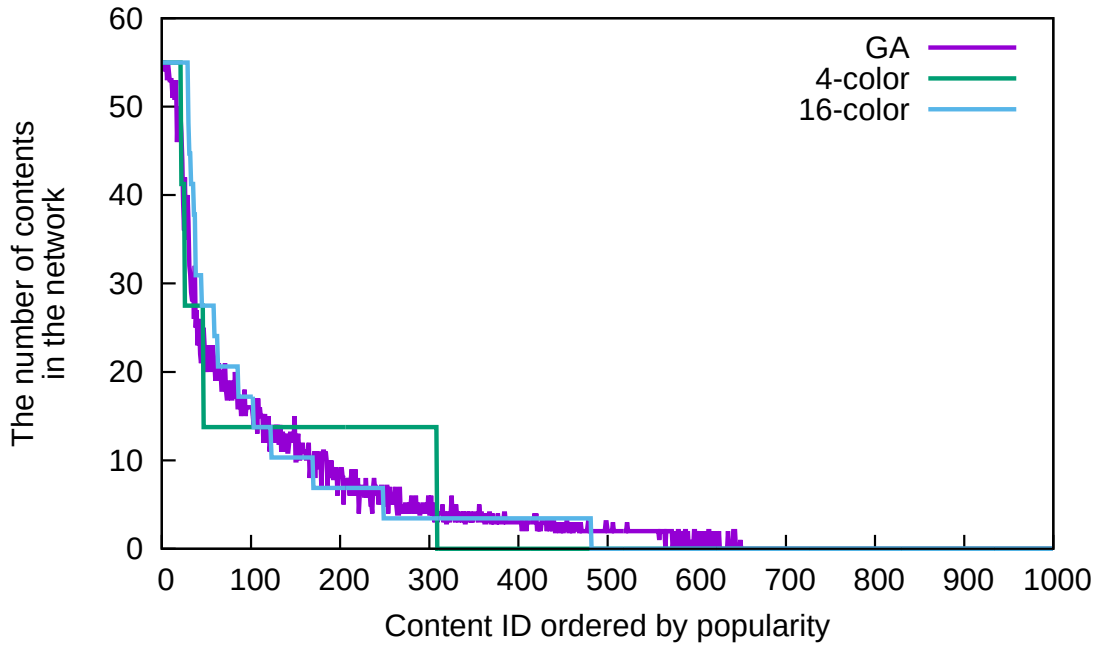


図 4.16: ネットワーク中に保持されるコンテンツ数

トポロジのすべてのキャッシュサーバで保持される．一方，2 番目に人気の 20 個のコンテンツは，4 ビット中 3 つのビットが 1 で構成される色タグが付与されるため，全体の 75%のサーバで保持される．なお，通信量削減効果を最大化するため，コンテンツの人気クラスは各キャッシュサーバの容量を超えないように設定した．提案するコンテンツ色分散は，準最適カーブと比較して大きく異ならないようになっている．

図 4.16 は，遺伝的アルゴリズムで求めた準最適キャッシュ配置および，異なる色数を使用して分散保持された，ネットワーク中コンテンツ数を示す．16 色でコンテンツ彩色した結果は，4 色で彩色した結果と比較して，準最適なキャッシュ配置のコンテンツ数に近いことは明らかである．すなわち，色タグベースの経路制御アルゴリズムを使用することで，色がマッチするキャッシュサーバへリクエストを転送できるようになるため，色タグベースの経路制御アルゴリズムを使用すると彩色色数が多いと通信量削減効果が高くなることが予想される．

4.3.3 通信量の評価

キャッシュなし，一般的な LFU，GA[4] と色キャッシュを比較した通信量の評価結果を図 4.17 に示す．コンテンツリクエストは静的なものとして評価したため，ハイブリッドキャッシュの比率は色キャッシュ領域を 100%とした．各コンテンツの色情報は，オリジンサーバがリクエストの偏りをもとに事前に彩色した．GA も同様に，リクエスト頻度が既知のものとして，通信量小さくなる準最適なキャッシュ配置を事前に計算した．GA は，

表 4.8: シミュレーションに使用したパラメータ

Total content	1000
Cache capacity	100
Popularity distribution	Gamma distribution
Gamma parameter k	0.475
Gamma parameter θ	170.6067
Topology	NTT-like network [44]
Number of servers	55

表 4.9: 各人気クラスに所属するコンテンツ数

	High	Mid-High	Middle	Mid-Low	Low
Ring	10	10	80	170	730
2D-mesh	25	25	40	145	765
NTT	5	20	60	200	715

表 4.10: 計算ホストの構成

CPU	Intel Core i7-3930K @3.80GHz
CPU Core	6 physical cores / 12 logical cores
RAM	64GB

キャッシュ配置を 1 個体とし，選択，評価，交叉，突然変異のプロセスを繰り返すことで，通信量を最小化するキャッシュ配置を計算する [4]．計算手順は文献 [4] と同様であるが，計算時間の短縮とプログラムの複雑化回避のため，各リンクの通信帯域と通信遅延の制約を排除して計算した．ここで，External traffic はオリジンサーバと東京に位置する黄色いノード間のリンクの通信量を示し，Internal traffic はその他のリンクの通信量の総和を示す．キャッシュ無しは，ネットワークにキャッシュサーバを導入しなかった場合に相当し，通信量の比較のために使用する．すべてのキャッシュアルゴリズムは最短経路制御を使用する．なお，すべてのクライアントから同量のリクエストが同一のリクエスト頻度で生成され，通信量は文献 [4] と同様に，キャッシュサーバで保持されているコンテンツとコンテンツごとのリクエスト頻度情報から各リンクが利用される期待値を算出し，その総和を計算した．そのため，各拠点のユーザ数やアクセス回数は設定されていないが，求めた期待値にアクセス数やユーザ数，コンテンツサイズをかけ合わせると通信量の概算として扱うことができる．

表 4.11: セパレータランクの事前計算に要した時間

# of colors	calculation time
4-color	2m2s
8-color	4m1s
16-color	16m2s
GA	763m35s

表 4.12: GA の計算時間と世代の関係

Topology	Nodes	Generation			
		1000	3000	8000	10000
Ring	8	5m33s	16m01s	42m05s	52m33s
2D-mesh	25	34m44s	103m11s	274m40s	343m17s
NTT	55	42m08s	127m34s	350m38s	440m25s

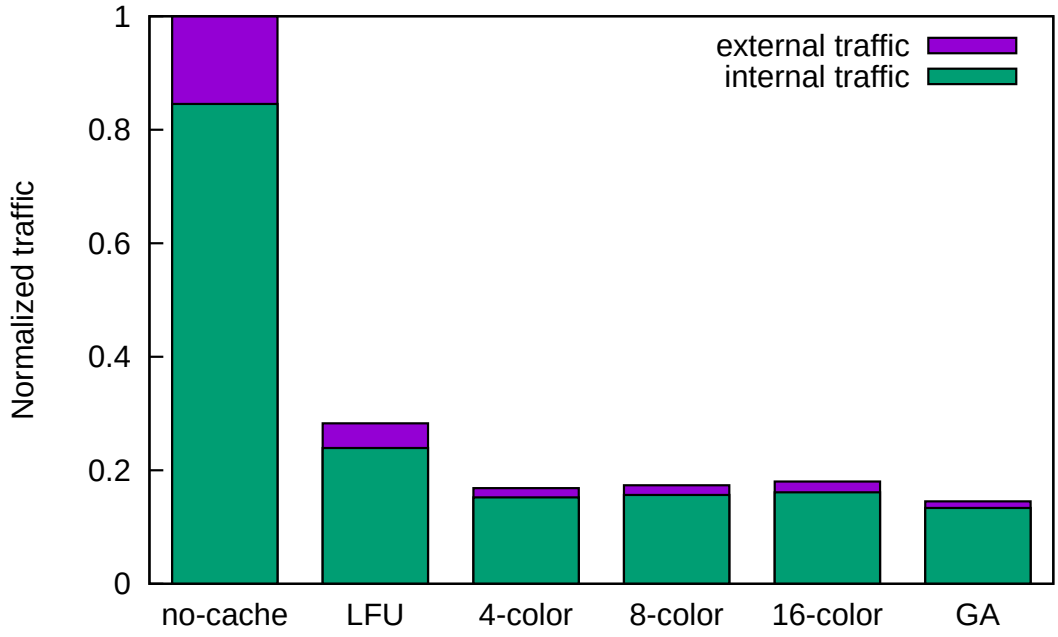


図 4.17: 色数と通信量の関係

図 4.18，4.19，4.20 に，単方向リング，2 次元メッシュ，および NTT ライクなトポロジでの通信量削減評価結果を示す．各ネットワークにおける準最適配置は，遺伝的アルゴリズムで 3000，8000，10000 世代計算したものを使用した．これらの図から，通信量削減効果はネットワーク規模が大きくなると高くなることがわかる．これは，ユーザからコンテンツまでの平均ホップ数を，相対的にユーザにより近い位置キャッシュでできるようになるためである．2 次元メッシュでは，サーバ彩色パターン 1 と 2 があるが，パターン 1 の方が通信量削減効果が高かった．これは，オリジンサーバまでに経由するキャッシュサーバに存在する色数の期待値が，彩色パターン 1 の方が大きいため，経路中でのヒット率が高くなった結果である．この結果から，通信量削減効果の高いサーバ彩色アルゴリズムが存在することが示唆される．

すべてのトポロジにおいで，色タグベースの分散協調キャッシュは，準最適に近い削減効果を示した．具体的には，GA と比較して，単方向リング，2 次元メッシュ，NTT トポロジで，ヒット率の差はそれぞれ 0.9%，1.5%，2.3%にとどまっている．これらの結果を踏まえ，コンテンツの重複・分散配置が，通信量削減に重要な役割を担っていることがわかる．

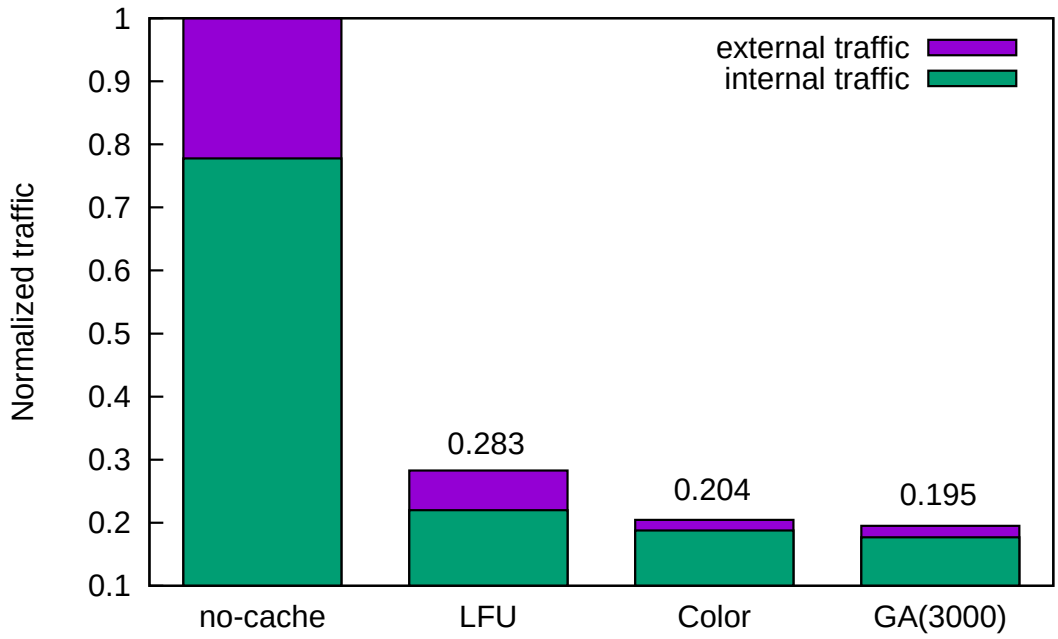


図 4.18: 単方向リングトポロジにおける通信量削減効果

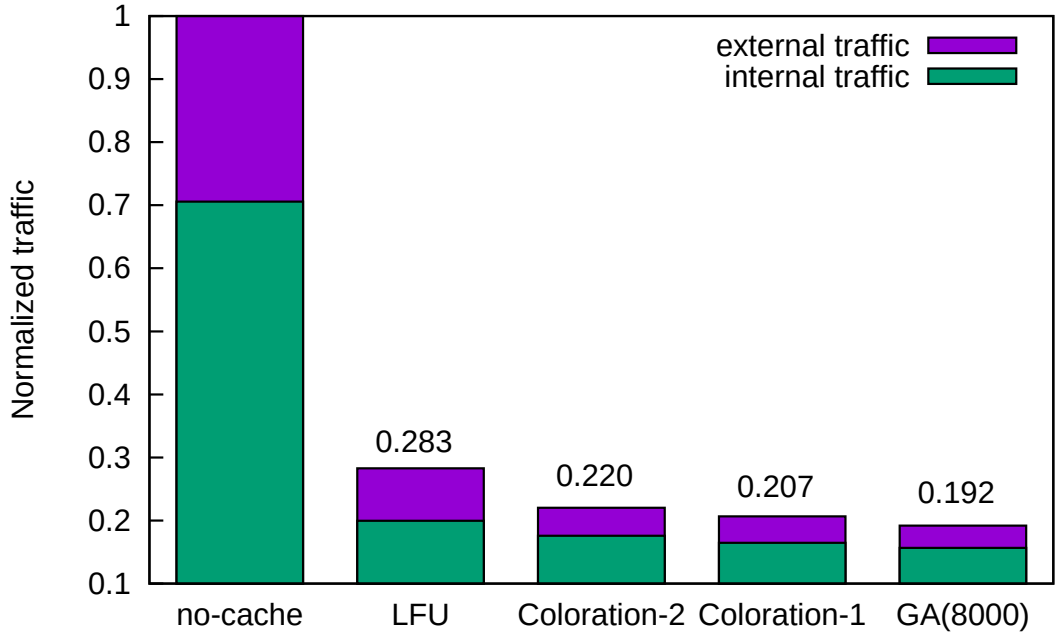


図 4.19: 2D メッシュトポロジにおける通信量削減効果

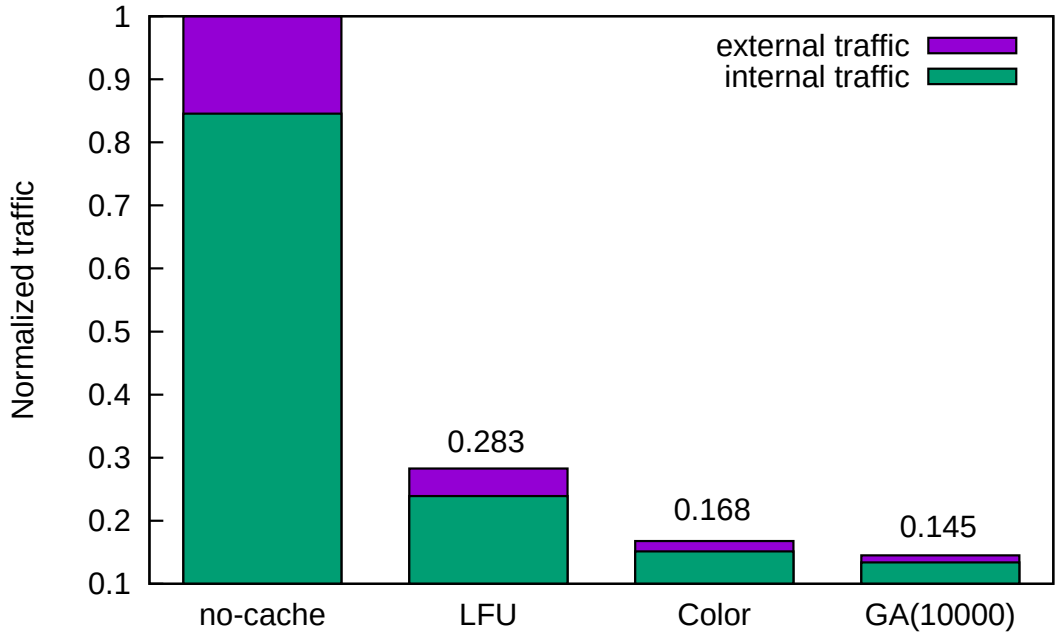


図 4.20: NTT ライクなトポロジにおける通信量削減効果

LFU アルゴリズム [31] は，アクセスパターンが変化しない場合に理論上通信量削減効果が最大となるアルゴリズムであるが，各ノードに接続されるユーザが生成するアクセスパターンは一定であるため，評価を通してすべてのキャッシュサーバで同じコンテンツを保持していた．その結果，キャッシュ領域を有効活用できず，十分な通信量を削減できていない．色タグベースの分散協調キャッシュは，LFU と比較して半分近い通信量を削減しており，4 色でコンテンツを彩色した場合，GA で求めたキャッシュ配置と比較して，通信量削減効果の差は 2.3%程度と大変小さい．8 色および 16 色でコンテンツを彩色すると，4 色でコンテンツを彩色するよりも通信量削減効果が小さくなっている．これは，使用する色数が多いと，キャッシュサーバがリクエストを転送する際に，最短経路上にリクエストに付与された色とマッチする色のサーバの存在確率が低くなるからである．色タグ情報を考慮した経路制御アルゴリズムを使用することで，さらなる通信量削減効果が期待される．色タグ情報を使用する経路制御アルゴリズムを適用した通信量削減の評価は後ほど述べる．

ハイブリッドキャッシュの通信量評価

通常，色キャッシュは一定時間ごとにコンテンツのアクセス頻度を集計し，各コンテンツを再彩色するが，ニュースや SNS の影響で急激に変動する，予測不可能なアクセス集中すると，色キャッシュ単体では対応できず，通信量が増大する．そこで，ある時刻に人気コンテンツが追加された状況を想定し，ハイブリッドキャッシュを有効にした際の通信量を評価した．ハイブリッドキャッシュは，Algorithm 2 に示す通りに制御され，色領域のみを使用する場合は，Algorithm 2 中の 3–5 行と 12–16 行が削除される．これらの行は，LRU 領域の制御を担っている部分である．

図 4.21 に，ある時刻に 5 つの人気コンテンツを，人気クラス最低（色タグが 0000）の扱いで追加した際のヒット率を示す．ハイブリッドキャッシュの比率は，色付き LFU 領域を 90%，Modified LRU 領域を 10%とした．新規コンテンツを追加する前は，ハイブリッドキャッシュの有効無効でヒット率は大きく変わらない．色付き LFU 領域単体の純粋な色キャッシュアルゴリズムでは，コンテンツが追加された直後にヒット率が 13.9%低下している一方で，ハイブリッドキャッシュを有効にすると，ヒット率の下落幅は 2.3%にとどまっている．これは，色がついていないコンテンツに対しても，Modified LRU 領域はキャッシュ対象としているためである．ごく短期的に見ると，アクセス変動時はキャッシュに存在しないコンテンツを取得する必要があるので，キャッシュヒット率が下落する．しかし，数回程度のキャッシュミスはその後のキャッシュヒット数と比較して非常に少ないため，1000 リクエストごとのキャッシュヒット率を計算している図 4.21 のグラフには，短期的なヒット率の下落は現れていない．

図 4.22 に，LRU 領域の比率と，コンテンツが追加された直後の通信増加量の関係を示す．この図を見ると，キャッシュ領域のごく一部を LRU 領域に変更するだけで，通信量を効率よく削減できていることが読み取れる．具体的には，キャッシュ容量の 5%を Modified LRU 領域とすると，1，5，10 種類のコンテンツを追加した直後の通信量を，96.3%，86.0%，71.7%削減できている．

4.3.4 計算量の評価

表 4.12 に，各トポロジにおける，GA の各世代までの計算に要した時間を示す．ネットワークの規模が拡大するにつれ，キャッシュ配置のバリエーションが増えるため，ある世

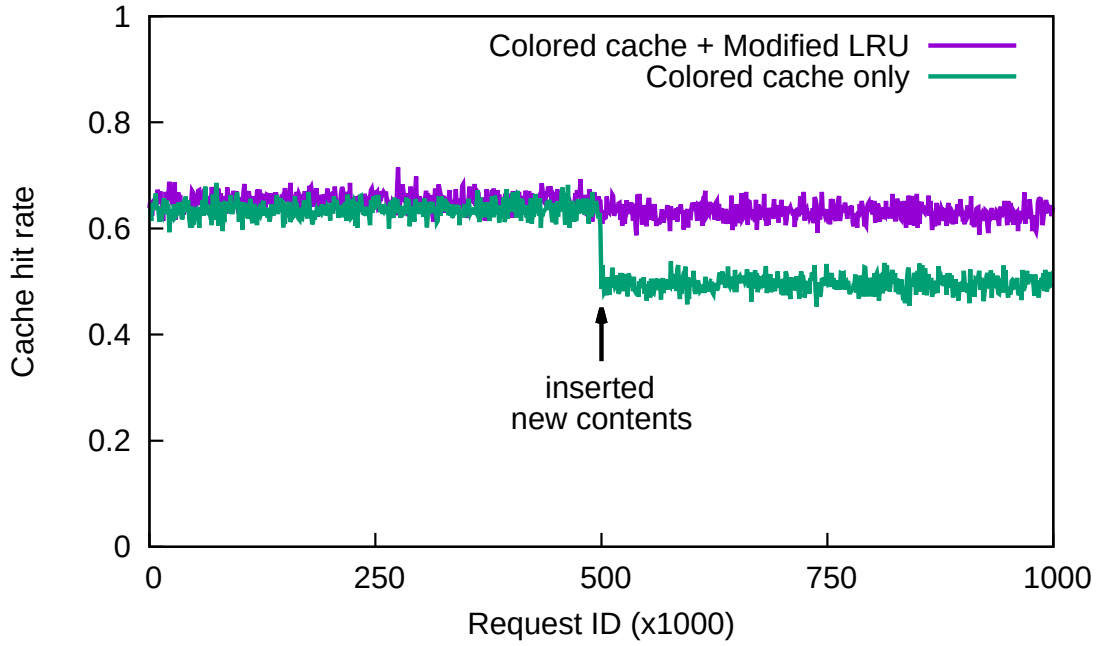


図 4.21: ハイブリッドキャッシュアルゴリズムの変動追従性能

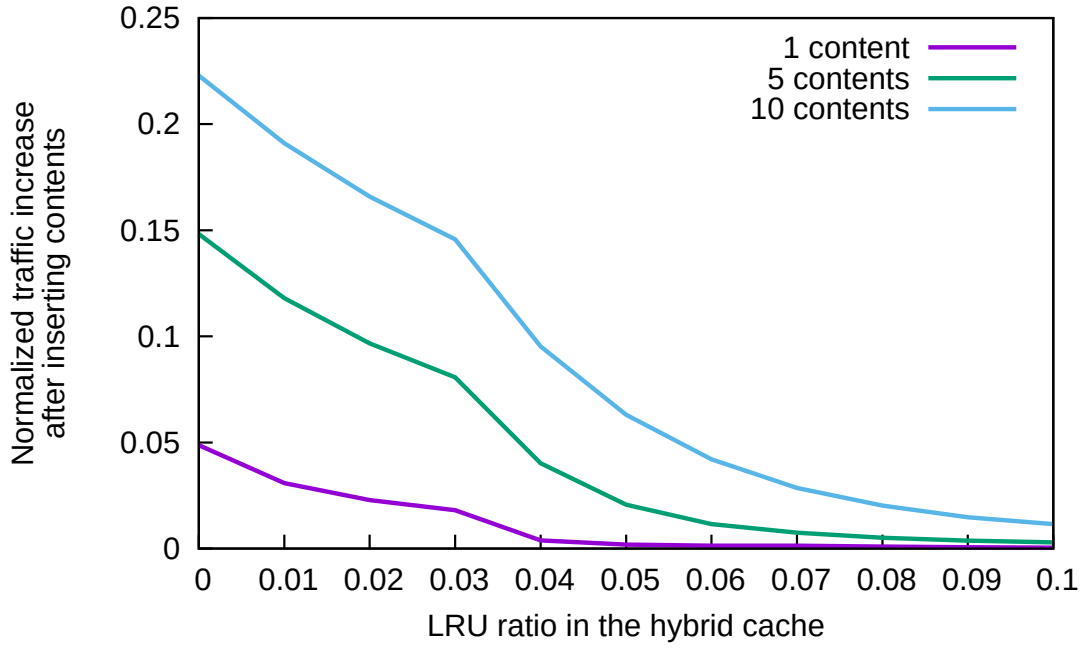


図 4.22: LRU 割合と通信量の関係

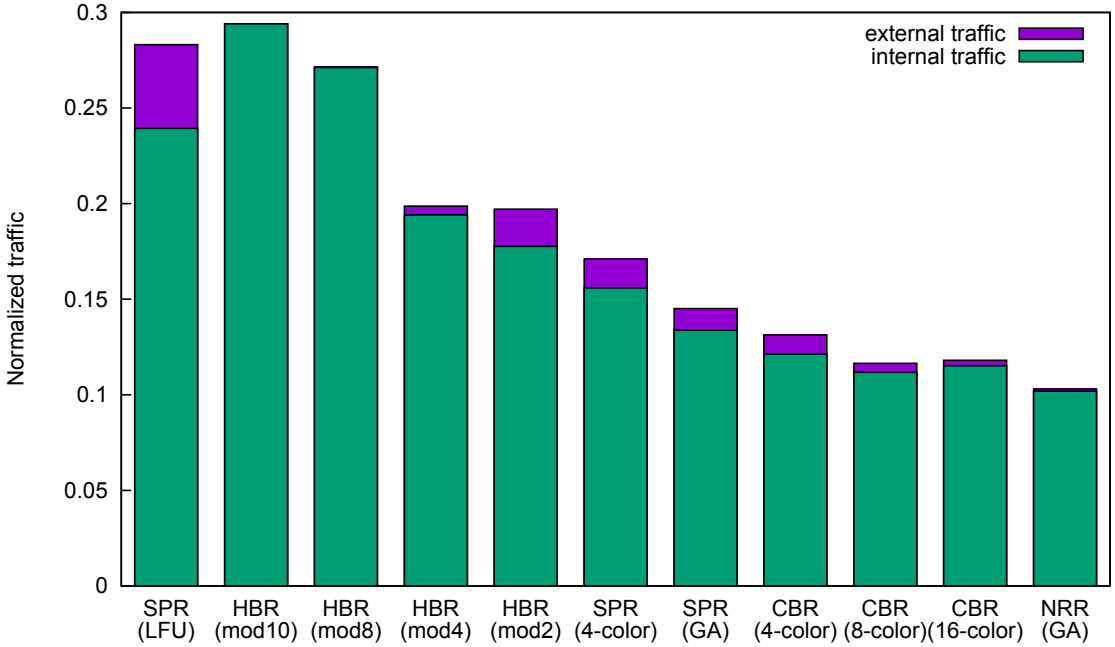


図 4.23: 経路制御手法と通信量の関係

代までの計算を終えるまでの計算時間は長くなっている．これは，より実際的で大規模なネットワーク環境に GA を適用しようとすると，膨大な計算時間を要することを示唆している．たとえば，55 ノードで構成される NTT ライクなトポロジでは，準最適キャッシュ配置を得られるまでに 7 時間以上の時間を要している．一方，コンテンツの再彩色に要する時間は，1000 コンテンツに対して数秒程度で終了した．これは，コンテンツの彩色手順が，ソートとアクセスの偏りパラメータの推計だけで完了するためである．この処理に要する計算量は，O($m$log$n$) である．

色タグ情報に基づく経路制御アルゴリズムの評価

色タグベースの経路制御（CBR），Dijkstra のアルゴリズムをもとにした最短経路制御（SPR）[48]，ハッシュベースの経路制御（HBR）[12]，および最も近いキャッシュへの経路制御（NRR）で，通信量削減効果を比較した．NRR は，コンテンツ指向ネットワークで採用されているアルゴリズムである．HBR では，南東のキャッシュサーバから順に ID を設定し，($content\_id$ mod N) = ($server\_id$ mod N) となる場合にキャッシュするよう制御した．

図 4.23 に経路制御アルゴリズムと通信量の関係を示す．使用したキャッシュアルゴリズムは，横軸のラベルの括弧内に記載してある．CBR（4 色）は，SPR（4 色）と比較して 23.2%の通信量を削減できた．また，CBR（8 色）は，GA を除くキャッシュ制御の中で最大の通信量削減効果（SPR（4 色）と比較して 31.9%の通信量削減）を達成した．これは，色数を多くすると，人気のないコンテンツはネットワーク中に数台のキャッシュサーバでしか保持されず，ユーザからコンテンツまでの距離が遠くなることがあるためである．しかしながら，CBR（16 色）は色キャッシュ制御の中で，External traffic の削減効果が最大である．これは，サーバが 16 種類の色で分散されているため，各サーバで保持するコンテンツが異なり，ネットワーク内で利用できる実効キャッシュ容量が大きくなるためである．ネットワーク形状や，内外の通信量のバランス，消費電力等のパラメータを導入することで，各ネットワークで適切な色数を決定できる可能性が示唆される．たとえば，巨大なネットワークや各サーバのキャッシュ容量が小さなネットワークでは，多数の色を使用すると通信量削減効果が大きくなると予想できる．また，CBR は HBR と比較して通信量削減効果が高かった．HBR（mod10）は，すべてのコンテンツをネットワーク内に保持でき，HBR（mod4）は，HBR の中でバランスよく通信量を削減できている．しかしながら，HBR 制御は人気のコンテンツが少数のキャッシュサーバでしか保持されず，ネットワーク内通信量を増大させてしまう．結果的に，CBR（8 色）は HBR（mod4）と比較して 41.4%の通信量を削減できている．

図 4.24 に，経路制御アルゴリズムと通信リンクごとの通信量の回数を示す．各拠点のクライアントから 5000 回のリクエストを発生させてキャッシュのウォーミングアップを実施した後，各クライアントで 1000 回ずつリクエストを発生させ，各リンクの通信回数を計測した．なお，キャッシュサーバは 8 色で彩色し，各サーバのキャッシュ領域は 95%を色制御の領域，5%を Modified LRU 領域として実験を行った．リンク ID は南から北に向かって設定されており，オリジンサーバとの通信リンクを最後に設定した．リンク ID は単方向ごとに設定されている．すなわち，あるノード A とノード B の双方向リンクは，A → B と B → A を別のものとしてリンク ID が設定されている．最短経路制御では，各キャッシュサーバからオリジンサーバまでの最短経路を使用してコンテンツが転送されるが，それ以外のリンクは使用されないため，リンク負荷のばらつきが大きい．一方，色情報を用いた経路制御では，最短経路以外のリンクが使用されており，ネットワーク全体の負荷が分散している．これは，色がマッチする場合はその方向へリクエストが転送され，コンテンツ

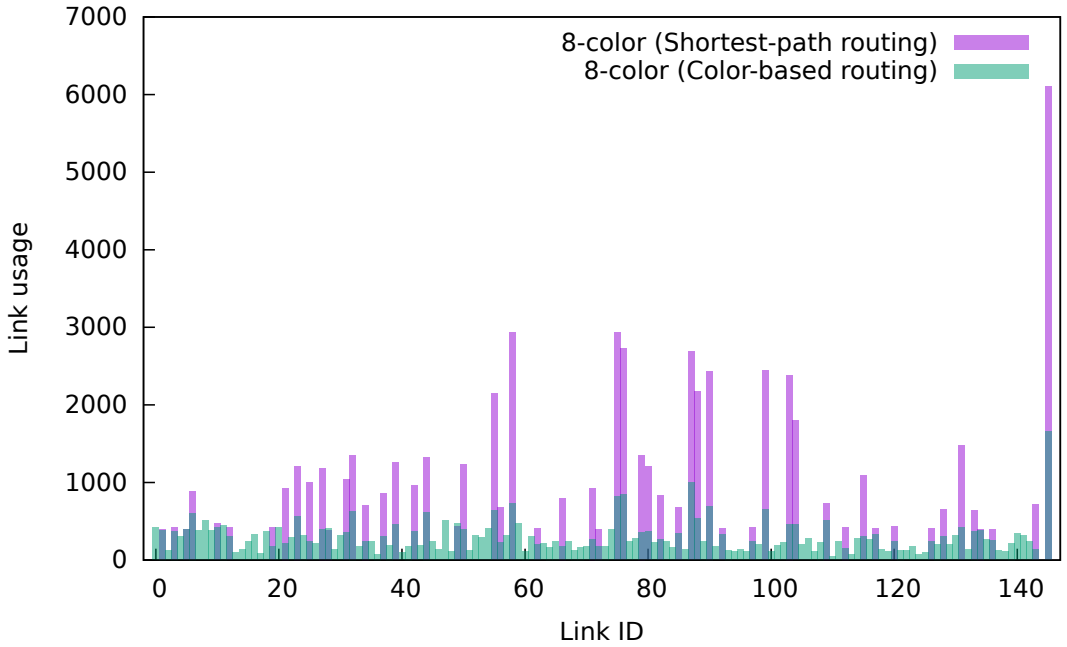


図 4.24: 経路制御手法とリンク使用回数の関係

はリクエストの転送路を逆順に辿って転送されるためである．このように，色情報を用いた経路制御手法は，通信負荷を分散して特定リンクへの負荷集中を軽減し，ネットワーク中のボトルネックを緩和する上で有用である．

コンテンツ彩色に要する計算オーバヘッドの評価

キャッシュ配置計算に要する時間は，従来の GA による方法と比べて大幅に削減された．表 4.13 に，あるガンマの偏りパラメータ k について，通信量削減効果を最大化するセパレータランクを計算するために要した時間を示す．GA での計算には 7 時間以上の時間を要していたが，この表を見ると，4 色，8 色，16 色のどの場合でも 20 分以内で計算が終了している．また，アクセスログから偏りパラメータ k を推定する計算についても，1,000,000 下位のアクセスログをもとに計算した結果，6 秒程度で完了した．さらに，コンテンツ彩色はアクセス頻度の計算，ソート，循環的なタグ付けにとどまるため，計算時間は数秒程度で終了した．これらの結果から，色タグベースのキャッシュアルゴリズムは，アクセスの偏りが変化したとしても，高い通信量削減効果を維持できる．

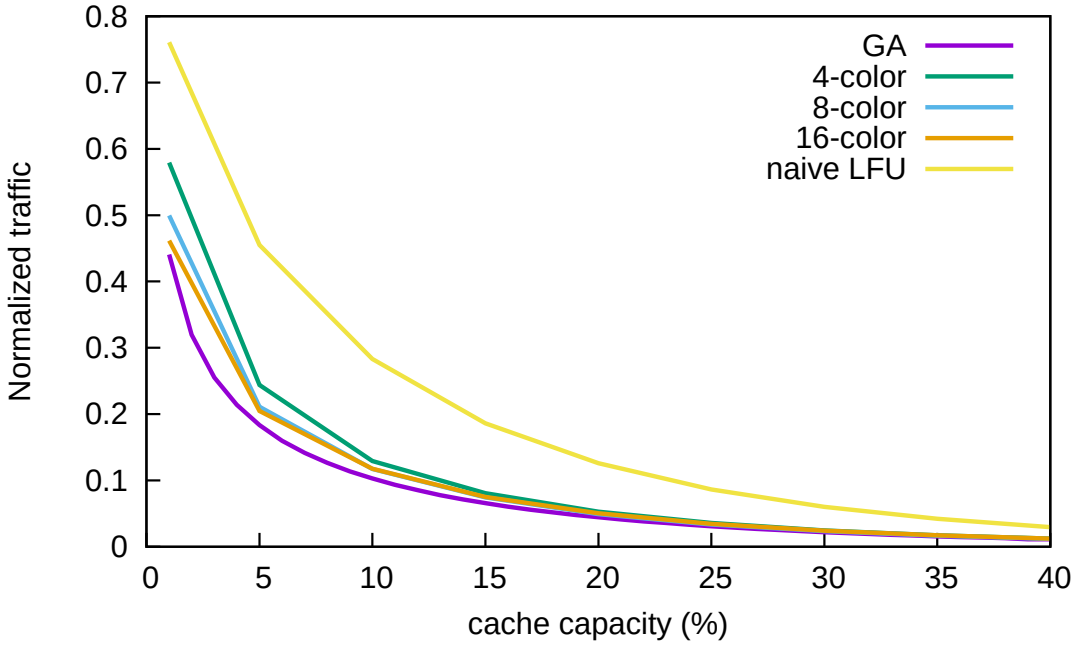


図 4.25: キャッシュ1 台あたりのストレージ容量と通信量の関係

キャッシュ容量と通信量の評価

図 4.25 に，キャッシュ容量と通信量の関係を示す．キャッシュ容量はオリジンサーバが保持するコンテンツ総数を 1 とした割合で示されており，1%から 40%まで評価した．なお，大手 VoD 事業者の Netflix が各所に展開しているキャッシュサーバは，1 台あたりコンテンツ全体の 10%をキャッシュできるとされている．4，8，16 色で制御した色キャッシュは，GA で求めた通信量削減効果に近い．キャッシュ容量が小さいと人気のないコンテン

表 4.13: セパレータランクの事前計算に要した時間

# of colors	calculation time
4-color	2m2s
8-color	4m1s
16-color	16m2s
GA	763m35s

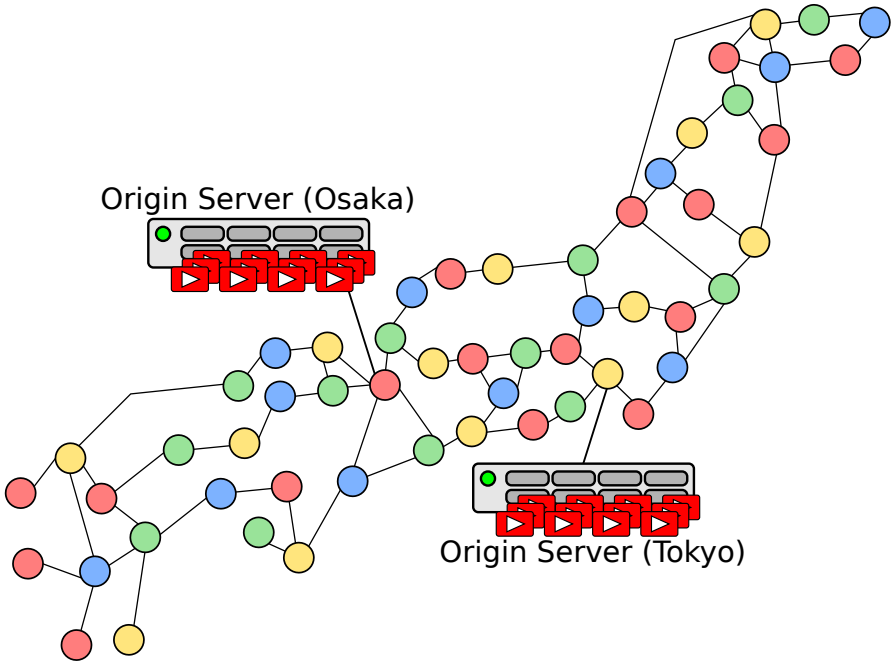


図 4.26: NTT コアネットワークを模したトポロジのサーバ彩色例

ツを保持したときのペナルティが大きくなるため，キャッシュ制御方法の選択で通信量削減効果の差が大きい．さらに，色キャッシュの結果は，色数が増えると GA に近くなっている．これは，巨大なキャッシュ容量があると，ほとんどのコンテンツをキャッシュできてしまうため，一般的にキャッシュアルゴリズムの効果が小さくなるためである．

複数オリジンサーバ状況下における通信量の評価

複数台のオリジンサーバが存在する環境で通信量を評価した．オリジンサーバは，図 4.26 に示すように東京と大阪に設置し，リクエスト元から近い方へ優先的にリクエストを転送するよう設定した．図 4.27 に，色キャッシュと LFU アルゴリズムによる通信量削減効果を示す．まず，4 色で制御した色キャッシュは，内外のネットワーク通信量を削減できており，オリジンサーバが 1 台の場合も 2 台の場合も LFU と比較して通信量削減効果が高い．これらの結果は，主に分散協調キャッシュの有無によって生じたものである．次に，内部通信量について，LFU を使用した場合も色キャッシュを使用した場合も，オリジンサーバが増えると内部通信量が削減されている．これは，リクエスト元とオリジンサーバまでの平均ホップ数が短縮するためである．最後に，外部通信量は，キャッシュ制御ア

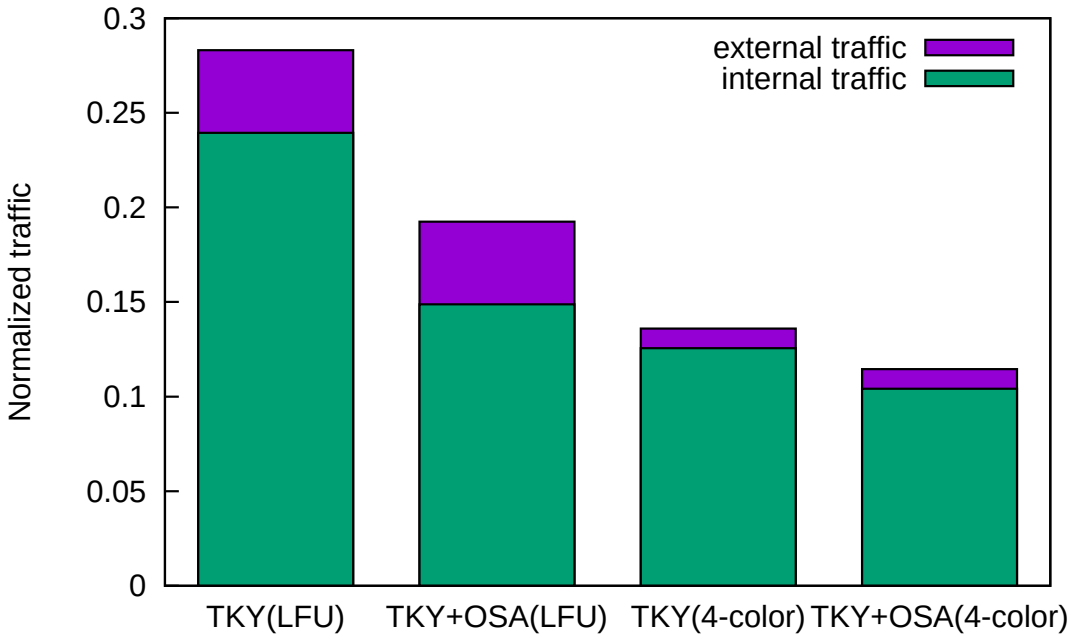


図 4.27: オリジンサーバの設置台数と通信量の関係

ルゴリズムが共通していれば，オリジンサーバの台数による変化はみられない．これは，キャッシュ制御アルゴリズムが同じであれば，ネットワーク中に保持されるコンテンツの種類数が変わらないため，オリジンサーバに到達するリクエストが一定であるためである．以上の結果から，オリジンサーバ台数が複数台ある環境下でも 4 色で制御したキャッシュ制御は効果的にオリジンサーバへの通信量を削減できることが示された．

階層ネットワークにおける評価

キャッシュネットワークを階層化した際に削減される通信量を評価した．ネットワークは LongHaul，Metro，Access の 3 階層で構成されており，各階層にキャッシュサーバを設置する．評価に使用したネットワーク構造を図 4.28 に示す．ここで，LongHaul には図 4.26 に示した NTT ライクなトポロジを，Metro には図 4.14 に示した単方向リングを，Access にはキャッシュサーバを単一ノードで設置し，ネットワークを構成した．オリジンサーバは東京のみに設置し，キャッシュ容量は LongHaul で 10%，Metro と Access でそれぞれ 2%に設定した．図 4.28 にはスペースの関係上，Metro および Access ネットワークが 1 つずつしか記載されていないが，LongHaul ネットワークの各キャッシュサーバには Metro ネット

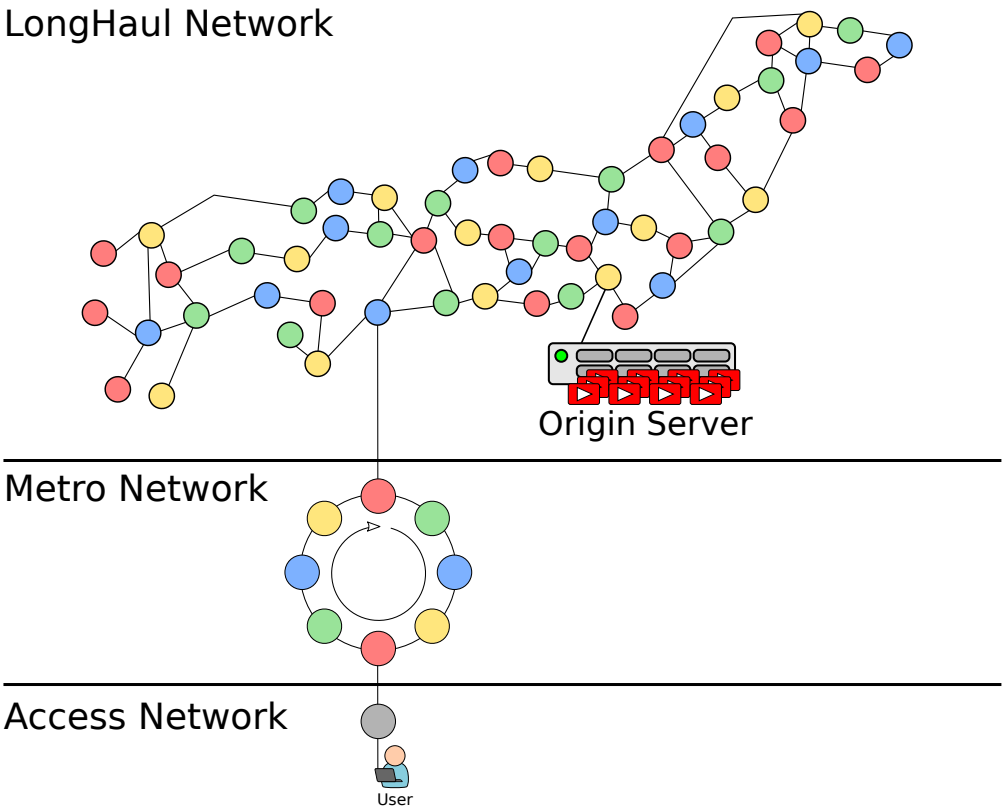


図 4.28: 3 階層の実験ネットワーク構造

ワークが，Metro ネットワークの各キャッシュサーバには Access ネットワークのキャッシュサーバがそれぞれ接続されている．なお，Access ネットワークのキャッシュサーバは 1 台しかなく，分散協調制御が適用できないため，LFU アルゴリズムでキャッシュを制御した．

図 4.29 に階層キャッシュの設置有無と通信量の関係を示す．どの階層にもキャッシュサーバを設置しない場合と比較して，LongHaul だけにキャッシュサーバを設置すると 47.8%，3 つの階層すべてにサーバを設置すると 76.1%の通信量が削減された．また，3 つの階層すべてにサーバを設置すると，LongHaul だけにキャッシュサーバを設置した場合と比較して 54.2%の通信量が削減された． 階層間リンクに着目すると，Access–Metro 間のリンク負荷が 32.3%，Metro-LongHaul 間のリンク負荷が 71%，LongHaul–Origin 間のリンク負荷が 76.1%，それぞれ減少した． これは，より下位のネットワークにキャッシュサーバを配置することで，上位ネットワークに要求されるリクエストが少なくなったためである．オリジンサーバから配信されるデータ量に着目すると，キャッシュサーバを設置しない場合と比較して，3 階層にキャッシュを配置すると 99.3%削減された．これは，ネットワーク

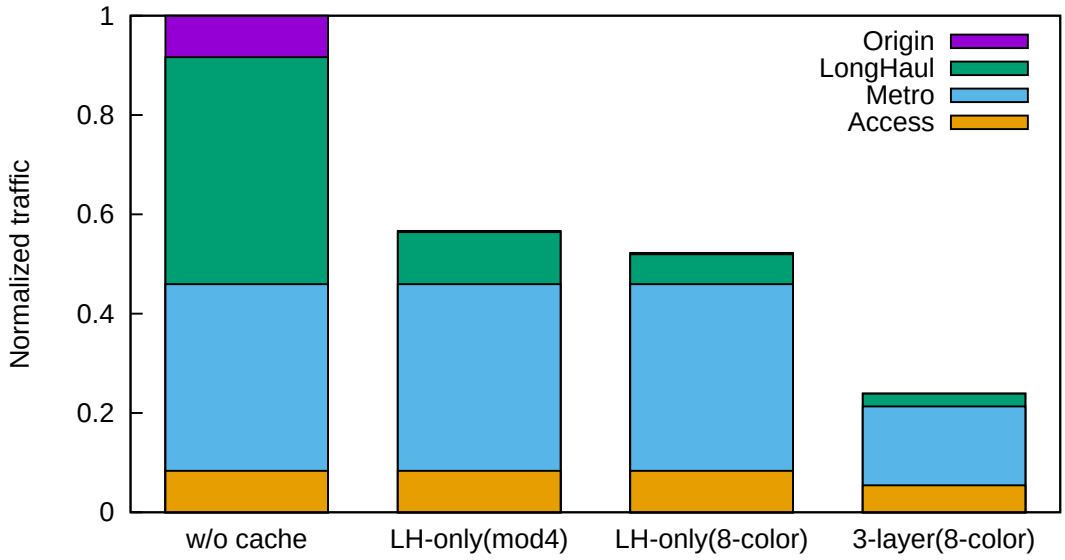


図 4.29: 階層キャッシュの設置有無と通信量の関係

内に設置されたキャッシュサーバが分散協調動作することで実行キャッシュ容量が拡大され，ネットワーク全体の 60%程度のコンテンツを LongHaul ネットワークまででキャッシュヒットさせることができたためである．たしかに，ガンマ分布のアクセス確率を示した図 2.2 を参照すると，人気上位 60%のコンテンツを保持できればヒット率は 99%を上回るため，理論上のヒット率と比較しても妥当な結果である．

4.4 まとめ

キャッシュネットワークに管理方法がシンプルな色タグを導入し，計算が軽量で通信量削減効果の高い分散協調キャッシュの仕組みを提案した．色タグベースのキャッシュ制御アルゴリズムは，GA で計算した準最適なキャッシュ配置に匹敵する通信量削減効果を達成した．さらに，ハイブリッドキャッシュアルゴリズムを採用することで，アクセス傾向が急激に変化してもヒット率を維持できることを示した．提案する軽量なコンテンツの再彩色アルゴリズムは，各キャッシュサーバから収集したアクセスログをもとにアクセスの偏りを推定し，効率のよい再彩色のバランスを数秒の計算で実現した．また，色タグ情報を活用する経路制御アルゴリズムは，最短経路制御と比較して 20%の通信量を削減し，経路制御のオーバヘッドが小さいことを示した．

第5章　結論

5.1　本研究の成果

本論文では，通信量が急増するオンデマンド動画配信サービスの通信量削減を目的として，動画配信ネットワークの構造と動画アクセス傾向に合わせた 2 つのキャッシュ制御手法を提案し，組み合わせて利用することで効率よく通信量を削減できることを示した．

第 1 章では，オンデマンド動画配信サービスの普及に伴い増大する通信量と，現状の対応方法について述べ，本論文のアプローチと構成について説明した．

第 2 章では，動画配信ネットワークにおける高効率な通信量削減方法を検討するために，ネットワーク構造と動画アクセス傾向について調査した．また，ネットワーク構造と動画アクセス傾向に合わせて，既に提案されているキャッシュ制御方法について調査し，その特徴について説明した．第 2 章の関連研究で得られた知見のまとめを表 5.1 に示す．

第 3 章では，2 章で得られた知見をもとに，動画配信ネットワークの通信を効率よくキャッシュするアルゴリズムを考案し，その効果を評価した．第 3 章および第 4 章で得られた成果を表 5.2 にまとめる．動画アクセス傾向は人気上位コンテンツに大きく偏っているため，アクセス傾向が一定の場合は LFU の利用が効果的である．一方で，新規コンテンツの追加や SNS の影響で動画アクセス傾向が急激に変化することがあり，LFU アルゴリズム単体では通信量の増大につながってしまう．そこで，LRU と LFU の 2 つのキャッシュアルゴリズムを組み合わせることで，動画通信を効率よくキャッシュして通信量を削減できることを示した．

第 4 章では，最適化アルゴリズムで求めた準最適キャッシュ配置から，その特徴を分析した．その後，色タグ情報をもとにキャッシュを制御する方法を提案し，その効果を評価した．また，階層構造のネットワークを前提に色タグの拡張方法を説明し，階層ネットワークにおいて高い通信量削減効果を実現できることを示した．

5.2 今後の展望

3 章で提案したハイブリッドキャッシュアルゴリズムは，ある時刻に新規コンテンツがまとめて追加される状況を想定していたが，実際の VoD サービスでは常に新規コンテンツが追加されており，その数は時間帯によって異なる [13]．新規コンテンツ数が異なると，通信量削減効果を最大化するハイブリッドキャッシュ割合も変わるため，ハイブリッドキャッシュの LRU/LFU 割合をその時間帯の新規コンテンツ数に応じて設定することで，通信量を削減効果を向上できる．具体的には，時間帯ごとに想定される新規コンテンツ数を設定しておき，新規コンテンツをキャッシュできるだけの LRU 容量を確保しつつ，LFU 領域を大きく設定するようなプリセットを切り替えればよい．

また，4 章で提案した色タグ情報に基づく分散協調キャッシュ制御は，動画アクセスに

表 5.1: 2 章で得られた知見のまとめ

番号	得られた知見	関連する章
(1)	動画配信ネットワークは動画配信サーバを最上位とする階層ネットワークで構成されている．階層ネットワークでは，下層のネットワークでアクセス頻度の高いコンテンツをキャッシュすることで，高い通信量削減効果を得られる．	3 章，4 章
(2)	動画アクセスはアクセス頻度の高いコンテンツに大きく偏っている．アクセス頻度の高いコンテンツを優先的に保持する LFU アルゴリズムが有効である．	3 章
(3)	動画アクセス傾向は短時間で大きく変化する．長期間のログ解析を必要とする LFU アルゴリズムよりも，LRU アルゴリズムの利用が有効である．	3 章
(4)	動画コンテンツは断続的に追加されており，キャッシュサーバのストレージ容量は相対的に小さくなる．複数のキャッシュサーバをまとめて実効キャッシュ容量を拡大する分散協調キャッシュの利用が有効である．	4 章

地域局所性がある場合は，ネットワークを地域ごとに分割して管理することで，通信量削減効果の向上が期待できる．たとえば，ヨーロッパのように隣接する地域で利用される言語が異なる場合，各国の母国語の動画再生数が多くなり，それ以外の動画の再生数は相対的に減少する．各地域ごとに色タグを設定するためのログ収集サーバを設置してコンテンツを別々に彩色するか，色タグ情報に地域情報を表現する ID を併記することで，地域的局所性のあるアクセスを効率よくキャッシュし，通信量削減効果の向上が期待できる．

表 5.2: 3 章および 4 章で得られた成果のまとめ

番号	3 章で得られた成果	関連する知見
(5)	LRU と LFU の異なるアルゴリズムを採用したキャッシュサーバでネットワークを構成することで，長期的にアクセス頻度の高いコンテンツと新規に追加されたコンテンツを効率よくキャッシュし，アクセス傾向が急激に変動する場合も高い通信量削減効果を維持できること．	(1)，(2)，(3)
(6)	LRU と LFU を組み合わせたハイブリッドキャッシュアルゴリズムが，人気コンテンツと新規コンテンツの 2 種類の動画アクセスを効率よくキャッシュし，アクセス傾向が急激に変動する場合も高い通信量削減効果を維持できること．	(1)，(2)，(3)
(7)	各キャッシュサーバの LFU 領域で異なるコンテンツを保持することで，人気コンテンツと新規コンテンツの 2 種類の動画アクセスを効率よくキャッシュし，アクセス傾向が急激に変動する場合も高い通信量削減効果を維持できること．	(4)，(6)
番号	4 章で得られた成果	関連する知見
(8)	最適化アルゴリズムで求めた通信量削減効果の大きいキャッシュ配置では，アクセス頻度の高いコンテンツほどネットワーク中のキャッシュサーバで重複して保持されていること．	(2)，(4)
(9)	サーバおよびコンテンツに色情報を付したタグを設定し，成果 (8) をもとにコンテンツ配置割合を調整することで，軽量で通信量削減効果の大きい分散協調キャッシュ制御を実現できること．	(2)，(4)，(8)
(10)	色情報に基づく分散協調キャッシュを構成するキャッシュサーバでハイブリッドキャッシュアルゴリズムを利用することで，アクセス傾向の急激な変化に追従できる，軽量な分散協調キャッシュを実現できること．	(7)，(9)
(11)	利用する色タグを工夫することで，階層ネットワークにおいても高い通信量削減効果を実現できること．	(1)，(10)

謝辞

本研究を通して熱心で根気強い研究指導を下さった，電気通信大学 大学院情報理工学研究科 情報・ネットワーク工学専攻の吉永努教授，同所属助教から株式会社フィックスターズに異動された吉見真聡氏に深く感謝申し上げます．論文審査を快く引き受けて下さった大学院情報理工学研究科の加藤聰彦教授，大坐畠智准教授，田野俊一教授，本多弘樹教授に心より感謝いたします．

本研究の一部は，電気通信大学と TIS 株式会社のキャッシュネットワークに関する共同研究課題「分散配置されたキャッシュサーバの効率利用による通信データ量の削減」の支援を受けて行われました．共同研究実施において産業の観点から助言いただいた TIS 株式会社 戦略技術センターの森元俊雄氏，村木暢哉氏，松井暢之氏，石橋靖嗣氏，油谷実紀氏に深く感謝申し上げます．また，共同研究の締結と進行をサポートして頂いたスーパー連携大学院コンソーシアム事務局の産形峰久氏，宇梶純良氏，松浦和子氏に心より感謝いたします．

本研究の一部は，TIS 株式会社 森元俊雄氏，村木暢哉氏，石橋靖嗣氏，油谷実紀氏，電気通信大学 産学連携センター 知的財産部門 加古彰子様，秀和特許事務所 弁理士 関誠之氏の協力のもと，特許出願されました．ご協力下さった皆様に心より感謝いたします．

最後に，英語論文の添削を快く引き受けて下さった Achraf Ben Ahmed 氏，研究室の運営をサポートしてくれた，城間隆行氏，ならびに吉永研究室，策力木格研究室の皆様に感謝申し上げます．

参考文献

[1] “The Zettabyte Era: Trends and Analysis,” June 2017. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html

[2] V.K. Adhikari, Y. Guo, F. Hao, V. Hilt, Z.L. Zhang, M. Varvello, and M. Steiner, “Measurement Study of Netflix, Hulu, and a Tale of Three CDNs,” IEEE/ACM Transactions on Networking, vol.23, no.6, pp.1984–1997, Dec. 2015.

[3] D.c. CoileÃąin and D. O’mahony, “Accounting and Accountability in Content Distribution Architectures: A Survey,” ACM Comput. Surv., vol.47, no.4, pp.59:1–59:35, May 2015.

[4] Z. Li and G. Simon, “In a Telco-CDN, Pushing Content Makes Sense,” IEEE Transactions on Network and Service Management, vol.10, no.3, pp.300–311, Sept. 2013.

[5] H. Lee, L. Duan, and Y. Yi, “On the competition of CDN companies: Impact of new telco-CDNs’ federation,” 2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pp.1–8, May 2016.

[6] F. Zhou, J. Liu, G. Simon, and R. Boutaba, “Joint Optimization for the Delivery of Multiple Video Channels in Telco-CDNs,” IEEE Transactions on Network and Service Management, vol.12, no.1, pp.87–100, March 2015.

[7] Z. Li and G. Simon, “In a Telco-CDN, Pushing Content Makes Sense,” IEEE Transactions on Network and Service Management, vol.10, no.3, pp.300–311, Sept. 2013.

[8] A. Balachandran, V. Sekar, A. Akella, and S. Seshan, “Analyzing the Potential Benefits of CDN Augmentation Strategies for Internet Video Workloads,” Proceedings of the 2013 Conference on Internet Measurement Conference, pp.43–56, IMC ’13, ACM, New York, NY, USA, 2013.

[9] G. Ibrahim, Y. Chadli, D. Kofman, and A. Ansiaux, “Toward a new Telco role in future content distribution services,” 2012 16th International Conference on Intelligence in Next Generation Networks (ICIN), pp.22–29, Oct. 2012.

[10] D.D. Vleeschauwer and D.C. Robinson, “Optimum Caching Strategies for a Telco CDN,” Bell Labs Technical Journal, vol.16, no.2, pp.115–132, Sept. 2011.

[11] Z. Wang, H. Jiang, Y. Sun, J. Li, J. Liu, and E. Dutkiewicz, “A k-coordinated Decentralized Replica Placement Algorithm for the Ring-Based CDN-P2P Architecture,” Proceedings of 2010 IEEE Symposium on Computers and Communications, pp.811–816, June 2010.

[12] W. Li, Y. Li, W. Wang, Y. Xin, and Y. Xu, “A Collaborative Caching Scheme with Network Clustering and Hash-routing in CCN,” 27th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp.2547–2553, Sept. 2016.

[13] H. Yu, D. Zheng, B.Y. Zhao, and W. Zheng, “Understanding User Behavior in Large-scale Video-on-demand Systems,” Proceedings of the 1st ACM SIGOPS/EuroSys European Conference on Computer Systems 2006, pp.333–344, 2006.

[14] H. Yin, X. Liu, F. Qiu, N. Xia, C. Lin, H. Zhang, V. Sekar, and G. Min, “Inside the Bird’s Nest: Measurements of Large-scale Live VoD from the 2008 Olympics,” Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, pp.442–455, 2009.

[15] Netmanias, “2013 Content Networking Trends - OTT, Global CDN and Operator CDN”. https://www.slideshare.net/Netmanias/netmanias20130904content-networking-trends2013

[16] L. Scott and C. Alistair, “Netflix OpenConnect & FreeBSD,” May 2013. https://people.freebsd.org/ scottl/Netflix-BSDCan-20130515.pdf

[17] B.M. Maggs and R.K. Sitaraman, “Algorithmic Nuggets in Content Delivery,” ACM SIGCOMM Computer Communication Review, vol.45, no.3, pp.52–66, July 2015.

[18] H. Li, L. Zhong, J. Liu, B. Li, and K. Xu, “Cost-Effective Partial Migration of VoD Services to Content Clouds,” 2011 IEEE International Conference on Cloud Computing (CLOUD), pp.203–210, July 2011.

[19] M. Calder, X. Fan, Z. Hu, E. Katz-Bassett, J. Heidemann, and R. Govindan, “Mapping the Expansion of Google’s Serving Infrastructure,” Proceedings of the 2013 Conference on Internet Measurement Conference, pp.313–326, IMC ’13, ACM, New York, NY, USA, 2013. http://doi.acm.org/10.1145/2504730.2504754

[20] Z. Feng, J. Li, o. Wu, and J. Zhi, “HD: A Cache Storage Strategy Based on Hierarchical Division in Content-centric Networking,” 2014 IEEE 17th International Conference on Computational Science and Engineering (CSE), pp.535–540, Dec. 2014.

[21] “総務省｜放送政策の推進｜4k・8k 放送の推進”. http://www.soumu.go.jp/menu_seisaku/ictseisaku/housou_suishin/4k8k_suishin.html

[22] “アップロードする動画の推奨エンコード設定 - YouTube ヘルプ”. https://support.google.com/youtube/answer/1722171

[23] “法人向け OCN サービス（IP バックボーン） | NTT コミュニケーションズ 法人のお客さま”. http://www.ntt.com/business/services/network/internet-connect/ocn-business/bocn/backbone.html

[24] 国立情報学研究所，“SINET4 学術情報ネットワーク,” 2015. http://w4a.sinet.ad.jp/document/sinet_doc/pamphlet/SINET4-2014_j.pdf

[25] 国立情報学研究所，“SINET5 について,” Nov. 2015. http://www.nii.ac.jp/userdata/openforum/PDF/2015/1_setsumeikai2015_sinet5_20151023.pdf

[26] X. Cheng, J. Liu, and C. Dale, “Understanding the Characteristics of Internet Short Video Sharing: A YouTube-Based Measurement Study,” IEEE Transactions on Multimedia, vol.15, no.5, pp.1184–1194, Aug. 2013.

[27] 君山博之，“IP ネットワークを使った高精細映像配信サーバシステムに関する研究,” PhD thesis，電気通信大学，Dec. 2010.

[28] X. Che, B. Ip, and L. Lin, “A Survey of Current YouTube Video Characteristics,” IEEE MultiMedia, vol.22, no.2, pp.56–63, April 2015.

[29] K.Y. Kamath, J. Caverlee, Z. Cheng, and D.Z. Sui, “Spatial Influence vs. Community Influence: Modeling the Global Spread of Social Media,” Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp.962–971, CIKM ’12, ACM, New York, NY, USA, 2012. http://doi.acm.org/10.1145/2396761.2396883

[30] L. Saino, “On the design of efficient caching systems,” Doctoral, UCL (University College London), Dec. 2015. http://discovery.ucl.ac.uk/1473436/

[31] G. Einziger and R. Friedman, “TinyLFU: A Highly Efficient Cache Admission Policy,” Proceedings of 2014 22nd Euromicro International Conference on Parallel, Distributed and Network-Based Processing, pp.146–153, Feb. 2014.

[32] Y. Zhou, L. Chen, C. Yang, and D.M. Chiu, “Video Popularity Dynamics and Its Implication for Replication,” IEEE Transactions on Multimedia, vol.17, no.8, pp.1273–1285, Aug. 2015.

[33] D. Lee, J. Choi, J.-H. Kim, S.H. Noh, S.L. Min, Y. Cho, and C.S. Kim, “LRFU: a spectrum of policies that subsumes the least recently used and least frequently used policies,” IEEE Transactions on Computers, vol.50, no.12, pp.1352–1361, Dec. 2001.

[34] B.H. Nesrine, M. Ruben, and M. Pascale, “ARMA based Popularity Prediction for Caching in Content Delivery Networks,” Wireless Days 2017, pp.113–120, Portugal, March 2017. http://wd17.inesctec.pt/program/.

[35] R.K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain, “Overlay Networks: An Akamai Perspective,” Advanced Content Delivery, Streaming, and Cloud Services, eds. by M. Pathan, R.K. Sitaraman, and D. Robinson, pp.305–328 ••, , 2014.

[36] C. Williamson, “On Filter Effects in Web Caching Hierarchies,” ACM Transactions on Internet Technology, vol.2, no.1, pp.47–77, Feb. 2002.

[37] N. Laoutaris, S. Syntila, and I. Stavrakakis, “Meta algorithms for hierarchical Web caches,” 2004 IEEE International Conference on Performance, Computing, and Communications, pp.445–452, 2004.

[38] K. Cho, M. Lee, K. Park, T.T. Kwon, Y. Choi, and S. Pack, “WAVE: Popularity-based and collaborative in-network caching for content-oriented networks,” 2012 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp.316–321, March 2012.

[39] “YouTube Now Sees 300 Hours Of Video Uploaded Every Minute,” Dec. 2014. http://www.tubefilter.com/2014/12/01/youtube-300-hours-video-per-minute/

[40] N. Choi, K. Guan, D.C. Kilper, and G. Atkinson, “In-Network Caching Effect on Optimal Energy Consumption in Content-Centric Networking,” Proceedings of 2012 IEEE International Conference on Communications, pp.2889–2894, June 2012.

[41] I. Tatarinov, A. Rousskov, and V. Soloviev, “Static caching in Web servers,” Proceedings of Sixth International Conference on Computer Communications and Networks, pp.410–417, Sept. 1997.

[42] S. Podlipnig and L. BÃűszÃűrmenyi, “A Survey of Web Cache Replacement Strategies,” ACM Comput. Surv., vol.35, no.4, pp.374–398, Dec. 2003.

[43] T. Shiroma, T. Nakajima, M. Yoshimi, and T. Yoshinaga, “An Efficient Cache Grouping Strategy for Multinode Cache Networks,” Proceedings of the 8th International Workshop on Autonomous Self-Organizing Networks, pp.295–298, Dec. 2015.

[44] A. Arteta, B. BarÃąn, and D. Pinto, “Routing and Wavelength Assignment over WDM Optical Networks: A Comparison Between MOACOs and Classical Approaches,” Proceedings of the 4th International IFIP/ACM Latin American Conference on Networking, pp.53–63, 2007.

[45] D.J.A. Welsh and M.B. Powell, “An upper bound for the chromatic number of a graph and its application to timetabling problems,” The Computer Journal, vol.10, no.1, pp.85–86, Jan. 1967.

[46] R. Fretcher, Practical Methods of Optimization, John Wiley & Sons, 2000.

[47] S. Hong, B.-J.B. Lee, C.-M.C. Yoo, M.-S. Do, and J. Son, “Comparative Study of Content-Centric vs. Content Delivery Networks: Quantitative Viewpoints,” Proceedings of The 10th International Conference on Future Internet, pp.35–40, 2015.

[48] E.W. Dijkstra, “A Note on Two Problems in Connexion with Graphs,” Numer. Math., vol.1, no.1, pp.269–271, Dec. 1959.

関連論文の印刷公表の方法および時期

論文誌（査読有り）

[1] Takuma Nakajima, Masato Yoshimi, Celimuge Wu, and Tsutomu Yoshinaga, “Color-based Cooperative Cache and its Routing Scheme for Telco-CDNs,” IEICE Transactions on Information and Systems, Vol.E100-D, No.12, pp. 2847–2856, Dec. 2017. （3 章および 4 章に関連）

国際会議（査読有り）

[2] Takuma Nakajima, Masato Yoshimi, Celimuge Wu, and Tsutomu Yoshinaga, “A Light-weight Content Distribution Scheme for Cooperative Caching in Telco-CDNs,” Proceedings of The Fourth International Symposium on Computing and Networking (CANDAR’16), 2016, pp. 126–132, Nov. 2016. （3 章および 4 章に関連）

研究会発表（査読無し）

[3] 中島 拓真, 岡田 浩希, 吉見 真聡, 策力木格, 吉永 努, “色タグ情報に基づく分散協調キャッシュおよびチャンク分割キャッシュ制御のプロトタイプの実装,” 電子情報通信学会技術研究報告, vol. 117, no. 314, pp. 9–14, Nov. 2017. （4 章に関連）

[4] 中島 拓真, 城間 隆行, 吉見 真聡, 策力木格, 吉永 努, “動画の人気変動に追従する異種キャッシュ混在ネットワークの検討,” 電子情報通信学会技術研究報告, vol. 115, no. 518, pp. 247–252, Mar. 2016. （3 章に関連）

[5] 中島 拓真, 吉見 真聡, 入江 英嗣, 吉永 努, “SDN による Web キャッシュサーバ間通信の効率化,” 電子情報通信学会技術研究報告, vol. 114, no. 302, pp. 31–36, Nov. 2014. （4 章に関連）