社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－DPS－134 (26)
2008－CSEC－ 40 (26)
2008／3／6

モバイル音楽共有システムJAMSにおけるキャッシュ管理方式

遠藤博樹　柴田浩明　渡部寿基　加藤由花

産業技術大学院大学 産業技術研究科
E-mail: *yuka@aiit.ac.jp*

複数の携帯端末で構成されるアドホックネットワークを利用し，場に応じて様々に変化するコンテンツを共有する音楽配信システムJAMSの研究を進めている．本稿では，JAMSを対象に効率的な音楽配信を実現するためのキャッシュ管理方式を提案する．提案方式では，キャッシュ配置の最適化を行わず，各ノードからの要求に従って適応的にキャッシュを配布することにより，JAMSに適したロバストなキャッシュ管理方式を実現する．本稿ではさらに，シミュレーション実験により，様々な環境下でのファイルへのアクセス要求に対する失敗率を測定し，提案方式が，場の変化を利用したシステムにとって適切な方式であることを示す．

A Cache Management Method for the Mobile Music Delivery System: JAMS

Hiroki Endo　Hiroaki Shibata　Toshiki Watabe　Yuka Kato

School of Industrial Technology, Advanced Institute of Industrial Technology
E-mail: *yuka@aiit.ac.jp*

We have proposed a music delivery system JAMS (JAMais vu System), which uses ad hoc networks with mobile devices for the localization services. In this paper, we focus on a cache management scheme for JAMS to deliver music files efficiently. The proposed scheme makes it possible to manage cache files robustly by using adaptive cache delivery according to each node request. In addition, we conduct a simulation experiment and obtain access failure ratio of music files under various system conditions. The simulation results indicate that the proposed scheme is suitable for the system using localization services.

1　はじめに

近年，モバイル端末の高機能化，モバイルネットワークの広帯域化に伴い，モバイルネットワークサービスに対する要求が高まってきている．我々は，日常生活に驚きや楽しみを感じてもらうことを目的に，モバイル端末を利用した音楽配信システムJAMS（JAMais vu[*1]System）の研究を進めている[1][2]．JAMSは，モバイル端末で構成されるアドホックネットワークによりサービスの局所性を実現し，利用するたびに受けられる内容の異なるサービスを提供するシステムである．近年，ユーザの状況や嗜好にマッチしたサービスを提供するためのコンテキストアウェアネス技術に対する注目が高まってきているが[3][4]，JAMSの発想はこれとは逆であり，偶発的に変化する状況を利用したサービスの提供がその目的となっている．

JAMSユーザは，あらかじめ自身の持つモバイル端末に音楽ファイルを蓄積しておき，通勤通学時やカフェでの待ち合わせ時間などに，同じ場に滞在する他のJAMSユーザ（不特定多数のユーザ）との間で各自が持つ音楽ファイルを共有する．このとき，ユーザが滞在する場は時間とともに変化していくので，場を構成するJAMSユーザも変化し，共有される音楽ファイルの種類も変化していく．その結果，滞在する場により受けられるサービスの内容が変化し，意外性，偶発性を持ったサービスの提供が実現する．

このように，モバイル端末から構成されるアドホッ

[*1] 「未視感」のこと．deja vu（既視感）の逆の概念で，見慣れたはずのものが未知のものに感じられること．

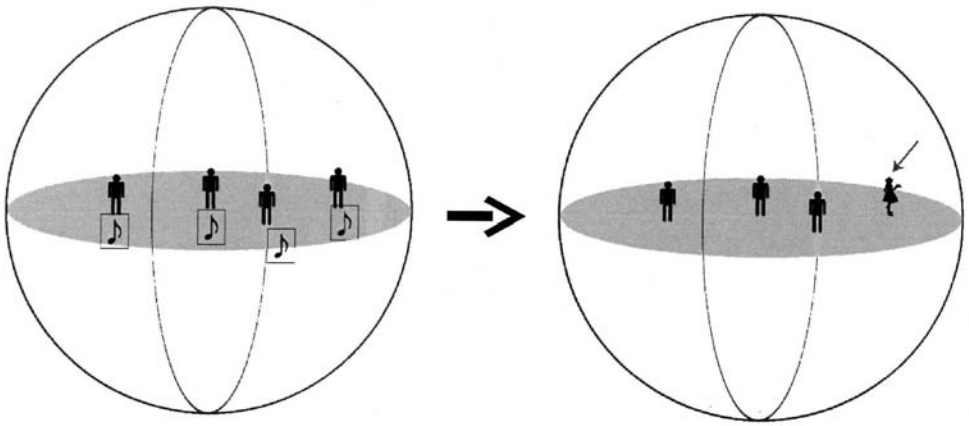

図 1　場のイメージ

クネットワーク上で音楽配信を行うことを考えると，1 台の端末に負荷が集中することを避けるために，ネットワーク上にキャッシュを配布し，効率的な音楽配信を実現する必要が出てくる．ファイル共有のためのキャッシュ配布方式としては，これまでも多くの提案が行われてきた．P2P ネットワーク上で効率的なキャッシュ管理を行う方式としては，Winny におけるキャッシュ管理方式 [5] や，木構造に基づく複製更新伝播法 [6] などがある．これらの方式では，リソースが潤沢な安定したノードの存在を仮定しており，ノードのモビリティに関する考慮も限定的であるため，JAMS への適用は困難である．アドホックネットワークを対象に，効率的にキャッシュを配布する方式も提案されているが [7][8]，ファイルのアクセス頻度を推定し，事前にキャッシュ配布を行う手法が考察対象となっており，コンテンツの局所性を利用した JAMS に適用することは難しい．

JAMS を対象としたキャッシュ管理方式を考察するためには，以下の 2 つの問題を解決する必要がある．

- コンテンツの局所性に関する問題：
  JAMS は局所的なネットワークを対象としているので，空間に存在するファイルは時々刻々と変化する．そのため，コンテンツの偏りに影響を受けない，ロバストなキャッシュ配布方式を実現する必要がある．
- 携帯端末に関する問題：
  CPU パワー，メモリ量，ディスク容量，バッテリなど，リソースに対する制約が多い．そのため，可能な限り簡易な処理で効率的にキャッシュを配布する必要がある．また，負荷の集中を避ける必要もある．

これらの問題を解決するために，本稿では，キャッシュ配置の最適化を行わず，各ノードからの要求に従って適応的にキャッシュを配布する方式を提案する．これにより，JAMS に適したロバストなキャッシュ管理を実現する．本稿では，さらに，シミュレーション実験を行い，提案手法の有効性を示す．

2　JAMS の概要

はじめに，本稿で考察対象とする JAMS の概要を述べる．JAMS では，コンテンツの局所性を利用したサービスを提供するので，本章ではまず，「場」による「コンテンツ」の違いについて説明する．その後，システムの構成を概観することにより，JAMS の機能を明らかにする．

2.1　コンテンツの局所性

JAMS では，コンテンツの局所性を「場」という概念で表現する．「場」のイメージを図 1 に示す．ここ

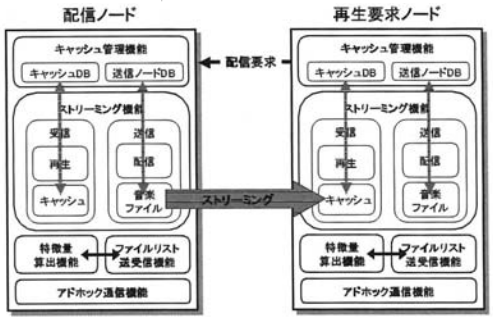

図 2　システムの構成

では，特定の範囲内にいるノード同士がネットワークを形成し（局所性），そのネットワークのことを「場」と定義している．JAMS では，この局所性は，アドホックネットワークにより実現する．ある限られた空間の中（電車の中，大学の教室など）で，各自が所有する携帯端末によってアドホックネットワークを構築し，そのネットワーク内で音楽ファイルを共有する．ここでの考察対象は，アプリケーションレイヤであり，下位レイヤのネットワークは IEEE802.11g 等により既に構築されているものと仮定する．

2.2 システムの構成

次にシステムの構成について説明する．JAMS のシステム構成を図 2 に示す．JAMS はピュア P2P 型ネットワークサービスであるため，全ての端末上に同一のアプリケーションが実装され，全ての端末が同等の立場で通信を行う．そのため，1 台の端末上に，音楽ファイルの送信機能と受信機能の両方が実装される（ストリーミング機能）．JAMS はまた，各端末が保有している音楽ファイルからノードの特徴量を算出し，ユーザに提示する機能，および同一の場に存在する端末から場の特徴量を算出する機能を有している．各ノードには，これらを実現するための機能（特徴量算出機能）が実装される（詳細については文献 [9] を参照）．

本稿で提案するキャッシュ管理方式としては，アドホックネットワーク上に音楽ファイルのキャッシュを配布する機能を有している．そのため，各端末には，自身が配布したキャッシュファイルを管理する機能（送信ノード DB）と，自身が保持するキャッシュファイルを管理する機能（キャッシュ DB）が実装される．

3 キャッシュ管理方式

3.1 設計指針

JAMS の特徴は，携帯端末による効率的な音楽配信を実現するために，アドホックネットワーク上に音楽ファイルのキャッシュを配布することである．キャッシュ管理方式の設計あたっては，以下の 3 つの設計指針を策定した．

- 各ノードが自律的にキャッシュを管理すること．
- 場の変化によりキャッシュの入れ替えが発生すること．
- 統計的な手法ではなく，適応的な手法を用いること．

一つめの指針は，携帯端末はリソースが限られているため，端末への負荷が少ない方式を採用する必要があることを考慮したものである．特に JAMS はピュア P2P 型ネットワークサービスであるため，全てのノードは対等であり，安定した一部のノードの存在を仮定することができない．そのため，あるノードに負荷が集中しないよう，キャッシュの分散管理が必要になる．二つめの指針は，キャッシュ配布方式がコンテンツの局所性に影響を与えないことを意図したものである．JAMS は滞在する場によって受けられるサービスの内容が異なるシステムなので，キャッシュの配布が場の特徴量に影響を与える方式は採用できない．三つめの指針は，場によってコンテンツの特徴が異なる状況で効率的なキャッシュ配布を実現するためのものである．場を構成しているファイルの種類は，時間，場所とともに変化していく．そのため，ファイルのアクセス頻度等を事前に予測することは不可能であり，統計的な手法は利用できない．

3.2 提案方式の特徴

設計指針に基づき，JAMS におけるキャッシュ管理方式を設計した．提案手法の特徴は，キャッシュが各ノード上で分散管理されること，タイムアウトにより一定時間アクセスされなかったキャッシュは消去されることの 2 点である．具体的な特徴を以下に示す．

- キャッシュの配布 については，能動的なキャッシュ配布を行わず，他のノードから音楽ファイルの配信を受けた場合にのみ，そのファイルを自ノードにキャッシュとして蓄積する．
- キャッシュの消去 については，あらかじめ設定されたタイムアウト時間を超えるまで配信要求がなかった場合には，自動的にキャッシュを消去する．
- キャッシュの置き換え については LRU を採用し，タイムアウトまでの時間が短いファイルから順に置き換えていく．
- キャッシュの管理 については，キャッシュの配信元が，キャッシュの配信先を管理する．その結果，ある音楽ファイルのキャッシュを保持するノードは，オリジナルファイルを持つノードを起点とした木構造を取ることになる．キャッシュの検索は，この木構造に対して再帰的に行われる．
- キャッシュの選択 については，あるノードへの配信要求数がしきい値（JAMS の場合は 3）を超えた場合に，要求ノードに対して自ノードが管理するキャッシュを紹介し，アクセスを分散させる．キャッシュを利用する順番は，オリジナルファイルを持つノードからの木構造の階層が少ない順としている．同じ階層に複数のキャッシュが存在する場合には，タイムアウトまでの時間が短いキャッシュから順に利用される．

音楽ファイルの配信を受けない限りキャッシュの配布は行われないので，キャッシュ管理に関しては，

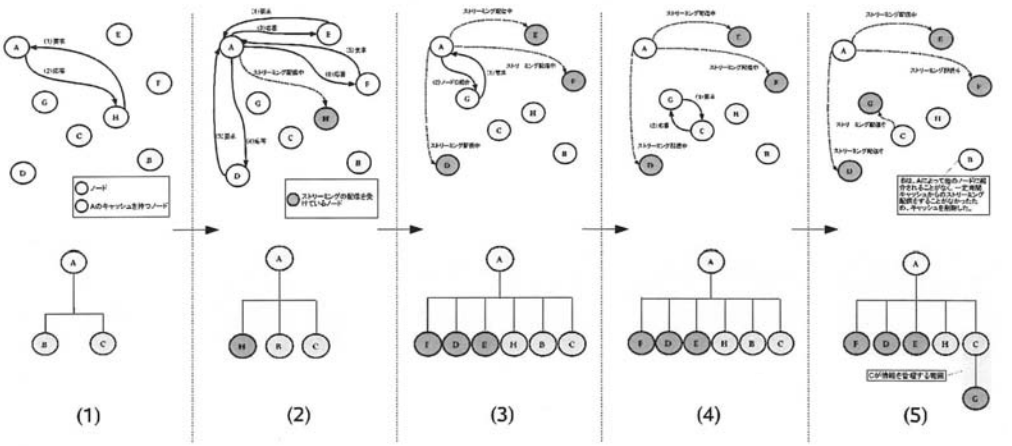

図 3　処理の流れ

ノードの参加による明示的な処理は発生しない．一方，タイムアウトの発生や，ノードの離脱等によりキャッシュが消滅した場合は，キャッシュ管理のための木構造の一部が崩れる．しかし，キャッシュ検索の起点は必ずオリジナルファイルを持つノードであるため，再び木を構築することが可能である．オリジナルファイルを持つノードが離脱した場合は，該当ファイルは検索対象から外れるので，その場に存在するキャッシュがタイムアウトによって消去されるのを待つだけでよい．

3.3　処理の流れ

本稿で提案するキャッシュ管理方式について，その処理の流れを図 3 に示す．ここでは，オリジナルファイルを持つノード A からキャッシュが配信される様子を示した．図の上部は，要求に従ってストリームが配信され，配信先ノードにキャッシュが蓄積されていく様子を示している．図の下部は，キャッシュ管理の木構造を示し，親ノードが管理しているキャッシュを線で結ぶことにより表現している．それぞれの処理について，以下に説明する．

1. B と C が A のキャッシュを保持している状態で，H が A に配信を要求してきた．B と C のキャッシュは A から直接配信されたものなので，キャッシュ管理の木構造は，A と B，C が直接結ばれる形になる．
2. H にストリーミングを配信している状態で，E, D, F が A に配信を要求してきた．H にはキャッシュが蓄積されているので，キャッシュ管理の木構造に，ノード H が追加された．
3. 配信要求に従い，E, D, F にストリーミングを配信する．この状態で G が配信を要求してきた．JAMS では，1 つのノードが同時に配信できるストリームの数を 3 としているので[*2]，A からの配信は行えない．そこで A は，自らが保持しているキャッシュのリストから，タイムアウト時間の最も短い C を G に紹介する．キャッシュ管理の木構造には，E，D，F が追加された．
4. A から C を紹介された G は，C に配信を要求する．C は配信可能な状態であるので，G に対してストリーミングの配信を開始する．
5. C から配信を受けた G には，キャッシュが蓄積されていく．このキャッシュは C が管理するので，キャッシュ管理の木構造において，G は C に直接結ばれる形で追加される．ここで，B は他のノードからの要求がないまま一定時間が経過したので，キャッシュは自動的に削除され，キャッシュ管理の木構造からも削除された．

4　性能評価

本稿では，提案手法の有効性を確認するために，キャッシュ配布機能を模擬するシミュレータを開発し，性能評価実験を行った．以下にその結果を示す．

4.1　実験の方法

ここでは，提案方式が，場の変化に依存せず，様々な環境下で効率的なキャッシュ配布を実現するロバストな方式であることを検証する．そのため，条件の異なる複数のシチュエーションを用意し，それらの条件の下での性能を測定することとした．性能評価尺度としては，ユーザの配信要求に対する失敗率

[*2] 同時配信数は携帯端末の性能や使用できる帯域等に依存するが，JAMS では PC を用いた事前実験を行い，この値を 3 とすることとした．携帯電話等を使用する場合でも，この程度の値が想定される．

表 1　実験条件

No.	シチュエーション	到着率（人/秒）	滞在時間（秒）	特徴
1	カフェ	0.0067	2700	ノード数が少なく滞在時間が長い.
2	駅のホーム	0.5	300	ノード数が多く出入りが激しい.
3	図書館	0.0017	5400	ノード数が少なく出入りが少ない.
4	電車	0.167	1200	ノード数が非常に多く出入りが激しい.

を利用した．失敗率は，以下のように定義される．

$$\text{失敗率} = \frac{\text{配信できなかった回数}}{\text{配信を要求した回数}}$$

本稿では，ノードごとにこの値を算出し，その平均値を求めた．JAMS においては，ストリーミングの同時配信数に制限があるため，効率的にキャッシュ配布を行わないと，配信に失敗する確率が高くなると予想される．

実験に使用したシチュエーションを表 1 に示す．到着率は表中の値を平均とするポアソン分布，滞在時間は表中の値を平均とする正規分布とした．ファイルへのアクセスを生成するノード，アクセス先ノード，ノード中のアクセスファイルは，全てランダムに決定した．実験では，比較のために，以下の 4 種類の方式に対して失敗率を測定している．

- 方式 A：本稿で提案する方式．キャッシュを配布したノードがキャッシュの管理を行う．
- 方式 B：キャッシュを利用しない方式．
- 方式 C：キャッシュの管理を，オリジナルファイルを保有するノードが一括で行う．
- 方式 D：事前にキャッシュを配布しておく方式．

方式 D では，場に存在するノードの中で，所有する音楽ファイルの傾向が似ているもの（具体的には曲のジャンルを利用）をグループ化し，ノード数の一番多いグループの中からランダムにキャッシュの送信元と送信先を選択した．そして，音楽ファイルの傾向に沿ったファイルをキャッシュとして配信した．これは，最も選択される確率の高いファイルをあらかじめキャッシュとして配布する状態を模擬してる．

4.2　実験の結果

それぞれの条件の下，6 時間分のシミュレーションを行った．実験の結果を図 4 に示す．ここでは，シチュエーションごとの失敗率の平均値（各ノードの失敗率の平均）と，各シチュエーションにおける平均値の標準偏差（σ）を示している．これらの結果から，全てのシチュエーションにおいて，提案方式（方式 A）の失敗率が一番低く，効率的なキャッシュ配布が行われていることが確認された．また提案方式は，シチュエーションの違いが失敗率に与える影響が小さく，ロバストな方式であることがわかった．

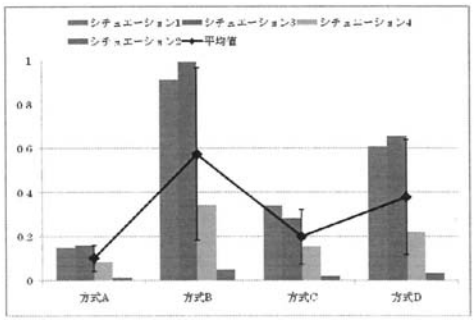

図 4　実験の結果

以下，実験結果の考察を行う．まず，今回設定したシチュエーションでは，場への滞在時間が長く，ノードの出入りが少ない 1 と 3 での失敗率が全体的に高くなっている．これは，ある程度の時間場に滞在することによりはじめて，人気のあるコンテンツへのアクセス集中が発生するためであり，人の出入りが激しい状態ではキャッシュの効果は相対的に低くなるためと考えられる．ただし，シチュエーション 4 のように，ファイルの種類に対してノード数が非常に多い場合にはキャッシュの効果が見込める．次に，各方式に対する考察を行う，まず，キャッシュの配布を行わない方式 B では，シチュエーション 1 と 3 において失敗率が 1 近くになり，著しく性能が劣化している．効率的な配信のためには，キャッシュ機構の実装は必須と考えられる．次に，ファイルへのアクセス要求頻度を推測し，あらかじめキャッシュを分配しておく方式 D では，推測の精度が分配の精度に影響を与えるため，JAMS のようにシチュエーションが変化し，その傾向の予測がつかないシステムでは，顕著な効果が得られない．またこの方式では，シチュエーションによる失敗率のばらつきが大きくなっていることもわかる．キャッシュの管理をオリジナルノードで行う方式 C では，キャッシュへのアクセスが失敗した時点で次のキャッシュの紹介が行われなくなるため失敗率が高くなり，その結果，シチュエーションによる失敗率の分散も，提案方式と比較して大きくなっている．

つまり提案方式では，シチュエーションに依存せ

ず，アクセスの集中したファイルのキャッシュが効率的にネットワーク全体に配布され，ファイルアクセスへの失敗率を抑制していると考えられる．

5 まとめ

本稿では，アドホックネットワークを利用したモバイル音楽配信システム JAMS を対象に，効率的な音楽配信を実現するためのキャッシュ管理方式を提案した．提案方式では，キャッシュ配置の最適化を行わず，各ノードからの要求に従って適応的にキャッシュを配布することにより，JAMS に適したロバストなキャッシュ管理を実現した．さらに，シミュレーション実験により，様々な環境下でのファイルへのアクセス要求に対する失敗率を測定し，提案手法の有効性を確認した．

JAMS には，場の特徴量をユーザに提示することによりサービス参加への動機付けをするという特徴があるが，今後，この特徴量の提示がユーザの行動に与える影響を分析し，それがキャッシュ配布方式の性能にどのような影響を与えるかについて，検討を進めていく予定である．

参考文献

[1] 柴田浩明, 富澤智, 遠藤博樹, 加藤由花. コンテンツの局所性に着目した P2P 型音楽配信システム. 情報処理学会 DPS ワークショップ 2007, pp. 37–42, 2007.

[2] 柴田浩明, 富澤智, 遠藤博樹, 渡部寿基, 大城裕史, 加藤由花. 場の特徴量を利用した意外性のあるモバイル音楽共有システム. 情報処理学会 DPS 研究会, Vol. DPS-133, pp. 25–30, 2007.

[3] D. J. Corbett and D. Cutting. AD LOC: Collaborative Location-based Annotation. *IPSJ Journal*, Vol. 48, No. 6, pp. 2052–2064, 2007.

[4] Y. Nakanishi, K. Takahashi, T. Tsuji, and K. Hakozaki. iCAMS: A mobile communication tool using location and schedule information. *IEEE Pervasive Computing*, Vol. 3, No. 1, pp. 82–88, 2004.

[5] 金子勇（編）. Winny の技術. アスキー, 2005.

[6] 渡辺俊貴, 原隆浩, 木戸裕樹, 中通実, 西尾章治郎. P2P ネットワークにおける木構造に基づく複製更新伝播法. 情処論, Vol. 48, No. 2, pp. 527–538, 2007.

[7] 林秀樹, 原隆浩, 西尾章治郎. アドホックネットワークにおけるトポロジ変化に適応した複製の再配置. 情処論, Vol. 47, No. 1, pp. 2–14, 2006.

[8] Liangzhong Yin and Guohong Cao. Supporting cooperative caching in ad hoc networks. *IEEE Trans. on Mobile Computing*, Vol. 5, No. 1, pp. 77–89, 2006.

[9] 富澤智, 大城裕史, 加藤由花. モバイル音楽共有システム JAMS における特徴量表示方式. 情報処理学会 DPS 研究会, Vol. DPS-134, , 2007.