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

2006－DPS－126（1）
2006－CSEC－32（1）
2006／3／16

ファイルシステムを利用した透過性・汎用性の高い連続メディア情報のリモートアクセス手法

嘉藤将之[1]　山井成良[2]　岡山聖彦[2]　久保亮介[3]　松浦敏雄[4]

[1]岡山大学 大学院自然科学研究科
[2]岡山大学 総合情報基盤センター
[3]シャープ株式会社
[4]大阪市立大学 大学院創造都市研究科

概　要

動画像・音声などの連続メディア情報のリモートアクセス手法として，NFS を用いる方式が提案されている．しかし，この方式は QoS 制御機能がないため，広域ネットワークなど帯域が十分でない環境では適用できないという問題があった．この問題に対処するため，我々の研究グループではファイルアクセス API を拡張して用いる方式を提案してきたが，適用の範囲が限定されるという問題があった．そこで，本稿では QoS 制御可能なファイルシステムを用いた連続メディア情報のリモートアクセス手法を提案する．本手法では，データの低品質化を行う場合，ファイルシステムがメディア情報の形式に沿ったパディングを行い，見かけ上のデータサイズが変わらないよう補完することによりファイルアクセス API を用いる方式の問題に対処する．また，提案方式の実装を行い，本方式が狭帯域ネットワーク経由でアクセスした場合でも有効であることを示す．

A Remote Access Method of Continuous Media Data with High Transparency and Portability Using File Systems

Masayuki Kato[1], Nariyoshi Yamai[2], Kiyohiko Okayama[2],
Ryosuke Kubo[3], and Toshio Matsuura[4]

[1]Graduate School of Natural Science and Technology, Okayama University
[2]Information Technology Center, Okayama University
[3]SHARP Corporation
[4]Graduate School of Creative Cities, Osaka City University

Abstract

NFS-based services have been proposed for transferring continuous media data. However, on a network environment like WAN that does not have enough bandwidth, NFS-based services are not suitable since they do not provide QoS function. To solve this problem, our research group proposed a remote access method of continuous media data by modifying file access APIs. However, the coverage of this method is limited. In this paper, we propose a remote access method of continuous media data using file system which provides QoS function. To solve the problem of the method with file access APIs, when a quality of media data need to be deteriorated, the file system of our method supplements data with padding according to the format of continuous media data so that file size of continuous media data will not be changed. This paper also discuss a design and implementation of the proposed method and shows that our method is effective even on narrow band network environment.

1　はじめに

最近，計算機の高機能化やネットワークの高速化に伴い，動画像や音声などの連続メディア情報をネットワークを介してアクセスして表示するストリーミングアプリケーションが広く利用されるようになってきた．これらのストリーミングアプリケーションの多くは，限られたネットワーク帯域で QoS(Quality of Service) を保証しながらデータを伝送するため，アプリケーション毎に独自に工夫されたデータ形式や伝送プロトコルを採用している．MPEG[1] や RTP[2] など一部のものを除き，一般にこれらの間には互換性がないため，ネットワークを介してメディアデータにアクセスするにはサーバ側のサーバプログラムと同種のプレイヤをクライアント側でも使用する必要があり，特定のプレイヤでなければメディアデータにアクセスできないという問題があった．

一方，連続メディアを扱うプレイヤの多くは，例えば MPEG のような標準的な形式のファイルをローカルディスクから読み込んで表示・再生する機能を有している．この点に着目した連続メディア情報のリモートアクセス手法として，NFS[3][4] を直接用いた方法 [5]（以下，NFS 方式）や NFS プロトコルをプロクシで変換する方法 [6]（以下，NFS プロクシ方式）がある．NFS 方式では OS が介在して通信機能を提供することにより，サーバ側のデータをクライアント側ではローカルファイルとしてアクセスできるため，サーバ側で MPEG などの標準形式でメディアデータを格納しておけばクライアント側ではその形式のローカルファイルを表示・再生できる任意のプレイヤをそのまま用いることができる．しかし，NFS 方式ではファイルシステムの性質上，データの正確性のみが重要視され実時間性などの QoS が考慮されないため，LAN のように高品質な通信を行えるようなネットワーク環境でしか利用できない．NFS プロクシ方式では，イントラネットのように十分な帯域を有するものの伝送遅延やジッタを無視できないようなネットワークでも NFS 方式を利用できるように，クライアントと同じ LAN 内にプロクシを配置してプロトコル変換を行い，サーバに連続的にメディアデータを送信させる手法が提案されている．しかし，この手法も帯域が十分でないネットワーク環境では利用できない．

上記の問題に対処するため，我々の研究グループではこれまでファイルアクセス API を拡張して用いる方法 [7]（以下，ファイルアクセス API 方式）を提案してきた．ファイルアクセス API 方式は，ファイルオープン時における先読み機能，ファイル読み込み時における実時間性保証機能，代替データのキャッシュ無効化などをファイルアクセス API に組み込むことにより，NFS 方式や NFS プロクシ方式の利点を活かしながら QoS 制御を行うことを可能にしている．しかし，ファイルアクセス API 方式はダイナミックリンクライブラリを利用しているプレイヤ以外には適用できず，データの補完方法が十分でなかったため適用の範囲が限定されていた．

このように，従来の連続メディア情報のリモートアクセス手法は特定のアプリケーションに依存したり，あるいは QoS 制御機能がなく利用可能なネットワーク環境が限定されたり，適用可能なプレイヤが限定されたりするなどの点で問題があり，いずれも不十分であるといえる．

そこで，本稿では QoS 制御可能なファイルシステムを用いた連続メディア情報のリモートアクセス手法を提案する．本手法では，ローカルディスク上のファイルが表示・再生できるプレイヤであれば任意のものを利用でき，NFS 方式や NFS プロクシ方式，ファイルアクセス API 方式の利点を活かしながら QoS 制御を行うことが可能となる．

2　ファイルシステムを用いたリモートアクセス

前章で述べたように，NFS 方式や NFS プロクシ方式はサーバ上のメディアデータをクライアントはローカルファイルとしてアクセスできるため，特定のアプリケーションに依存しないという特長を有する．さらに，これらの方式にはファイルアクセス API 方式のように適用範囲が限定されるという問題がない．そこで，本稿ではこれらの特長を活かした連続メディア情報のリモートアクセスを実現するため，QoS 制御機能を有するファイルシステムを提案する．

2.1　ファイルシステムの利用

NFS 方式において QoS 制御機能を実現するには，NFS プロクシ方式のようにクライアント側にプロクシを設置し，その中に同機能を組み込む方法が考えられる．しかし，QoS 制御機能を導入するとデータの正確性が犠牲になるため，ネットワーク帯域が不十分な期間に一度低品質（不正確）なデータがプロクシから OS に渡されるとこのデータがキャッシュされ，ネッ

トワーク帯域が回復した後に再びアクセスしても低品質なデータしか得られないという問題が生じる．

そこで，本稿では open，read といったファイルシステムにおけるファイル処理を拡張する手法を提案する．ファイルシステムレベルで QoS 制御を行えば，各プレイヤのファイルへのアクセス方法に関わらず，全てのプレイヤに適用可能である．

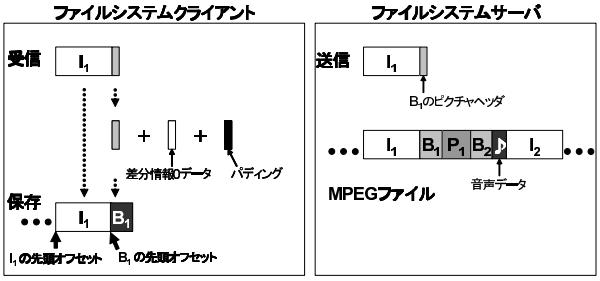

図 1: 解像度優先方式でのデータの低品質化

2.2 ファイルシステムの設計方針

2.2.1 データの先読み

通常，広域ネットワークの特性による映像や音声の乱れは，クライアントのプレイヤ内にあるバッファのサイズを大きくすることで，ある程度は回避することができる．しかし，既存のプレイヤに対してこのような変更を加えることは一般には困難である．

そこで，提案手法ではファイルシステム内に大きなバッファを設け，open や seek などのファイル処理が行われるとき，あるいは read などの処理によりバッファが枯渇しそうになったときに先読みを行うようにする．これによりプレイヤや OS に変更を加えることなく，広域ネットワークの特性による映像や音声の乱れを抑えることが可能になる．

2.2.2 データの低品質化

データの先読みにより，バッファリングを行なったとしても，ネットワークの利用可能帯域が長期的に十分でない場合は，バッファの枯渇が防げない場合がある．このような場合，従来のファイルシステムの概念では，データを誤りなく読み書きすることが重要であるが，連続メディア情報を表示・再生する場合はデータの正確性を多少犠牲にしても，実時間性を保証することが望ましい．そこで，提案手法では，データの先読みが要求期限までに間に合わない場合はサーバ側からの送信データの品質を低下させ，必要なネットワーク帯域を軽減することにより実時間性を保証するようにする．ここで，データの低品質化が生じた場合，プレイヤに低品質の部分を切り詰めて提供してしまうと，データのファイル内でのオフセットやファイル全体のサイズが見かけ上変わってしまうという問題が生じる．つまり，プレイヤがデータを読み込む際に本来要求しているデータが得られず，再生に不具合が生じることが予想される．そこで，データの低品質化が生じた場合には，データ構造を保つようにパディングなどを行うことによりオフセットの変更が生じないようにする．

低品質データの提供方法としては，解像度を優先して重要度の低いフレームを欠落させる方法（以下，解像度優先方式）とフレームレートを優先して解像度を落す方法（以下，フレームレート優先方式）が考えられる．

解像度優先方式でファイル形式として MPEG を用いる場合の例を図 1 に示す．MPEG ファイルは，I ピクチャ，P ピクチャ，B ピクチャの 3 種類の画像データと音声データによって構成される．ピクチャの中で特に重要なものが，他のピクチャ生成の際にも基準となる I ピクチャである．また，P，B ピクチャは前もしくは前後のピクチャとの差分情報である．ネットワーク帯域が十分である場合には，全てのデータが先読みされてバッファに格納される．ここで，I_1 の前までのデータ転送が完了した時点でネットワークの帯域不足によりバッファの枯渇が発生する状態になったと想定すると，これ以降はサーバは I ピクチャ（I_1，I_2 のデータ）と音声データ（♪のデータ）のみを伝送し，他のピクチャ（B_1，P_1，B_2）はデータを欠落させる．B_1 を欠落させる場合，サーバは I_1 を送信した後，B_1 のピクチャのデータ部分を欠落させ，ピクチャヘッダのみを送信する．クライアントではピクチャヘッダを受信すると，ピクチャのデータ部分に前のピクチャとの差分情報がないというデータ（差分情報 0 データ）を格納し，さらにパディングを行うことで元の B_1 ピクチャのサイズにして，データのオフセットが変わらないように保存する．パディングには，MPEG で映像と音声を多重化するための構造であるパケットのヘッダのスタッフィングバイトを利用する．

フレームレート優先方式では，サーバ上に予め高解像度と低解像度の 2 種類のデータを用意する．ファイル形式として MPEG を用いる場合の例を図 2 に示す．MPEG では GOP(Group Of Picures) という単

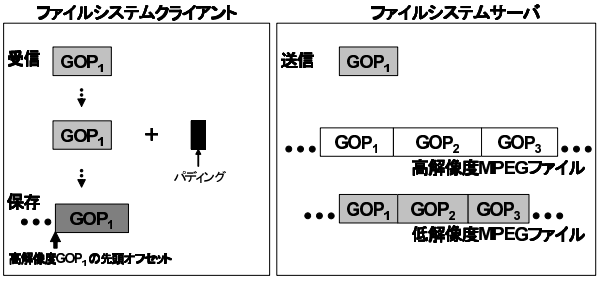

図 2: フレームレート優先方式でのデータの低品質化

位でランダムアクセスを可能にしている．ネットワーク帯域が十分である場合には，高解像度のデータが先読みされバッファに格納される．ここで，GOP_1 の前のデータ転送が完了した時点でバッファの枯渇が発生する状態になったと想定すると，これ以降はサーバは低解像度のデータを転送する．クライアントは低解像度データを受信すると，解像度優先方式と同様にパディングを行い，高解像度のデータサイズと同じデータサイズにして高解像度データと同じオフセットに保存する．

以上のような方法の採用により，提案手法ではオフセットやファイルサイズを変更することなく，実時間性を保証しながらメディアデータをファイルとして提供することが可能になる．

2.2.3 代替データのキャッシュ無効化

多くのファイルシステムでは，同一データに対するアクセス高速化のため，キャッシュが設けられている．したがって，NFS プロクシ方式に単純に QoS 制御機能を組み込むと，2.1 節の冒頭で述べたように，ネットワーク状態が長期的に悪化した際には低品質な代替データがキャッシュされることになる．したがって，その後ネットワーク状態が改善されたとしても，同じデータに再びアクセスする際にはキャッシュされた代替データしか得られない可能性がある．

このような問題を避けるため，提案手法ではファイルシステムでキャッシュ制御を行い，代替データへのアクセスが予想される場合にはその部分だけキャッシュを無効化して高品質データの再取得を試みる．なお，キャッシュされた高品質データに対してアクセスがあった場合には，キャッシュが有効に機能し，サーバからのデータ再取得は行わない．

3 提案手法の設計・実装

独自のファイルシステムをゼロから作り上げることは困難であり，汎用性にも欠けることから，多くのプラットフォームで利用可能でファイル共有機構として幅広く用いられている Samba[8] に QoS 制御機能を組み込むことで，QoS 制御可能なファイルシステムを提案する．ここで，NFS プロクシ方式を拡張する方法が考えられるが，Samba は Windows 系 OS でのファイル共有サービスで利用されている CIFS[9] に対応しており，Windows 系 OS で容易に利用可能であることから上述の方法を採用した．

以下では，提案するシステムの詳細について述べる．

3.1 試作システムの構成

提案システムでは，NFS プロクシ方式と同様にクライアントと同じ LAN 内にプロクシを導入して前章で述べた様々な機能を組み込み，QoS 制御可能なファイルシステムとして機能するように設計した．

Samba では，サーバ上に SMB[1] メッセージの処理を行うプログラム smbd が存在する．本システムでは，クライアントがサーバに透過的にアクセスできるように，Samba プロクシプログラム smbp(samba-proxy) に smbd で使われる SMB プロトコルを中継する機能を導入する．また，サーバ上に新たに read-server スレッドと read-server sender スレッドを導入し，Samba プロクシはクライアントからの SMB メッセージのうち read メッセージだけを分離し，read-server スレッドとの間で QoS 制御を行いながら，read-server sender スレッドから送られるマルチメディアデータを受信する．read などのファイル操作以外のメッセージはサーバの smbd に単に中継する．サーバ・Samba プロクシ間の通信には，フロー制御，伝送誤り発生時の再送処理，データの順序保証などを行うため，これらの機能を全て有する TCP を用いたが，UDP や RTP などのプロトコルを用いることも可能である．クライアント・Samba プロクシ間の通信には従来の SMB プロトコルを用いるため，クライアントから見ると Samba プロクシは Samba サーバと等価であり，クライアントの OS やプレイヤに変更を加えずにアクセスすることが可能となる．

試作システムは以下のように動作する．

smbd は open 要求を受け取ると，read-server に先

[1]Samba では，クライアントは SMB (Server Message Blocks) と呼ばれるメッセージの交換により，サーバの資源にアクセスしている．

読み開始メッセージと要求のあったファイル名を通知する．read-server はこの通知を受け取ると read-server sender に同様の通知を行い，read-server sender は指定されたファイルデータの送信を開始する．smbp では一定量のデータがバッファに格納された時点で open の応答をクライアントに返し，さらにバッファが満杯になるまで先読みを続ける．クライアントから read 要求が送られると，smbp は先読みしたデータあるいは先読みが間に合わなかった場合には低品質データを read 応答として返す．

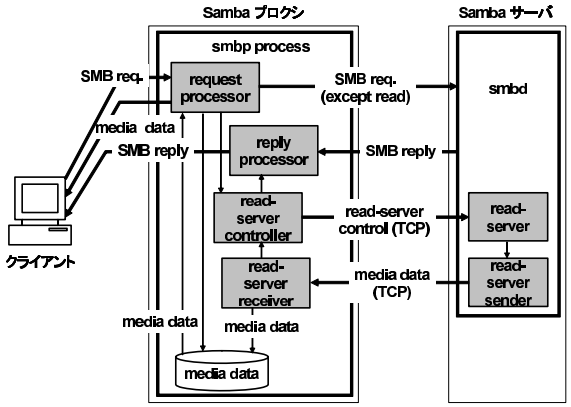

図 3: プロクシの内部構造とメッセージの流れ

3.2 プロクシの内部構成と動作

次に Samba プロクシの内部構造を述べる．図 3 に Samba プロクシの内部構造と，Samba プロクシを使用した場合のメッセージの流れを示す．

Samba プロクシの内部には，主にクライアントからの SMB メッセージのサーバへの中継を担当する request processor スレッド，サーバからの SMB メッセージのクライアントへの中継を担当する reply processor スレッド，データの受信状況と読み込み状況を管理し QoS 制御を担当する read-server controller スレッドおよびサーバから送られてくるデータを受信しデータ格納を担当する read-server receiver スレッドの 4 種類が存在する．

read-server controller では，クライアントが次に要求するデータのオフセットと先読み済みのデータのオフセットを管理し，データ受信が間に合わない場合は read-server にデータの低品質化メッセージを送信する．read-server はこのメッセージを受信すると，read-server sender を制御し，これ以降は低品質のデータを送出する．但し，その場合でもデータ構造を保持するために必要なデータは常に送るようにする．二通りあるデータ低品質化方法の選択については，メディアファイル毎にそれぞれ適用する方法を記述したファイルをサーバが予め保持するようにした．

read-server sender は指定されたファイルのデータをオフセット付きで送出する．read-server receiver はこれを受信すると指定されたオフセットの位置にそのデータを格納する．データの低品質化が行われた場合には，高品質のデータとサイズとオフセットが変わらないように保存する．ここで，解像度優先方式では差分情報 0 データが必要となるが，試作システムでは，同一の静止画を複数用意し，それらから同一フレームで構成される動画を作成し，その動画からピクチャデータを抽出することにより作成した．また，read-server receiver は低品質データのオフセットとサイズを保存し，再度同一ファイルに対してアクセスが発生した場合には，前回再生した品質よりも高品質で再生できるように低品質部分のデータを read-server に要求する．

3.3 動作試験

提案手法の有効性を検証するため，試作システムの動作試験を行った．この試験ではサーバに同一内容の MPEG-1 ファイル (サイズ 56.3MB，再生時間 4 分 38 秒，ビットレート 1.45Mbps）を 2 つ用意し，データの低品質化方法を一方は解像度優先方式，もう一方はフレームレート優先方式とした．フレームレート優先方式でデータの低品質化を行う際に必要となる低解像度ファイルとしては，上記と同一内容の MPEG-1 ファイル（サイズ 12.6MB，ビットレート 0.15Mbps）を用いた．また，プレイヤとして Windows Media Player を用いた．サーバの追加スレッド，プロクシは C 言語で実装した．

動作試験では，利用可能なネットワーク帯域が不十分な状況下での有効性を検証するため，サーバ側で dummynet[10] の帯域制限機能を利用し，帯域を 5.0Mbps，4.0Mbps，3.0Mbps，2.0Mbps，1.5Mbps，1.0Mbps，0.75Mbps，0.5Mbps の 8 通りに設定した場合について各 10 回再生を試み，正常に再生できるかどうかを確認した．また，比較の対象として通常の Samba を用いた場合についても同様の試験を行った．

動作試験の結果を以下に述べる．

まず，帯域を 5.0Mbps，4.0Mbps，3.0Mbps，2.0Mbps に制限した場合は，提案手法，通常の Samba を用いた場合ともに毎回映像は途切れることなく正常な速度

で再生された．これは試験に使用した MPEG-1 ファイルのビットレートが 1.45Mbps であることから十分な帯域であると考えられ，妥当な結果であると言える．

帯域を 1.5Mbps に制限した場合，通常の Samba を用いると，毎回において音声が途切れ，映像がゆっくり再生された．このように再生された理由としては，1 回の read 処理で得られたデータのみがプレイヤで再生され，プレイヤが正常な速度で再生できる期限内に次の read 処理の応答が返ってきていないものと思われる．試験に用いた MPEG-1 ファイルのビットレートと比較すると，十分な帯域であると考えれられるが，read 処理以外の SMB メッセージの交換により read 処理に遅れが生じたものと推測される．一方，提案手法では 10 回の試行でそれぞれ発生するタイミングは異なるもののデータの低品質化が発生し，正常な速度で再生された．すなわち，解像度優先方式ではコマ送り状態で再生され，フレームレート優先方式では低解像度で再生された．いずれの方式においても音声は途切れず，正常に再生された．

これ以降，帯域が狭くなるにつれて低品質で再生される時間は増加するものの，解像度優先方式では 0.75Mbps，フレームレート優先方式では 0.5Mbps に帯域を制限した場合まで正常な速度で再生された．

次に，低品質データのキャッシュ無効化機能を確認する実験を行った．この実験では，まず帯域を 0.75Mbps に制限した状態で再生を行い，多くの低品質データが格納される状況にした後，帯域を 5.0Mbps に緩和した状態で再び再生を行った．その結果，2 回目の再生では低品質データのキャッシュが無効化され，1 回目の再生では低品質で再生された部分も含めて高品質で再生された．

さらに，他のプレイヤでも再生できるかどうかを確認する実験を行った．使用したプレイヤは Real Plyer と DivX Player の 2 種類である．試験の結果，両方のプレイヤにおいて Windows Media Player を用いた場合と同等の結果が得られた．

以上の結果により，提案手法は帯域が不十分な環境においても低品質な映像で実時間性を保ちながら再生を続けることが可能であり，多くのプレイヤへの対応が期待できるため，有効かつ実用的であると言える．

4 まとめ

本稿では，QoS 制御可能なファイルシステムにより，リモート上の連続メディア情報のアクセス手法を提案した．また，試作システムの動作試験を通してその有効性を確認した．

今後の課題としては，データの低品質化を行った場合のメディア情報の表示・再生品質を人の主観で評価を行い，それに基づきバッファリングやデータの低品質化を開始したり，解除したりする閾値の検討を行うことが挙げられる．これに加えて，実時間性保証機能の改良も今後行いたい．試作システムにおける解像度優先方式とフレームレート優先方式の一方だけでは不十分な場合，両方を組み合わせることにより必要なネットワーク帯域を更に減少させ，提案手法を適用できるネットワーク環境を増やすことや，パディング方法を改良し，データの低品質化を行った場合の映像の品質を向上させることも重要な課題である．

謝辞

本研究の一部は平成 15〜17 年度科学研究費補助金 (基盤研究 (C)(2)，課題番号 15500048) の補助を受けている．

参考文献

[1] 藤原洋，“最新 MPEG 教科書，” アスキー，1994.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications,” RFC3550,2003.

[3] Sun Microsystems,Inc., “NFS:network file system protocol specification,” RFC1094,1989.

[4] B. Callaghan, B. Pawlowski, and P. Staubach, “NFS version3 protocol specification,” RFC1813,1995.

[5] 大村猛，廣田照人，中西正典，岡秀幸，“ビデオサーバソフトウェア—Video Network Server—その設計と実装評価，” 電子情報通信学会論文誌 (D-II)，Vol.J79-D-II,No.4,pp.626–633,1996.

[6] 山井成良，浪平大輔，安倍広多，下條真司，松浦敏雄，村山孝三，“広域ネットワークにおける NFS を用いた連続メディアデータアクセス方式，” 情報処理学会論文誌，Vol.42,No.2,pp.178-187,Feb.2001.

[7] 嘉藤将之，山井成良，岡山聖彦，久保亮介，松浦敏雄，“ファイルアクセス API を用いた連続メディア情報のリモートアクセス手法，” 情報処理学会研究報告，2004-DPS-119,Vol.2004,No.89,pp.85-90 (2004)

[8] Samba, http://us1.samba.org/samba/

[9] CIFS, http://www.snia.org/tech_activities/CIFS

[10] dummynet, http://info.iet.unipi.it/~luigi/ip_dummynet/