

多対多通信を支援するオーバレイネットワーク技術

電子情報学専攻 川田 雅人

1 はじめに

広帯域かつ遍在する接続環境を取得したユーザは、さらに SIP やコンテキストウェア支援による拡張性の高いブートストラップ手段を利用することで、1対1かつ固定メディアの電話型コミュニケーションを超えて、多対多かつ多様なメディア・端末を通じた双方向コミュニケーションを行うことが予想される。

このようなコミュニケーションの利用形態からの要求条件は (1) 2人以上のコミュニケーションを行うための効率的なメディア配送網 (2) コミュニケーションの参加・脱退が他の参加者に影響を与えずに自由に行われること (3) メディアの変換やミキシング出力、複製能力のある端末を経由する柔軟なルーティングができること (4) 使用可能な帯域 (通信コンテキスト) とユーザの行動・嗜好・注目 (ユーザコンテキスト) に対応し、コミュニケーションの品質を高めるレート制御・メディア変換を行うこと、である。さらに端末の多様性に対して (5) 端末の処理能力・機能 (デバイスコンテキスト) を考慮して (1) - (4) を行うことが望ましい。

以上の要求に対し、同時に双方向コミュニケーションをする人数は高々数十人程度であることを条件に、(1) ~ (3) を満たす手法としてオーバレイネットワークが挙げられる。オーバレイネットワークは、特定のサービスやアプリケーションに従った IP 層より上位層で柔軟に展開するために、複数のホストによって論理的に構築されるネットワークのことである。オーバレイネットワークでは、経路の耐障害性 [1] や QoS 提供 [2] などの目的に特化したネットワーク構築が行われ、特にマルチキャスト機能を持つものはアプリケーションレベルマルチキャスト (以下、ALM) と総称される (Narada[3] など多数)。

本章では、オーバレイネットワーク技術を基本とし (4) 音声と階層符号化映像を用いたコンテキスト適応型レート制御 (5) 端末能力を考慮したオー

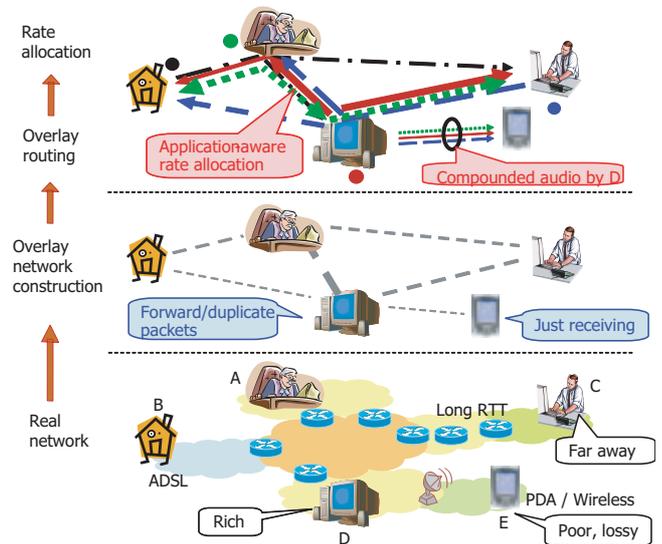


図 1: 動作概要

バレイネットワーク構築、の機能を持つ CORTH (Collaborative Overlay Routing and Transmission scheme considering Heterogeneity) を示す。

2 概要

CORTH の目的は大きく分けて 2 つある。1 つは、先に触れた 2 人以上のリアルタイムコミュニケーション環境を実現するオーバレイネットワークとルーティング、もう 1 つは (4) のコンテキスト要求をメディア品質割当にマッピングすること、である。

CORTH の動作過程を図 1 を用いて説明する。CORTH では多対多環境で効率的なネットワーク管理をするために、まず端末同士を双方向接続する共通のオーバレイネットワークを構築する (図 1 下中)。ここで、各端末は端末間の遅延・帯域を測定して自律的に接続リンクを決定し、さらに各端末の接続リンク数と端末能力を対応させることでオーバレイネットワーク上の転送役の重み付けを行う。次

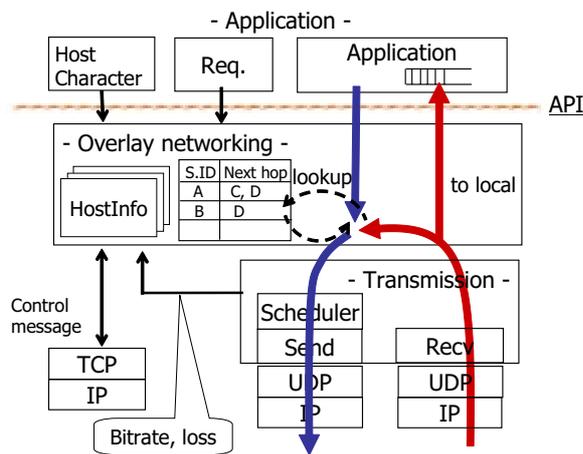


図 2: CORTH の構成

に、構築されたオーバーレイネットワーク上で各ソースに関する経路を決定し、データを転送する（図 1 上）。経路は、遅延を抑えつつ端末の特性を考慮した決定を行う。そして最後に、各リンクが輻輳制御を行ってリンク帯域を決定するとともに、そのリンクを通過する複数のソース同士で帯域の資源割当を行い、アプリケーション要求に従った通信品質割当を実現する。

ここで本稿で用いられる端末能力とは、アプリケーション品質に影響を与えずに処理が可能なスループットとし、CODEC 等に要する処理能力や使用可能帯域の制約によるロスや転送遅延が生じない上限値を用いる。端末能力を算出するにあたり通信データ量と処理量、およびそれぞれの品質や負荷を測定し、処理中のスループットで品質劣化が生じない値は下限として、劣化が起きた値は上限として利用する。

CORTH の機能要素は大きくオーバーレイネットワーク機能とデータ伝送制御機能の 2 つに分けられ、アプリケーションを含めて相互に制御情報の交換を行って目的とする機能を達成する。図 2 に各端末に備えられる CORTH の全体構成を示す。CORTH は UDP(TCP)/IP 層とアプリケーション層の間に存在するミドルウェアとして実現される。

CORTH を利用するアプリケーションは、ライブラリまたはローカルプロキシを介してデータの送受信、命令交換を行い、TCP や階層符号化、CODEC パラメータの変更などのレート制御機能を持つことが求められる。またユーザのコンテキスト（UI 操作から得られる注目や動作）やデバイスコンテキス

ト（ミキシング機能、CPU 能力など）を取得する API を備えている。レート制御部は、接続先ホストと TFRC[4] のような TCP に親和性のあるレート制御機能を持つ UDP コネクション（以降パイプと呼ぶ）を導入し、遅延やビットレートなどの通信コンテキストを正確かつリアルタイムに取得するインタフェースを上位に設ける。パイプを流れるデータは、パイプのスループットに応じた量で伝送される。アプリケーション部とレート制御部に挟まれたオーバーレイルーティング部は、インタフェースを通して集めたホスト情報、コンテキスト情報に基づき、オーバーレイネットワークの構築とルーティングテーブル（データ伝送先ホスト）を決定する。

上記の構造は、基本的に参加する全てのホストが持ち、ホストの参加/消滅に対する耐障害性のために自律分散制御が行われる。ただし、PDA など伝送能力や表示能力が小さく、しかも移動性が強いホストに対しては、自律分散制御における役割を小さくし、制御に必要な計算量・処理量を節約するとともに、表示能力や移動特性に応じた特別な支援処理を行うことを想定している。

3 アルゴリズム

本ルーティング手法では、ホストの機能（ミキシング、フォーマット変換）・伝送能力の異種性を考慮すること、多対多コミュニケーションのために複数ビデオの ALM 経路を保持すること、ユーザの嗜好が大きいビデオを優先制御することを目標とする。

3.1 オーバーレイネットワークの構築

まずホストがセッションに参加するとき、セッションの情報を元に SIP などのブートストラップを行い、参加ホストの IP アドレスを取得する。このホストと制御メッセージを交換してさらに他のホスト情報を受け取り、ホストリスト（図 2 中の HostInfo）を作成する。各ホストは HostInfo 中のホストとの遅延を測定し、さらに遅延の小さいいくつかのホストに対してテストデータを伝送し、使用可能帯域を取得する。これらの情報は定期的に監視・更新され、これらのメトリックとホストの伝送能力を元に、伝送能力に比例した接続本数のオーバーレイリンクを確立

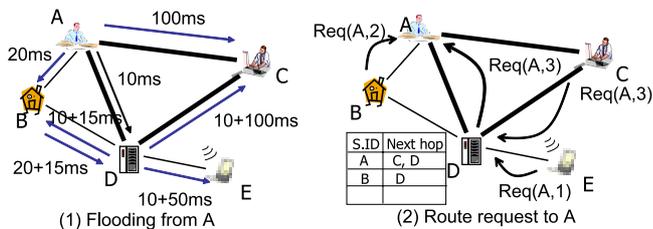


図 3: 経路メッセージプロトコル

する．オーバーレイネットワーク構築例は図 1 中のようになる．これにより，ホストの伝送能力と使用可能帯域が許す限りにおいては中継ホストを経由せずに直接送るため，遅延の小さい伝送網が出来上がる．

3.2 オーバレイルーティング制御

オーバーレイネットワーク上の経路は，その上で一意な送信ホスト ID やデータ ID 毎に構築され，そのデータを欲するホストと ID 毎に要求される経路条件（遅延最小，必要帯域など）に基づいて決定される．ソースホストは，データの ID を持った制御パケットをオーバーレイネットワーク上でブロードキャストする．データを中継するホストは，送られてきた上流リンクの遅延を制御パケットに書き込んでから全ての下流リンクに転送する．こうすることにより，各ホストはソースホストからの遅延を上流リンクごとに持つことができる．同様に，制御パケットにリンクの帯域情報やホストの機能情報を次々に付加して下流に転送する．図 3 左に，ホスト A のブロードキャスト例を示す．

ホストが当該データを受信する場合，遅延と映像品質（伝送レート）の双方を重視し，ある遅延の条件内で伝送能力の高い，あるいは特別機能が存在する上流ホストを選択して受信要求クエリーを送信する．クエリーにはユーザが当該データに対して持つ要求値を付与する．この経路決定ポリシーを本稿では Widest Shortest Path Tree (WSPT) と呼ぶ．図 3 右に，WSPT に従ってホスト A にリクエストが届く例を示している．クエリーを受け取った上流ホストは，自分がクエリーにある ID のソースでない場合，経路表に当該 ID の転送先として下流のホストと userReq を登録した後，WSPT に従ってさらに上流リンクにクエリーを転送する．これらを繰り返してクエリーがソースホストにたどり着いたときに経

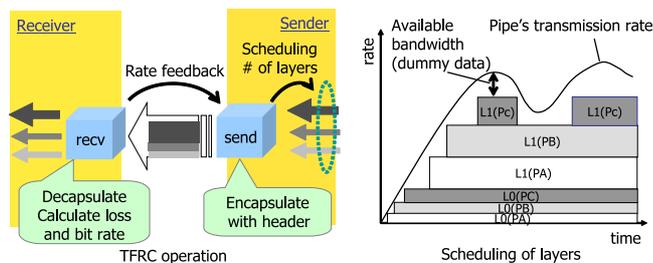


図 4: レート制御

路が完成し，図 2 の RoutingInfo に保存される．この経路表に従い，自ホストがエンコードしたビデオと受信した映像のパケットに対し，経路表の lookup をして次の送り先ホストの RateControl 部に渡す．

3.3 リンク毎のレート制御

本手法ではオーバーレイリンクのレート制御として，TCP に親和性の高い UDP 上のレート制御を実現する TFRC[4] を用い，さらに TFRC で得られた送出レートを達成するために階層符号化されたビデオの送出量を調整するスケジューラを用いる．

図 4 左に TFRC の送信側・受信側の動作イメージを示す．送信側では時刻やシーケンス番号などを持つヘッダでビデオデータをカプセル化して送信し，受信側ではヘッダを解析してパケットロス率の検出とその時刻を記録し，RTT とパケットロス率から受信レートを算出する．算出されたレート値は制御チャネルを通じて送信側にフィードバックされ，送信側ではパケットの送出 (sendto) 間隔を制御して算出した伝送レートを実現している．

図 4 右に伝送レートの時間変化と階層化されたビデオデータのスケジューリングの様子を示す．スケジューラはまず，OverlayRouting 部から得られる一つ以上の送信ビデオ ID，各階層の平均ビットレート，およびユーザの嗜好情報から，(1) 全てのビデオの基本階層（音声含む）は最大優先度を持つ (2) 拡張階層は下位から順に嗜好に比例した優先度を持つ，の条件を満たすように優先度を与える．そして，優先度順に割り当てた各階層のビットレートの総和が，TFRC 送信側から得られる送信レートを超えないように，送信する階層をスケジュールする．送信レートが次の階層を送信するほどは小さくなく，パケット送出タイミングに送るべきデータがないとき

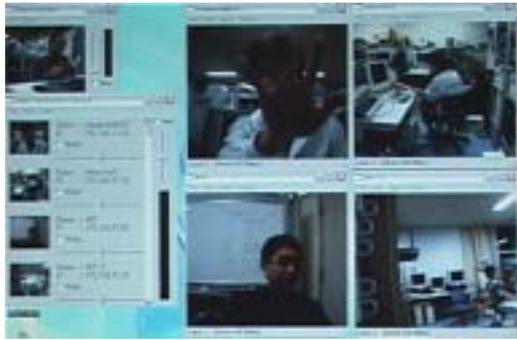


図 5: スクリーンショット

は、ビデオと同一パケット長のダミーデータを送信し、受信側で廃棄する。

4 実装評価

本章では、提案するホスト構成とアルゴリズムをビデオチャットシステムに実装した結果を示す。本システムは Visual C++ 6.0 with DirectX SDK 8.0 上で実装されている。スクリーンショットを図 5 に示す。

実装評価のために用いた実験ネットワークを図 6 に示す。SOHO を含めた日本の各地とアメリカとを結んだビデオ会議を想定し、Dummynet ルータによって遅延・帯域を設定した。この実験環境で作成されたオーバーレイネットワークを図 7 に示す。矢印の向きと太さはオーバーレイ上のビデオデータの経路とその伝送レートを視覚的に示したものであり、実験環境における遅延と帯域の異種性に適応したオーバーレイネットワークの構築と階層ビデオの伝送が行われていることが確認された。

5 むすび

本稿では、ビデオコミュニケーションの多様な利用形態とホストの環境に対し、複数のホストやビデオフローが分散協調的に適応するためのホスト構成と連携制御アルゴリズムを示した。今回は特に多対多コミュニケーション対応とホストの環境異種性に関して実装評価を行い、オーバーレイルーティングと各リンクの輻輳制御およびスケジューリングによって帯域とユーザの嗜好に動的に対応することにより、コミュニケーション品質を高められることを示した。

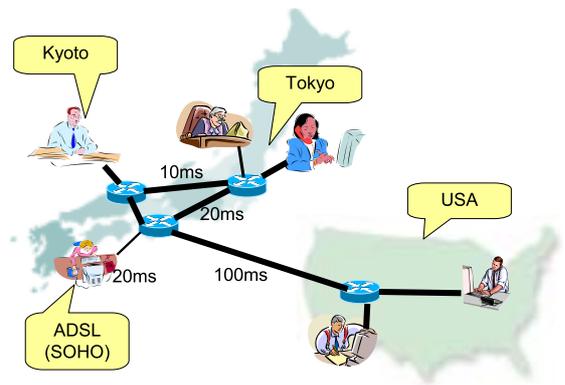


図 6: 実験物理トポロジー

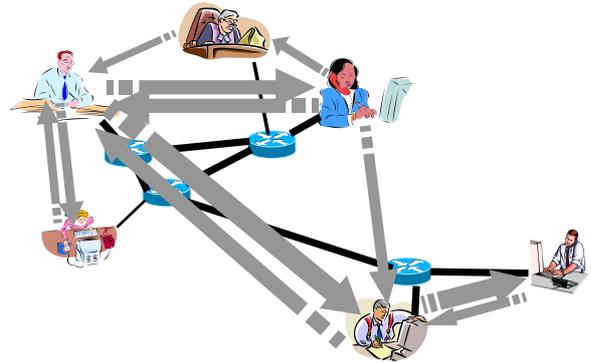


図 7: 構築されたオーバーレイネットワーク

参考文献

- [1] D.Andersen, H.Balakrishnan, F. Kaashoek, and R. Morris, " Resilient Overlay Networks, "In Proceedings of SOAP2001, 2001.
- [2] L. Subramanian, I. Stoica, H. Balakrishnan, and R.H. Katz, " OverQoS: Offering Internet QoS Using Overlays ", In Proceedings of HotNets-I, 2002.
- [3] Y.Chu, S.Rao, S.Seshan, H.Zhang, " Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture, "In Proceedings of SIGCOMM2001, Aug. 2001.
- [4] M. Handley, S. Floyd, J. Pahnay, and J. Widmer, " TCP Friendly Rate Control (TFRC): Protocol Specification, "RFC 3448, Proposed Standard, Jan. 2003.