自律広域マルチキャスト方式 Flexcast における ソケット API の設計と実装*

井上 武† 谷 誠一郎† 湊 真一† 宮崎 敏明†
ＮＴＴ 未来ねっと研究所‡

1. はじめに

インターネットは，www（world wide web）や電子メールといったクライアントサーバモデルを採用したアプリケーションを中心に，世界中に普及した．クライアントサーバモデルは１対１の通信を基本としている．今後は，ＴＶ放送のように１対多通信を行うアプリケーションがインターネットに新たなサービスをもたらすと期待されている．このような通信モデルに基づくサービスを「放送型ストリーミング」サービスと呼ぶことにする．

クライアントサーバモデルによって放送型ストリーミングを実現した場合，サーバに過大なトラフィックがかかることとなる（図１左）．放送型ストリーミングには，ネットワーク内で分岐を行い冗長トラフィックを削減するマルチキャスト技術が適している（図１右）．しかし，代表的なマルチキャスト技術である IP マルチキャストは，経路上の全ルータが IP マルチキャストに対応していなければならないという制約があるため，普及が遅れている．また，ルーティングプロトコルのスケーラビリティにも問題がある．

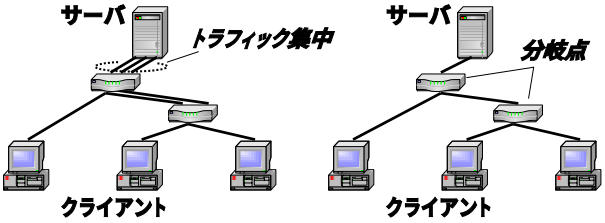

図１クライアントサーバ（左）とマルチキャスト（右）による放送型ストリーミング配信

我々はこれらの問題を解決するために，ユニキャストアドレスのみを用いたマルチキャスト型ストリーミングプロトコル - Flexcast[1]を開発した．本稿では，Flexcast に基づく放送型ストリーミングサービスを対象とし，ソケット抽象化モデルの提案を行う．また，広く普及している BSD ソケット[2]と同様の API（Application Programming Interface）を採用し，既存アプリケーションの Flexcast への移植性を高めた．

2. Flexcast プロトコル

Flexcast は受信要求とストリーム配信の２種類のメッセージを用いる．どちらも UDP パケットである[1]．受信要求の宛先ポート番号は決められている．Flexcast では，ストリーム送信者のアドレスと，送信者内で一意に定められる ID 番号の組によって，ストリームを識別する．この組を「チャネル」と呼び，（送信者，ID 番号）のように丸括弧で括って表記する．

図２を用いて Flexcast の動作を説明する．図中のパケットには，宛先，送信元及びチャネルを記す．まず，受信要求処理について説明する．ストリーム受信者 C1 は，チャネルを記載した受信要求を送信者 S に宛てて送信する．受信要求は S に向けてルーティングされる．この経路上に中継者が設置されていると，宛先ポート番号を検査し，受信要求用に割り当てられた番号であれば自分宛でなくても捕獲し，要求元を配信表に追加する．図では中継者 B が C1 及び C2 からの受信要求を捕獲し，配信表に追加している．また，中継者は要求元に成り代わって受信要求を送信する．中継者 A も，B と同様に受信要求を捕獲し，代理送信を行う．受信要求は，各受信者や中継者から定期的に送信され，一定時間要求が途絶えると，配信木から切り離される．このようにしてストリーム配信木が構築される．図では A，B を分岐点とする配信木が構築されている．

次に，ストリーム配信について説明する．S は，S にとっての要求元である A にのみストリームを送信する．A は B と C3 に転送する．同様に，B は C1 と C2 に転送する．このようにして，ストリームが配信木に沿って各受信者まで配信される．なお，ストリームの送信元アドレスは常に送信者のアドレスとなるため，受信者からはクライアントサーバモデルと同様に見える．また，Flexcast はユニキャストアドレスのみを用いるため，IP マルチキャストパケットをルーティングしないネットワークにおいても，マルチキャストによるストリーム配信を行うことが可能である．

3. 放送型ストリーミングにおけるソケット抽象化

クライアントサーバモデルでは，IP アドレスとポート番号の組からなるソケットアドレスを通信対象として指定する．このような抽象化は，BSD ソケットを初めとして広く利用されている．BSD ソケットの TCP API では，クライアントはサーバソケットアドレスを指定し，connect 関数によってサーバに接続する．また，サーバは bind 関数によって待ち受けソケットアドレスを指定し， listen 関数によってクライアントからの接続要求を受信し，

---

* Socket API Design and Implementation for Flexcast – Wide Area Autonomous Multicast
† Takeru Inoue, Seiichiro Tani, Shin-ichi Minato, Toshiaki Miyazaki
‡ NTT Network Innovation Labs.

[1] TCP による実装も開発中である．

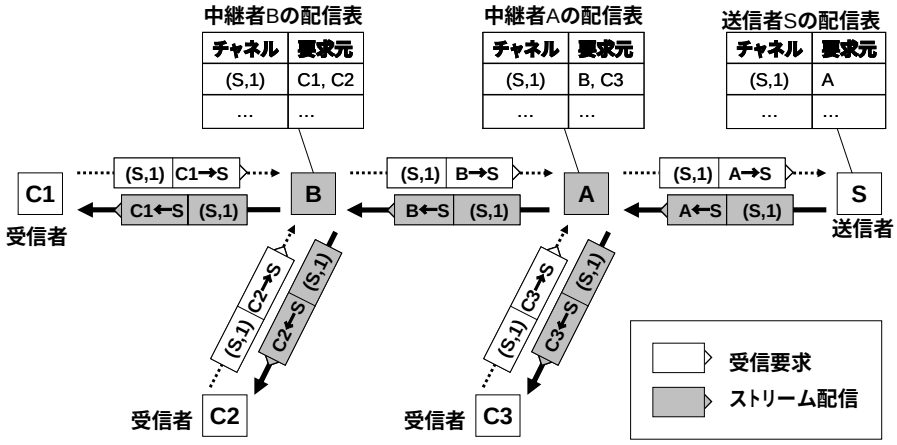

図2 Flexcast プロトコル

accept 関数によってクライアントに接続するためのソケットを得る．

これに対し，Flexcast では中継者が介在するため，送受信者同士が直接通信を行うとは限らないし，その必要もない．そこで，ソケットアドレスに代わりチャネルを通信対象として指定する．また，受信要求の送受信処理をバックグラウンドで行い，Flexcast ソケットを利用するプログラマから隠蔽する．これは，TCP における Ack 送受信処理の隠蔽と同様の効果があり，Flexcast プログラマは受信要求処理を意識することなく，ストリーム送受信に関するコードのみを記述することができる．この結果，放送型ストリーミングにおいても BSD ソケットと同様の抽象化を行うことが可能となり，クライアントサーバモデルで開発されたアプリケーションをそのまま移植することが可能となる．

4. Flexcast ソケット API の実装

Flexcast ソケット API を実装したユーザ空間ライブラリを作製した[2]．実装は Linux2.4 上で C 言語により行った．リスト１を用いて API を説明する．受信者はソケットを開くと（flexcast_socket），送信者アドレスと ID 番号を指定したチャネルによって配信木に接続する（flexcast_connect）．一方，送信者は受信を待ち受けるチャネルを指定し（flexcast_bind），受信要求を受信し（flexcast_listen），ストリーム配信に用いるソケットを得る（flexcast_accept）．

flexcast_connect 関数を発行すると，そのソケットを閉じるまでバックグラウンドで受信要求が定期的に送信され続ける．また，flexcast_listen 関数を発行すると，バックグラウンドで受信要求の受信処理が行われ，要求元が配信表へ追加あるいは配信表から削除される．flexcast_send 関数を呼ぶと，全要求元にストリームデータが送信される[3]．このとき，送信者は要求元の増減を意識しなくてよい．

また，送受信者の動作を組み合わせて中継者を作成することもできる．中継者は，受信要求を待ち受け，配信ソケットを取得すると供に，要求チャネルに対して connect を行う．自分宛ではない受信要求の捕獲は，OS のフィルタリング機能によって行う．

受信要求を受け付けるときに認証などの特別な処理を行いたい場合には，コールバック関数を指定することができる．flexcast_getsockopt 関数によって個別の要求元を取得することも可能である．

5. おわりに

Flexcast に基づく放送型ストリーミングサービスを対象としたソケット抽象化モデルを提案し，実装を行った．アプリケーション開発者は Flexcast の詳細なプロトコルを意識することなく容易にプログラムを実装することができ，ソフトウエア生産性の大幅な向上が期待される．今後は，転送能力などの性能測定を行う予定である．

```
/* 受信者 */
  /* チャネルとして IP アドレスと ID 番号を指定 */
  struct flexcast_channel servch = { serverip, serverid };
  int sockfd = flexcast_socket(AF_INET, SOCK_DGRAM, 0);
  /* バックグラウンドで受信要求送信開始 */
  flexcast_connect(sockfd, &servch, sizeof(servch));
  /* 上流からストリームを受信 */
  while ( flexcast_recv(sockfd, buff, bufflen) > 0 ) {
    /* 受信したストリームデータを処理 */
  }

/* 送信者 */
  struct flexcast_channel servch = { INADDR_ANY, 0 };
  int listenfd = flexcast_socket(AF_INET, SOCK_DGRAM, 0);
  flexcast_bind(listenfd, &servch, sizeof(servch));
  /* バックグラウンドで受信要求待ち受け開始 */
  flexcast_listen(listenfd, 20);
  for (;;) {
    /* 配信用ソケットを取得 */
    int connfd = flexcast_accept(listenfd, NULL, NULL);
    /* 全要求元へストリーム送信 (必要なら繰り返す) */
    flexcast_send(connfd, buff, bufflen);
  }
```

リスト１ Flexcast ソケット API を用いた擬似コード

参考文献

[1] S. Tani et al., “Adaptive Stream Multicast Based on IP Unicast and Dynamic Commercial Attachment Mechanism: An Active Network Implementation,” IWAN2001.

[2] W. R. Stevens, “UNIX Network Programming Networking APIs: Sockets and XTI,” Prentice-Hall, Inc., 1998.

[2] ライブラリ内部では BSD ソケットによって通信している．

[3] flexcast_send 関数は要求元がゼロになるとエラーを返す．