第 3 部

マルチキャスト通信

第 1 章

はじめに

WIDE マルチキャスト通信ワーキンググループでは，近年重要になってきている IP マルチキャストを用いた通信技術について研究・実験を行なっている．

広域分散システムにおいては，電子掲示板システムや一般に公開されたファイルの転送といったことにより多くの情報が共有されている．また，実時間的な会話型応用ソフトウェア，メーリングリストによる電子メールのグループへの配送といった情報共有や情報配送が非常に活発に行なわれており，インターネットのような広域分散システムにおいて非常に重要な機能となっている．

音声や画像を扱うアプリケーションを用い, 会議などをマルチキャストで中継する実験がインターネット上で活発に行なわれているが，既存のマルチキャスト経路制御技術では，対応が十分とはいえず，今後のネットワークの規模の拡大には対応できない．

マルチキャスト通信ワーキンググループでは，広域性のある経路制御機構，また，実時間性を必要とする音声・画像通信のための，信頼性のある遅延のばらつきの少ない通信のための資源予約機構，広域のマルチキャスト型通信媒体としての衛星通信機構について研究を行なっている．さらに，マルチキャスト通信を有効に利用した 1 対多型の通信に適したアプリケーションとして，ニュースやメーリングリストの配送，FTP アーカイブの転送といったものの研究も行なっている．

本稿では，1993 年度，マルチキャスト通信ワーキンググループで行なった次のような研究・実験について報告する．

- 共有木によるマルチキャスト経路制御機構 SnowCrystal の提案
- 資源予約に基づく信頼性のあるトランスポートプロトコルの提案
- スーパーバード B を用いた衛星通信媒体によるマルチキャスト通信実験および広域マルチキャスト通信媒体を考慮した経路制御プロトコルの提案
- MBONE の運用と管理，運用ガイドの作成

第 2 章

共有木に基づくマルチキャスト経路制御と資源予約

2.1 はじめに

現在音声や画像を扱うアプリケーションを用い, 会議などをマルチキャストで中継する実験がインターネット上で盛んに行なわれている. このような実験はマルチキャストの可能性を全世界に知らしめた一方で, 現在の技術の未熟さを露呈した. 例えば, 現在のマルチキャスト経路制御技術では, ネットワークの規模が拡大したときに, ルータが保持する情報が巨大となり機能しなくなるなどである.

広大なインターネットでは，広域な範囲で効率よく機能する広域性のある経路制御を行ない, また, 遅延のばらつきを押さえるために資源予約を行なわなければならない. そこで, 本章では共有木を用いることによって広域性のある経路制御を実現し, 同時に資源予約を行なう制御プロトコル SnowCrystal を提案する.

2.2 現在のネットワーク層における問題点

この節ではインターネットのネットワーク層を形成している, インターネットプロトコルの転送方式について述べ, 次に, 現在マルチキャストのために用いられている経路制御技術に関して, 考察を行なう.

2.2.1 転送方式

現在インターネットで用いられている, マルチキャストのためのネットワークプロトコルは, IP マルチキャストと呼ばれている. これは, IP パケットがマルチキャストルータにおいて, 必要ならば複製され分岐するのみであり, 転送方式は全く IP と同じである.

IP は, 全体のスループットを最大にすることを目的とした, best effort 型のネットワークプロトコルである. すなわち, ネットワーク資源になんらかの予約を行なって通信するのではなく, 資源の使用状況にかかわらず通信を行なう. よって, 回線などに競合が少ない場合はパケットがスムーズに転送され, 輻輳が起こった場合は, パケットは転送を待た

されるか, 捨てられる. 資源の競合状態は, 動的に変化するので, ある通信における個々のパケットが到達に有する時間は一定でない.

これは連続性が不可欠である音声通信において致命的な品質劣化を招く. 事実, MBone を使ったリモート会議の実験では, 音声が途絶える, 割れる, 聞きとれないなどの問題が生じる [1]. また, 以前のフレームに対し差分情報を送る動画通信では, パケット到達の遅れからそのパケットを落とすと, 画像を復元することが不可能となり, 映像は乱れる. このように実時間性を求めているアプリケーションにとって, なんらかの方法でネットワーク資源を予約し, 遅延時間を一定に保つことが必要不可欠である [2].

現在考案されている代表的な資源予約プロトコルに, ST-II(Experimental Internet Stream Protocol)[3] と RSVP(Resource ReSerVation Protocol)[4] がある. ST-II は送信者指向であり, メンバーの状況や資源予約状況を送信者が管理する. よって, メンバーが増大した時に送信者への負担が増加するため, 広域性の面で難点がある. また, 資源予約の要求が受信者から出された場合, 要求はいったん送信者まで配送され, 送信者から受信者に向けて資源が予約される. これは, 他の受信者が共に利用可能な資源をすでに予約していた場合も, 必ず送信者まで要求を届ける必要があり応答性がない.

例えば, 図 2.1において, a が送信者, e と f が受信者であるとする. ここで, a と e 間の回線はすでにグループ G に対して資源が予約されており, f が新たにグループ G へ加わることを考える. f からの資源予約の要求は, ゲートウェイ d, c, b を経由して a に到着する. ここで, a は f へ向けて資源予約を行なう. 回線 1, 2, 3 はすでに G へ予約されているので, 最終的に 5 の回線が予約されて, 資源予約の過程が終る. このように, すでに予約された回線上を送信者に向かって要求が送られるのは無駄であるし, 応答が遅くなる.

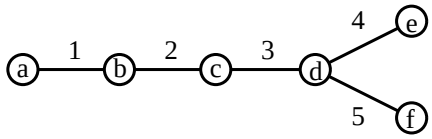

図 2.1: 送信者指向と受信者指向の資源予約

これに対し, 受信者から送信者に向けて資源予約を行なえば, この冗長性を排除できる. 例えば, 図 2.1において, f から a へ向けて資源予約が行なわれた場合を考えてみる. まず, 回線 5 が予約され, 次に, 要求が d へ届きた時, 既に d から a までの回線は予約されているので, 予約の過程はここで終る. このように, 受信者指向の資源予約は応答性に優れているし, また, 送信者がメンバーを管理しないので, 広域性の点でも有利である.

受信者指向の資源予約プロトコルには, RSVP がある. RSVP につては 2.4節で詳しく考察する.

2.2.2 経路制御

現在 MBone で使われているマルチキャストのための経路制御アルゴリズムは, 送信者木型のブロードキャスト (TRPB:Truncated Reverse Path Broadcast)[5] である. 名前から明らかなように, ブロードキャストに近い通信方式が使われている理由は, 単に送信者木型のマルチキャスト (RPM:Reverse Path Multicat)[5] の実装が遅れているためである. 現在では, MBone 上で TTL の制限なしに通信を行なうと, MBone の全てのネットワークにトラフィックが転送され, そのグループへの参加者がいないネットワークへ負荷をかける. ただし, これは送信者木型のマルチキャストが実装されれば解決される問題である.

一般にマルチキャストを実現するためには, 送信者からそのグループの全ての受信者へ配送木を構築する必要がある. 送信者木型のマルチキャストは, 送信者ごとに木を保持し, それをグループごとに枝刈りをすることで配送木を構築する. 送信者木型のマルチキャストを理解するために, まず, 送信者木型のブロードキャストから説明を始める.

ユニキャストにおける経路情報は, ある受信者を頂点とした最短経路木を構築する. 図 2.2左は a を頂点とした最短経路木を, 右は d を頂点とした最短経路木を示している. この受信者を逆に送信者と考えると, 送信者を頂点とした全域木が存在することになる. このように最短経路木を逆向きに使かってパケットを配送すれば, ブロードキャストが実現できる [6]. この時, ゲートウェイが保持する情報はユニキャストの経路制御表に加えて, 木として選ばれていない回線を判断する情報のみである. つまり, 送信者の数に比例する.

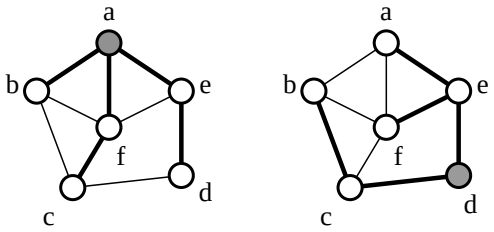

図 2.2: ユニキャストの最短経路木

この送信者木型のブロードキャストを拡張することで, 送信者木型のマルチキャストが実現できる. つまり, 送信者木において各グループ毎にメンバーのいない枝を刈り, 不必要なパケットの配送を防止する [5]. 図 2.3左は, a を頂点とした配送木をグループ A に対して, 左はグループ B に対して枝刈りを行なった例である. 送信者木型のマルチキャストでは, ゲートウェイが保持する情報が送信者アドレスの数とグループ数の積となる. 送信者木型のマルチキャストを実現したプロトコルには, DVMRP(Distance Vector Multicast Routing Protocol)[7] [1]と MOSPF(Multicast Extensions to OSPF)[8] がある.

DVMRP は, その名の通り, distance vector 型 [9] の経路制御プロトコルである. メト

[1]現在の DVMRP は文献 [7] とはかなり異なっている. 実際の仕様は DVMRP をサポートしている経路制御デーモン mrouted のソースコードを読む他ない.

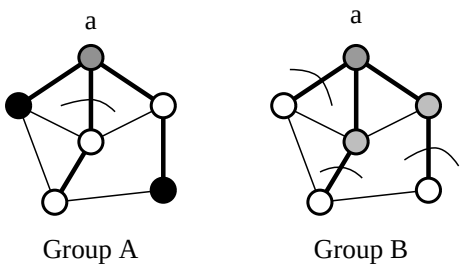

図 2.3: 配送木の枝刈り

リックにはゲートウェイの数を用い, IGMP(Internet Group Management Protocol)[10] のパケットを利用して経路制御表の交換を行なう. 一般に distance vector 型の経路制御プロトコルには, メトリックの最大値を大きくとると経路が収束するまでに時間がかかるという問題があるため, DVMRP ではメトリックの最大値を 32 と低く抑えてある. DVMRP は情報の爆発を抑えるために, 実際のパケットの転送時に送信者木の枝刈り, 継ぎ木を行なう. すなわち, 不要なマルチキャストパケットが転送されてきた場合, 「こちらにはそのグループのメンバーはいない」という否定的な情報を交換する.

MOSPF は OSPF(Open Shortest Path First)[11] の拡張として設計された経路制御プロトコルである. MOSPF は link state 型 [12] の経路制御プロトコルであるため, distance vector 型の経路制御プロトコルにみられるメトリックの制約などの問題点を解決している. OSPF のコストは, 16 ビットまで取れ, 回線速度を反映するように設定するのが一般的である. (早い回線ほどコストは小さいため, 回線速度の逆数にある値を掛けた値を用いるのが目安となっている.) 変化が生じたグループ情報は, 管理領域内のゲートウェイ全てに伝搬される. つまり, 「こちらにグループのメンバーが存在する」という肯定的な情報を交換する. 経路情報を抑えるためにパケットの転送時に木の計算を行なう.

これまで説明してきたように, 共有木型のマルチキャストにおける経路制御プロトコルでは, ゲートウェイが保持する情報が「送信者アドレス数×グループ数」と爆発し, ゲートウェイが保持できなくなる. 例えば, 現在の NSFNET のバックボーンで動作しているゲートウェイのユニキャストの経路制御表の大きさは, 約 10,000 を超えている. 経路制御表を減少させることが CIDR(Classless Inter-Domain Routing)[13] で試みられているように, 大きな経路制御表はゲートウェイに大きな負担をかけている.

仮に, 将来的にゲートウェイのメモリが増え, 約 40,000 のエントリが許されるとする. また, 現在の MBone に参加しているネットワーク数 800 であるから, 将来のネットワーク数を (非常に楽観的に見積もって)1,000 になるとする. この時, 許されるグループ数は単純計算で $40,000/1,000 = 40$ となり, 明らかに制約があることが分かる. 無論, 要求時に経路情報が生成されるため実際はこれほど単純ではない. また, (グループの肯定的な情報を必要とするプロトコルもあれば, 否定的な情報を必要とするプロトコルもあるといったように) プロトコルによって必要となる情報が違うので, プロトコル変換を行なうこと

ができないという問題がある. これは, DVMRP や MOSPF が狭い範囲にしか適応できないことを考えると, 致命的な問題である.

送信者木型のマルチキャストの問題点を解決するものに, 共有木型のマルチキャストがある. 共有木型のマルチキャストでは, グループのメンバーで 1 つの木を共有することで, ゲートウェイが保持する情報をグループ数にまで低減している. ただし, ある送信者からあるグループのメンバーへ送られるパケットは, 最短経路を通らないことが多い. 共有木型のマルチキャストを実現したプロトコルには, CBT(Core Based Tree)[14] がある.

CBT は, それぞれのグループに対し中心となるゲートウェイ (コア) をあらかじめ用意する. グループに参加するホストは, コアに向けて参加要求をユニキャストを用いて送信する. 参加要求が, コアか共有木に届くまでに参加要求を中継したゲートウェイは, そのグループの共有木の一部となる. マルチキャストパケットの送信者は, グループのメンバーである必要はない. 送信者は, グループアドレスを IP オプション部に入れ, マルチキャストパケットをコアに向けてユニキャストで送信する. そのマルチキャストパケットがコア, あるいは, 共有木の一部となっているゲートウェイに届いた場合, マルチキャストパケットの宛先であるコアのアドレスは, グループアドレスで置き換えられ, 共有木全体へ送信される. 親のゲートウェイが機能を停止した場合は, 子のゲートウェイはコアへ参加要求を出し, 再び木の一部となる.

CBT の掲げる長所を以下に示す.

- 広域性 — ゲートウェイの保持する情報がグループ数に抑えられているため, 広域で利用できる.
- 受信者による木の構築 — 木の構築が受信者によって行なわれるため, 関係のないゲートウェイは全く情報を保持しなくてよい.
- ユニキャストの経路制御プトロコルからの分離 — ユニキャストの経路制御プトロコルが算出した結果のみを利用するため, どのようなユニキャストの経路制御プトロコルが用いられていてもよい.

CBT の弱点は, (1) 最適なコアの場所を決定する適切な手段がないこと, (2) 木の (再) 構築時にループが生じる可能性があること, (3) あるグループのコアが支障をきたした場合代替経路があるにも関わらずそのグループへの通信が全く行なえなくなることである. よって, これらの問題を解決しなければならない.

2.3 マルチキャストにおけるネットワーク層の設計目標

ネットワーク層では, パケット転送, 経路制御, および, 資源予約を行なう. 本稿では, パケット転送のためのプロトコルとして, 資源予約機能を有するものを想定し議論を行なう. パケット転送のためのプロトコル自体は特に議論の対象とはしない. 2.4節で述べるように, 我々の提案する共有木に基づいた経路制御と資源予約は密接な関係がある.

2.2節で考察したように, 広大なインターネットにおけるマルチキャストの経路制御プロトコルは以下の特徴を有さなければならない.

- 広域性 — 適応範囲が広大になっても機能すること
- 安定性 — 障害時に復旧するための機能があること

前述のように広域性は, 我々の枠組を貫くテーマである. また, パケットが安定して配送されることは, 上位層への基礎となるものである.

遅延のばらつきの範囲を小さく抑えるための資源予約プロトコルは, 以下の要求を満たさなければならない.

- 広域性 — 適応範囲が広大になっても機能すること
- 応答性 — 資源予約にかかる時間をできるだけ小さくすること

一般的に応答性を高めることは, 広域性を高めることにつながる.

2.4　共有木に基づくマルチキャスト経路制御と資源予約

この節では, マルチキャストにおいて経路制御と資源予約の両方の機能を提供する制御プロトコル SnowCrystal の特徴を述べる. そして, 2.5節で経路制御プロトコルとしての側面, 2.6節で資源予約プロトコルとしての側面を解説する.

SnowCrystal は, マルチキャストにおいて経路制御と資源予約を行なうネットワーク層の制御プロトコルである. SnowCrystal では, マルチキャストの共有木がコアを中心に四方に伸びていくように構築される. このさまが雪の結晶の成長に似ているため, SnowCrystal という名前を付けた. 経路制御プロトコルとしてみると, SnowCrystal は共有木型の経路制御プロトコルであり, 資源予約プロトコルとしてみると, 受信者指向の資源予約プロトコルである. 資源予約は共有木に対して行なわれる.

以下に SnowCrystal の特徴を示す.

- SnowCrystal は制御メッセージは**経路制御メッセージ**, **経路破棄メッセージ**, **資源予約メッセージ**, **資源維持メッセージ**, および, **資源解約メッセージ**からなる. 資源予約メッセージおよび資源解約メッセージ以外は, ローカルのネットワーク上で, マルチキャスト機能を持った全ホスト／ゲートウェイが参加しているグループ G_{sc}に対し送られる. (G_{sc}宛のパケットはホストおよびゲートウェイは, 状態に関わらず必ず受け取る. 別の言い方をすれば, 全てのホスト／ゲートウェイは, 常にグループ G_{sc} に参加している.) 資源維持メッセージは経路制御メッセージに同期している. 資源予約メッセージと資源解約メッセージは, 特定のゲートウェイ宛に送信される.
- 受信者と同様, 送信者もグループのメンバーとなる. 送信者は, グループのメンバーであるため同時に受信者でもあるが, 本稿では単に送信者と呼ぶ.

- 第一送信者がコアとなり, グループアドレスを経路制御メッセージより発生し, プレゼンテーションを開始する. プレゼンテーションの終了時には, コアは経路破棄メッセージを送出する.

- 受信者, および, 第二以降の送信者がグループへ参加する際は, 資源予約を行ないながらコアに向かって共有木を構築する.

- SnowCrystal におけるマルチキャストは, 送信木に対するブロードキャストとして実現される. パケットの宛先はグループアドレスである. 送信者といえども, 発せられたパケットを受け取る.

- 資源予約のパラメータを決定できるのは, コアのみである. 資源予約メッセージは経路決定メッセージに同期して, 定期的に更新されるので, 経路変更時の共有木の変化に追従できる.

ここで, 本章で使用する言葉を以下のように定義する.

経路 コアへの最短の道筋.

経路制御 コアを中心とした最短経路木を構築すること. 経路の決定は, 拡張の施されたユニキャストの経路制御アルゴリズムを使用する.

経路破棄 コアが最短経路木を明示的に破棄すること.

共有木の構築 グループの全メンバーで共有する共有木を構築すること. 共有木の構築は, (コアへの) 経路に従って行なわれ, 同時に資源が予約される.

資源予約 資源予約の視点から見ると, 共有木の構築は資源を予約することである.

資源解約 獲得している資源を解放すること.

経路制御表 コアへの経路と共有木の情報を保持する表. ホストおよびゲートウェイが持つ.

経路情報 あるクラス G の経路を決定するために, 経路制御メッセージを用いて交換される情報. 経路の決定のための最低限の情報として, メトリックが必要である.

親ゲートウェイ あるグループ G に対し, 経路が向いている (next hop となっている) ゲートウェイ.

子ゲートウェイ あるゲートウェイにおいて, そのゲートウェイに経路を向けているゲートウェイ.

SnowCrystal における経路制御と資源予約には密接な関係があり, 両者を区別するのは非常に困難である. しかしながら, SnowCrystal の経路制御機能は 2.5節で, 資源予約機能は 2.6節で個別に説明することを試みる. 経路制御の議論は, 経路の決定, および, 共有木の構築について行ない, 資源予約を議論する時は, 共有木の構築を資源予約の過程とみなす.

2.5　マルチキャスト経路制御プロトコルとしての SnowCrystal

この節では, SnowCrystal の経路制御に関する部分のみについて解説を行なう. 説明を簡略化するため, まず, ゲートウェイと point to point ネットワークのみからなるグラフ上で SnowCrystal の挙動を示す. その後で, マルチアクセスネットワーク, および, ホストが存在するネットワークでの考察を行なう.

ゲートウェイ (ホスト) が保持する経路制御表は, グループアドレス, 親ゲートウェイ, 枝情報からなる. 枝情報とは, インターフェイスが共有木の枝であるかどうかを示す情報である. 本稿では, 枝として選ばれている場合は○, そうでなければ×という表記を用いる. 表 2.1は, グループ G に対し, 親ゲートウェイが N で, インターフェイス IF_1が共有木の枝となり, インターフェイス IF_nが共有木の枝ではない状態の経路制御表の例である.

表 2.1: マルチキャストゲートウェイの経路制御表

グループ	親	IF_1	$\cdots$	IF_n
G	N	○		×

2.5.1　経路情報の発生 — プレゼンテーションの開始

プレゼンテーションの開始は, 第一送信者によって行なわれる. 第一送信者はコアとなり, 割り当てられたグループアドレス G に対する経路情報を, 経路制御メッセージにより発生する. この経路制御メッセージの配送は, 拡張の施されたユニキャスト経路制御技術を用いる. 以下の説明では, 経路制御アルゴリズムとして, distance vector アルゴリズムを用いるが, path vector[15] や link state アルゴリズムでもここでの議論は同様に当てはまる. グループアドレス G を経路制御メッセージを用いて, ローカルネットワーク上で G_{sc}宛でアナウンスする. この経路情報を受け取ったゲートウェイは, 他のネットワークへ転送を行なうため, 経路情報が全域に伝播する. この時コアを中心とした最短経路木ができる.

point to point のネットワークの例として, 図 2.4左のネットワークを用いる. a から q のアルファベットのついた丸がゲートウェイであり, 1 から 25 の数字がついた実線がリ

ンクである. point to point ネットワークでは, 親ゲートウェイ, リンク, および, インターフェイスが一対一に対応しているので, しばらくの間厳密な区別を行なわない. (しかしながら, マルチアクセスネットワークを考えた場合, 実質的に重要であるのはインターフェイスである.)

ここで, j がグループ G の第一送信者となる場合を考える. G を経路制御メッセージを用いてアナウンスし, j を中心とした最短経路木が構築された状態が図 2.4右である. 最短経路木に関係のないリンクは点線で示した. この時の c における経路制御表を表 2.2に示す. 親ゲートウェイが e で, インターフェイス 2, 3, 5, 7, 8 は共有木の枝として選ばれていない.

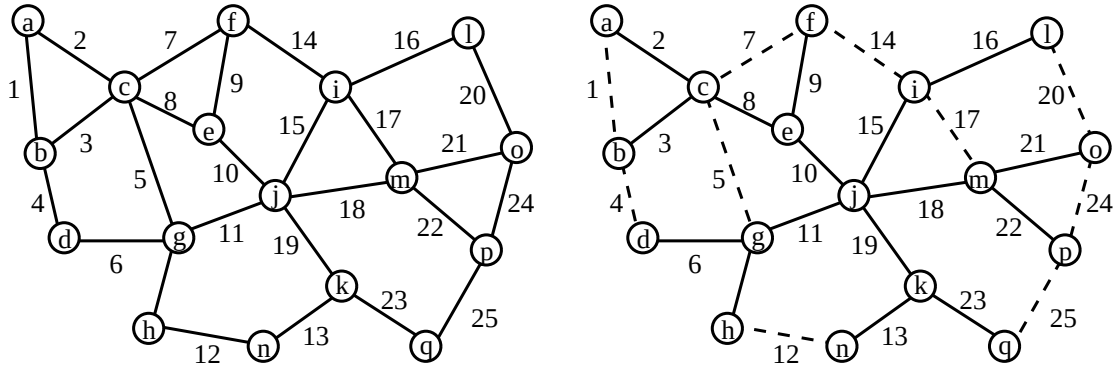

図 2.4: 最短経路木の形成

表 2.2: マルチキャスト経路制御表

グループ	親	2	3	5	7	8
G	e	×	×	×	×	×

共有木の枝として選ばれていないインターフェイスから, マルチキャストのパケットが到着した場合は, 誤って配送されたパケットとして単に捨てられる. このように, ゲートウェイの初期は, マルチキャストパケットを受け取らない状態にある. (ただし, G_{sc}へは常に参加しており, G_{sc}宛のパケットは常に受け取る.)

2.5.2 グループへの参加

グループ G に関する経路制御メッセージを受け取っている間は, グループ G に対するコアが存在することが保証されている. これは, ゲートウェイ上のプロセスが, そのグループの存在を知ることができることを意味している. ここで, ゲートウェイ上のプロセスが, グループへ参加する場合を考える.

グループへの参加は, 資源予約メッセージを用いて行なわれる. 資源予約メッセージは, グループ G に対する親ゲートウェイへ送られる. 資源予約メッセージを受け取ったゲートウェイが, まだ共有木の一部でないときは, さらに, 親ゲートウェイへ転送される. このように, 資源予約メッセージは, 共有木かコアに到着するまで連鎖的に転送され続ける.

資源予約メッセージが出力されるインターフェイスは, 出力時に共有木として選ばれる. 本稿の表記法でいえば, 経路制御表のエントリが×の場合, ○に置き換えられる. 資源予約メッセージが入力されたインターフェイスは, それがグループ G に対する親ゲートウェイへのインターフェイスでなければ, 共有木として選択される. 親ゲートウェイへのインターフェイスと同じであれば, (ping pong) ループが発生していると考えられるので, 資源予約メッセージは無視される.

例えば, 図 2.4において, b がグループ G に参加する場合を考えてみる. b は, インターフェイス 3 を共有木の枝として選択し, 資源予約メッセージを c へ送る. c は 3 を共有木の枝として選び, まだ親ゲートウェイへのインターフェイスである 8 が共有木として選ばれていないので, 8 を共有木に選択し, 資源予約メッセージを e へ転送する. 同様に, e は 8,10 を共有木として選択し, j へ転送する. j が 10 を共有木として選択したときに, b と j 間の共有木の構築が完成する (図 2.5左). (コアにおいては, 親ゲートウェイはコア自身となる.)

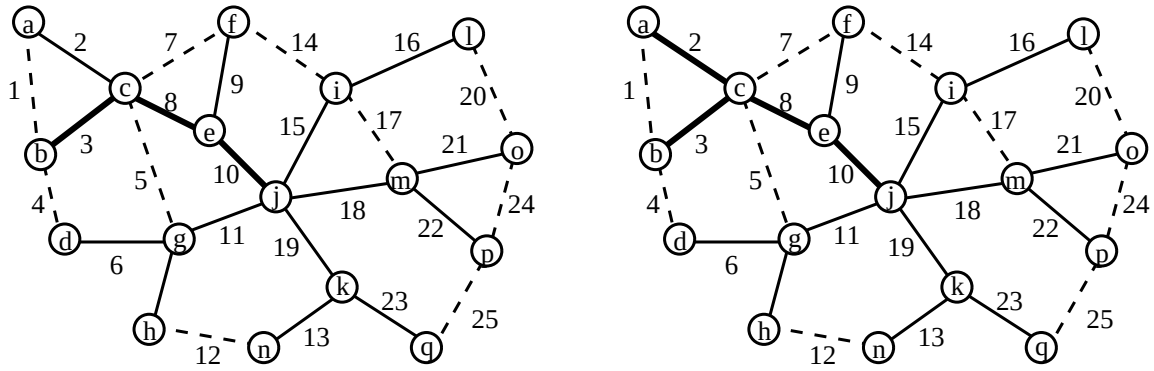

図 2.5: 共有木の構築

b から j までの共有木が構築された際の c における経路制御表を表 2.3に示す. ここで, a がさらにグループ G に参加した場合を考えてみる. a はインターフェイス 2 を共有木の枝として選び, c へ資源予約メッセージを送る. 資源予約メッセージを受け取った c は, インターフェイス 2 を枝として選ぶが, すでに親ゲートウェイへのインターフェイス 8 は, 枝として選ばれており, 資源予約メッセージをさらに転送することは行なわない. この時の c における経路制御表を表 2.4に示す.

表 2.3: マルチキャスト経路制御表

グループ	親	2	3	5	7	8
G	e	×	○	×	×	○

表 2.4: マルチキャスト経路制御表

グループ	親	2	3	5	7	8
G	e	○	○	×	×	○

2.5.3 トラフィックの配送

送信者といえどもグループのメンバーに参加するため, グループへのトラフィックは共有木の中から発生する. 送信者はグループ G へパケットを送信する場合, 宛先に G を指定する. 送信者は, 共有木の枝として選ばれている全てのインターフェイスに対してパケットを送出する. ゲートウェイはパケットを受け取らなかったインターフェイスのうち, 共有木の枝として選ばれているインターフェイスに転送を行なう. 送信木の枝として選ばれていない, インターフェイスから G 宛のパケットを受け取った場合は, 何らかの異常が発生しているのでそのパケットを破棄する.

例えば, 図 2.5右において, 第一送信者であるコアがトラフィックを発生したとしよう. コアは共有木として選ばれているインターフェイス 10 に, パケットを出力する. これを受け取った e は, 入力インターフェイス 10 以外で枝として選ばれているリンク 8 へパケットを転送する. 同様に c はインターフェイス 2, 3 へパケットを送り, 受信者 a, b がマルチキャストのトラフィックを受信できる.

2.5.4 経路の変更と共有木の維持

共有木の維持のため, 資源維持メッセージが経路情報メッセージへの応答として送出される. このときは, 宛先として G_{sc}を用いる. つまり, グループ G に対する親ゲートウェイから, 経路制御メッセージが到達した時に, G に対する資源維持メッセージを G_{sc}宛に返す. 複数の資源維持メッセージを異なったインターフェイスから受け取る, 共有木の分岐点に位置するゲートウェイにおいても同様である. つまり, 経路制御メッセージの応答として, 複数の資源維持メッセージを異なったインターフェイスから, 異なったタイミングで受け取ったとしても, 親ゲートウェイへの資源維持メッセージは, 親ゲートウェイからの経路制御メッセージに同期して返される. 一定時間資源維持メッセージが到着しない場合, そのインターフェイスは共有木の枝から外される.

リンクやゲートウェイの障害により, 共有木として選ばれた枝に関係する経路が変更されると, ゲートウェイに対し次の変化が生じる可能性がある.

- 親ゲートウェイが変わる.
- 今まで受け取っていた資源維持メッセージが到着しなくなる.

これまで述べてきたように, 親からの定期的な経路制御メッセージに対して, 子は資源維持メッセージを G_{sc}宛に返す. 経路制御メッセージは親から子への keep alive, 資源維持メッセージは子から親への keep alive となっているため, 上記の経路の変化に追従できる.

例として, 図 2.5右において, リンク 8 に障害が発生した場合を考える. リンクの障害がハードウェアで検出できれば, e は直ちに 8 を共有木の枝から外し, (e がグループに参加していないなら)j に資源維持メッセージを送ることを止める. リンクの障害が検出できない場合, e はインターフェイス 8 から資源維持メッセージを受け取らなくなるので, 一定時間待った後, リンク 8 を共有木の枝から外し, j への定期的な資源維持メッセージの送出を止める. c ではリンク 8 から経路制御メッセージを受け取らなくなるため, 親ゲートウェイが例えば g に変更される. そこで, c は 8 のインターフェイスを枝から外し, g の経路制御メッセージに対し, 資源維持メッセージで応答する. よって, 最終的に共有木は, 図 2.6のように変更される.

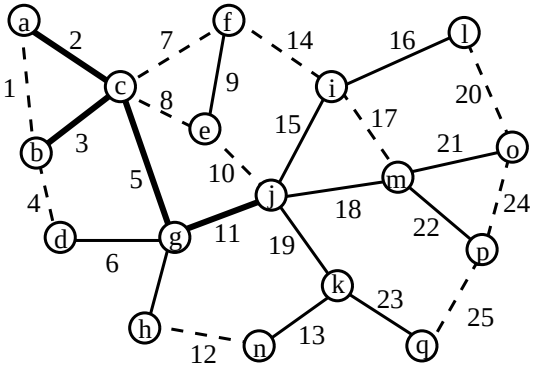

図 2.6: 経路の変更

このように, 経路が変更されたゲートウェイでは, 古い親ゲートウェイへの枝を外し, 新しい親ゲートウェイへのインターフェイスを枝として選ぶ. しかしながら, 他の枝についてはなにも行なわないことに注意せよ. 例えば, 上記の例で c は 2, 3 を枝から外したりはしない. これは, マルチキャストパケットが重複しようとも, できるだけメンバーにそのパケットを送ることに重点を置いているからである.

例えば, 図 2.7左において, a がコアで, d, e がグループに参加しており, 1, 2, 4, および, 5 が共有木として選ばれているとする. ここで, リンク 1 に障害が起こり, 経路が図 2.7右のように変化したとする. この図では, d の親ゲートウェイが e に変化しているが, b のまま変わらない場合も起こりうる. b が親ゲートウェイをを a から c へ切り替えた際に, 4 を共有木の枝から外すと, 4 へはトラフィックが流れない. d の親が b のままであった場合は, d はこのトラフィックを落としてしまうという不都合が生じる. また, 経路変更時に

元の親ゲートウェイへ経路解約メッセージを送ることは保証できない. 例えば, b はリンク 1 に障害が起きたのだから, a へ経路解約メッセージを送ることはできない. しかしながら, d からの資源維持メッセージは到着しなくなるので, 時間が経てば b は 4 を枝から外してもよいことが分かる. よって, 経路変更時は, 子ゲートウェイに関する状態を保持する方がよい.

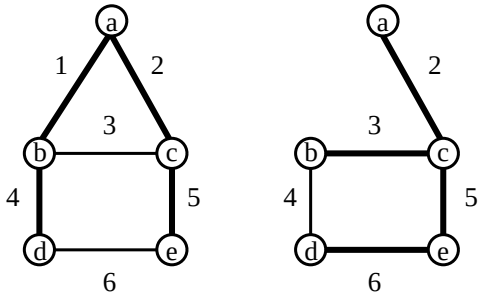

図 2.7: 経路の変更

2.5.5 グループからの脱退

グループから脱退する受信者, および, 送信者は, 経路制御メッセージに対する定期的な応答, つまり, 資源維持メッセージの送信を停止し, 資源解約メッセージを親ゲートウェイに送る. この時, 同時に共有木の枝として選ばれていたインターフェイスを共有木の枝から外す.

グループ G に対する資源解約メッセージを受け取ったゲートウェイは, 入力インターフェイスに対しグループ G の経路制御メッセージを, G_{sc}宛で送出する. ここで, 資源予約メッセージが戻ってきた場合はなにも行なわない. (これは, マルチアクセスネットワーク上に他のメンバーが存在する時に起こる. 詳しくは 2.5.8節を参照.) 資源維持メッセージが戻ってこない場合は, そのインターフェイスを枝から外す. また, 他に資源維持メッセージを受け取っているインターフェイスがなく, そのゲートウェイ上のプロセスもグループに参加していない場合は, そのゲートウェイの親ゲートウェイに資源解約メッセージを転送する. 他に資源予約メッセージを受け取っているインターフェイスがある場合, または, ゲートウェイ上のプロセスがグループに参加している場合は, 引続き親ゲートウェイの経路情報に, 資源維持メッセージで応答する.

図 2.5右において, b が脱退する場合を考える. まず, b はインターフェイス 3 を共有木の枝から外し, c へ資源解約メッセージを送る. この時点で b はマルチキャストパケットを受け取らなくなる. c はインターフェイス 3 へ経路制御メッセージを送り, 資源維持メッセージによる応答がないのを確認後, インターフェイスを枝から外す. c はインターフェイス 2 から a の資源維持メッセージを定期的に受け取っているので, ここで b の脱退による連鎖反応は終る.

2.5.6　経路情報の破棄 — プレゼンテーションの終了

コアがグループから脱退した時に, プレゼンテーションが終了する. コアがグループから脱退する時は, 経路制御メッセージのアナウンスを停止し, 経路破棄メッセージを送信し, すべてのメンバーにプレゼンテーションの終了を告げる. 経路制御メッセージのアナウンスの停止で, 経路の破棄を代用せず, 経路破棄メッセージを明示的にアナウンスするのは, 障害の発生のために, 経路制御メッセージが到達しない場合と区別するためである. グループ G の経路破棄メッセージを受け取ったゲートウェイで, まだプロセスがグループ G へ参加している場合は, プロセスへシグナルを送りプレゼンテーションの終了を告げる.

2.5.7　コアの障害

コアの障害時には, 他のホスト／ゲートウェイがコアにとって代わる必要がある. 現時点では, 確定的な方法を決定していないが, ここではコアの障害時の復旧に関するアイディアを示す.

コアが障害を起こした時に, コアにとって代わって経路情報を発生するのは別の送信者であるとよい. それは, なるべく送信者から受信者へ最短経路を確立する方がよいからである. コアの他に送信者がいない場合は, コアが障害を起こした時点でプレゼンテーションは消滅すると考えられる. よって, 他に送信者が存在する場合, どの送信者がコアにとって代わるかを考えればよい.

コアの代替となる送信者はなるべくコアに近いものがよい. なぜなら, コアの近くから経路情報が発生するため, 経路の変更が少なくてすむ可能性があるからである. よって, SnowCrystal でコアに最も近い送信者 (代替送信者) を決定しておけばよい. 資源予約の過程で, 代替送信者を決定し, 2.6.6節で説明する MTU(Maximum Transmission Unit) の場合のように, コアへ伝えることは容易である. コアは, その送信者へ指示を出し代替送信者へなることを依頼する. 代替送信者は, コアを見張りコアが障害を起こした時は経路情報を発生することで, 障害を復旧することができる. また, コア自体が復旧した場合, 代替送信者は経路情報のアナウンスを停止すればよい.

2.5.8　マルチアクセスネットワークに関する考察

これまで, point to point ネットワークのみの場合について考察してきた. 現実のインターネットでは, イーサーネットのようなマルチアクセスネットワークが存在するため, point to point ネットワークでは生じなかった問題が発生する.

図 2.8の単純なネットワークを例にとって説明する. グループ G のコアが a で, 経路情報が同じメトリックで, b, c へ届いていた場合, d の親ゲートウェイが b, e の親ゲートウェイが c となる可能性がある. d が b へ, e が c へ資源予約メッセージを出した場合, a からの G のトラフィックを b, c の両方が N へ転送し無駄が生じる. また, c からの G のトラフィックを b が受け取とり, a へ転送するため, ループが生じる (逆も真).

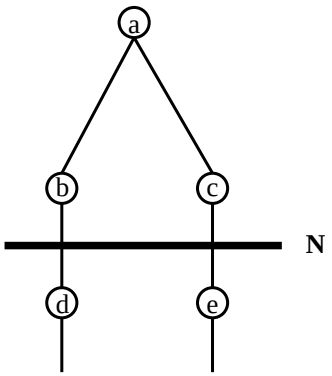

図 2.8: マルチアクセスネットワーク

このような問題を解決するためには, 経路情報の交換時に親ゲートウェイの決定を行なう必要がある. つまり, メトリックが同じ場合には, (例えば IP アドレスなどの) ゲートウェイ ID といった別の尺度を用い, 親ゲートウェイを決定すればよい. 親ゲートウェイにならなかったゲートウェイは, 経路情報のアナウンスを抑制する.

あるゲートウェイがグループへ参加する時には, 伝播を素早く行なうために必ず資源予約メッセージを送出しなければならない. コアあるいは共有木までへの伝播は連鎖的に起こる. しかしながら, マルチアクセスネットワークにおいて, 定期的な経路制御メッセージへは, 1 つの子ゲートウェイのみが資源予約メッセージで応答すればよい. 後で述べるが, 資源予約メッセージには, コアまで最小の MTU を決定するために, そのゲートウェイまでの最小の MTU が含まれている. そこで, 定期的な資源予約メッセージへの応答タイマは, MTU で重み付けがなされる. もし, 自分が保持している MTU より大きな MTU が先に答えられた場合は, そのゲートウェイは余分な資源予約メッセージを送り, MTU の値を正しく親ゲートウェイに告げる必要がある.

2.5.9 ホストに関する考察

これまでホストの存在を仮定していなかったが, ここでは実際のインターネットのようにホストが存在する場合について考察する. 基本的には, ホストは 1 つのインターフェイスを持ったゲートウェイと考えてよい.

例えば, 図 2.9において, ホスト b が第一送信者, つまり, コアになる場合, グループ G に対する経路制御メッセージを G_{sc}宛でネットワーク N2 へ送出する. ゲートウェイ a, c はこの経路情報にコストを加えて, それぞれ N1, N3 へ中継する. ホスト d が G へ参加するした場合, 必ず資源予約メッセージを c へ送出しなければならない. 定常時の c からの経路制御メッセージにも, ホスト d は応答をする. また, 例えば, ホスト d とゲートウェイ e が G へ参加しており, 定常状態ある場合を考えてみる. この時, c からの経路制御メッセージへは, MTU で重みの付けられたのタイマを用いて, MTU の小さな方が応答を行な

う. ホストの MTU は, 自分が属しているネットワークの MTU とすればよい.

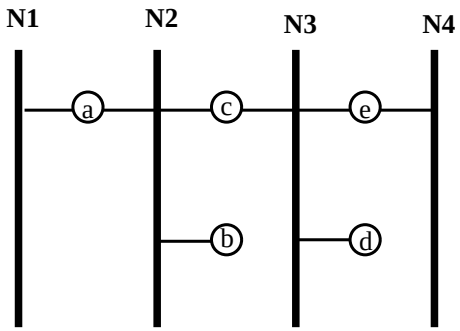

図 2.9: ホスト

2.5.10　CBT との比較と評価

この節では, SnowCrystal と CBT の比較を行なう. まず, CBT に対して, SnowCrystal が劣っている点を挙げ, 次に優れている点を説明する.

SnowCrystal が CBT に比べて劣っている点は広域性である. CBT は, ユニキャストの経路制御表を利用するため, コアに対する経路情報を保持する必要がない. また, あるグループに関係のない (共有木を形成しない) ゲートウェイでは, 全くそのグループに関する情報を保持しなくてよい. これに対し, SnowCrystal では, 存在する全てのグループのコアに対する経路の情報を持つ必要ある. また, あるグループに関係のないゲートウェイも, 存在する全てのグループに対する情報を持つことになる. この点から, SnowCrystal は CBT に対して広域性が劣っているといえる. (むろん, 送信者木のプロトコルと, 共有木型の SnowCrystal とを比較した場合は, SnowCrystal の方が格段に優れている.)

SnowCrystal が CBT に比べて優れている点は, (1) コアの自由度, (2) グループの存在の保証, および, (3) 経路変化への追従である. まず, (1) のコアの自由度であるが, CBT ではあらかじめコアを静的に決定しておく必要があるし, コアの位置を決定する適切な方法がない. これに対して, SnowCrystal では, 第一送信者がコアとなることができる. また, 送信者が 1 つの場合は受信者への最短経路が保証されており, コアを決定する適切な方法を提供しているといえる.

次に, (2) のグループの存在の保証であるが, CBT ではある時点でグループが存在するかどうかを知る手立てがない. よって, 送信者がいないにも関わらず受信者が参加を行なうといったことが起こりうる. これに対し, SnowCrystal では, 全てのホスト (ゲートウェイ) 上でグループに対する経路情報の有無により, グループの存在を知ることができる. また, 経路情報の到着は少なくとも 1 つ送信者が存在することを保証している.

経路の決定をユニキャストの経路制御表に頼っているため, CBT では (3) の経路変化への追従は困難である. これは, コアへの経路の変化を簡単には検知することができないた

めである. つまり, コアへの経路を意味するエントリは無数に発生しうる (例えば, IP ではコアへの経路であるアドレスとマスクの組は複数存在する) ので, 経路を検知するためには, まず, 確答するエントリを検索しなければならない. 次に, このエントリが以前の状態から変化しているかを調べる必要がある. これは複雑な機構であるので, CBT では経路の変化を検知しない.

その代わり, CBT では親ゲートウェイへ参加要求を出し, 親が認めれば枝を構成する. これは, 経路情報を瞬時的にしか検索しないことを意味しており, 経路が変化する過渡期に参加が起こった場合, 共有木にループが発生する可能性がある. また, 親ゲートウェイに障害が生じた場合, 子を従えたまま再参加を行なう. このとき, あるゲートウェイで親ゲートウェイとコアへの経路上にある次のゲートウェイが一致しなくなる場合がある. 結果として, 共有木のループが生じる. そこで, CBT ではループ検出を行なう必要がある. CBT では, ループ検出の機構を提供しているけれども, それは CBT の動作を複雑にしている.

これに対し, SnowCrystal では, 単純にグループアドレスの経路情報を見張ることで, 経路の変化を検知できる. また, 経路制御メッセージは親から子への, 資源予約メッセージは子から親への keep alive となっており, 経路の変化に追従し, 共有木を維持することができる.

CBT と SnowCrystal の長所短所は, 結局トレードオフの関係にあるといえる. つまり, SnowCrystal では厳密な経路情報を発生するので, 細かい動作が簡単に行なえるが, 保持すべき情報は多い. CBT では抽象的な経路情報に頼るため, 保持すべき情報は減るが, 細かい動作が行なえないわけである.

2.6 マルチキャスト資源予約プロトコルとしての SnowCrystal

この節では, SnowCrystal を資源予約プロトコルの面から説明する. 2.2章で示したように, メンバー管理を厳密に行なわないでよい場合は, 資源予約プロトコルは受信者指向の方が優れている. そこでまず, 受信者指向の資源予約プロトコル RSVP の機能を調べることから始め, RSVP において現実的に必要な機能と不必要な機能を分ける. これを踏まえて, SnowCrystal の設計を行なう.

2.6.1 RSVP のフィルタ

RSVP は, マルチキャスト経路制御プロトコルとの独立を唱っているものの, 送信者木型の経路制御プロトコルを仮定している. 送信者からは定期的に path メッセージが届いているので, ゲートウェイは, どのインターフェイス側にどの送信者がいるかを知ることができる. 資源予約の機能として, RSVP は以下のフィルタを提案している.

- Wild card フィルタ

- 固定フィルタ
- 動的フィルタ

以下の節で各フィルタの説明を行なう.

2.6.2 Wild card フィルタ

Wild card フィルタで予約された資源は, あるグループに関係する全てのトラッフィクを通過させる. 受信者はグループ G の送信者が存在する方向のインターフェイスに, 予約メッセージを送出する. グループ G に対して予約された回線上を通過するパケットは, 送信者や受信者に依存しないため, グループに対する予約は可能な限りマージされる. マージされた時に予約される回線容量は, 個々の予約のうちの最大値である.

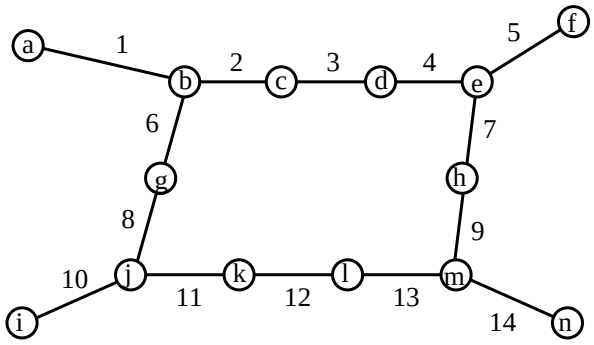

図 2.10: Wild card フィルタによる資源予約

例えば, 図 2.10において, a から f への経路が 1-2-3-4-5, a から n が 1-2-3-4-7-9-14, i から f が 10-11-12-13-9-7-5, i から n が 10-11-12-13-14 であるとする. この時 f が wild card フィルタにより, 回線容量 B を予約すると, (1 2 3 4 5 7 9 10 11 12 13 14) が B ずつ予約される. 次に n が 3B の予約を要求すると, B よりも 3B の方が大きいため, (1 2 3 4 7 9 10 11 12 13) の予約は B から 3B となり, また, 新たに (14) が 3B 予約される.

Wild card フィルタにおいて, 受信者が同じ質の資源を予約する時は, 経路が送信者木である場合よりも, 共有木である方が全体で予約する資源を節約できる. 単純なトポロジーではあまり差がでないが, 代替経路が多いほど送信者木では予約される回線が増える. これに対し, 共有木では回線を共有するので, 代替経路の数にかかわらず一定の資源が予約される.

受信者が異なった質で資源を予約する時は, 最大の資源を予約する受信者と送信者のネットワーク的な距離が, 全体で予約する資源の量に大きな影響を及ぼす. しかしながら, 資源の質の変わり目となるゲートウェイは, RTP(Real Time Protocol)[16] などの符号変換機能が必要となる. このため, 同じグループで極めて違う質の資源を予約する需要があるなら, グループを分けて通信を行なった方が, 符号変換の問題がないためよいであろう.

2.6.3 固定フィルタ

固定フィルタは, あるグループにおける送信者を選択して資源予約を行なう機能である. 指定した送信者が等しければ, 複数の受信者からの要求はまとめられる. 固定フィルタは, 送信者木型の経路制御を行なっていないと使用できない. なぜなら, 共有木型の経路制御では, コア以外の送信者への最短経路は存在しないからである.

固定フィルタは, グループのある送信者から各受信者への配送木において, 必要な部分だけを資源予約する機能であるといえる. 送信者の数が少なければ, 送信者を選択することは難しくないが, 多くなると困難になると考えられる. また, 同じグループの送信者を選択するという需要があるかは疑問である. そういった需要がある場合は, グループを分ける方がよいのではないだろうか.

2.6.4 動的フィルタ

動的フィルタは, ある受信者が単数あるいは複数の送信者に対して, 資源を予約する. 送信者への経路が重なっている場合は, マージして予約を行なう. 別々の受信者の予約が, マージされることはない. 動的フィルタは, 受信者 1 人に対して複数の送信者がいる場合のスイッチング機能であるといえる.

例えば, 図 2.10おいて, 受信者 n が送信者 a, i へ回線容量 B を予約した場合を考えてみる. この場合, 例えば, (1 6 8 10 11 12 13 14) の回線が予約され, n の指定で, 送信者 a と i のどちらのからのパケットを通過されるかを動的に指定できる. 受信者 f が, 送信者 a, i へ回線容量 B を予約した場合, 回線容量は別に確保される.

送信者が増えれば存在する方向はばらつき, 必ずしも予約する回線を集約できないため, 高速なスイッチングのみが主な利点である. また, ある時点で受信されていない送信者への回線も確保しておかねばならず, その回線上にはパケットがながれないことを考えると予約に無駄が多いといえる. しかも, ユニキャストで簡単に模倣できることを考えると, 必ずしも必要ないことが分かる.

動的フィルタの高速スイッチング機能を利用する例として, 文献 [17] ではテレビ局のチャンネルを挙げている. 複数のテレビ局がチャンネルを提供し, 受信者はチャンネルを高速に切り替えるのである. しかし, テレビ局は点在すると考えるのが自然であるし, 違うテレビ局同士が同じグループに属すとは考えにくい. テレビ局ごとに 1 つのグループを形成するであろう. よって, 動的フィルタが必ず必要であるという端的な例を挙げることができない.

2.6.5 SnowCrystal の資源予約機能

これまで考察してきたように, 共有木型の経路制御を行なった場合, 実際に用途があるのは wild card フィルタであろう. また, wild card フィルタは, 受信者が同じ質の資源を予約する場合, 共有木型のマルチキャスト経路制御プロトコルに非常にうまく適合することは述べた通りである.

よって, 共有木型の経路制御を行なう SnowCrystal では, wild card フィルタの機能を提供するように設計されている. SnowCrystal の資源予約機能の特徴は以下の通りである.

- コアがプレゼンテーションの開始時に資源予約のパラメータを決める. つまり, コアがプレゼンテーションの質を決定する.
- Wild card フィルタのみを提供する.
- 共有木における最小の MTU を求める.

今回の設計では, 単純かつ確実に機能することに重点をおいているため, 受信者が個々にパラメータを決定することはできない. むろん, 全体で予約できる最大値がコアによって決定された場合, それぞれの送信者が最大値よりも小さい値で, 資源を予約することは可能である. しかしながら, 例えば 2B から B へ回線容量が変化する場所では, RTP のような符号変換機能が必要である. 今回は, ゲートウェイが符号変換機能を持つことを仮定していないので, 全ての受信者はコアの決定した値で資源予約を行なう.

2.6.6　資源予約と解約の動作

ここで資源予約メッセージ, 資源維持メッセージ, および, 脱退メッセージに関係する動作をまとめる. あるホスト／ゲートウェイ上で, プロセスがグループ G へ参加する場合, 親ゲートウェイに対して資源予約メッセージを送る. 資源予約メッセージには, グループ G と MTU が含まれている.

資源維持メッセージは, 定期的な経路制御メッセージへの応答への応答として, グループ G_{sc}宛に送る. 資源維持メッセージを受け取った, ホスト／ゲートウェイは MTU で重み付けされた (短い) タイマを開始させ, タイマが切れた時点で資源維持メッセージを送出する. MTU は, ホストであれば自分が属すネットワークの MTU, ゲートウェイであれば自分以下の部分木における最小の MTU を用いる. 資源維持メッセージには, グループと MTU が含まれている. 一番小さい MTU を持つホスト／ゲートウェイが, 資源維持メッセージを送出すると期待できる. この資源維持メッセージは他のメンバーも聞いており, 自分よりも小さい MTU が指定されていれば, タイマを破棄し, 資源維持メッセージによる応答は行なわない. もし, 自分より大きな MTU が指定されていれば, タイマが切れた時点で資源維持メッセージを送る. こうすることで, 親ゲートウェイは, 部分木における最小の MTU を知ることができる.

グループからの脱退時には, 資源解約メッセージを親ゲートウェイに対して送る. 資源解約メッセージを受け取ったゲートウェイは, いったん経路情報メッセージを入力インターフェイスから送出する. このメッセージに応答がない場合は, 入力インターフェイスを共有木の枝から外す.

2.7 考察

今回の設計では, distance vector アルゴリズムの経路制御プロトコルを基に, SnowCrystal を設計した. 経路制御に最低限必要な情報は, コアへの経路と親ゲートウェイだけであるので, path vector や link state アルゴリズムとのプロトコル変換は比較的容易に行なえる. 事実, distance vector, path vector, および, link state アルゴリズム間での経路情報交換は, ユニキャストの経路制御において実現されている. また, link state の代表的なプロトコル SPF(Shortest Path First) と path vector の BGP(Border Gateway Protocol) は親決めを行なう機能がある. よって, 共有木型の経路制御をこれらのプロトコルに適応するのは容易である.

また逆に, 資源予約に強い依存関係を持たせず, 各種の経路制御プロトコルがコアへの経路を算出した結果のみを, CBT のように利用する方がよいかもしれない. つまり, 経路制御と資源予約に独立性を持たせるのである. この時, 資源予約プロトコルは次のような性質を満たさなければならないと考えている.

- 資源予約／維持の間隔は, 経路の変化に追従できるよう十分に短くなければならない.
- 資源予約／維持の間隔は, ネットワークに負荷をかけないように十分長くなければならない.

上記の項目を考慮した上で, もう一度設計を議論し直し, 実際に実装を行なおうと計画している.

2.8 おわりに

本章では, 現在のネットワーク層の問題点を指摘し解決策を示した. ネットワーク層では, 広域性のある経路制御プロトコルを用いる必要があり, また, 遅延時間を一定にするための資源予約が必須である. 我々は, これらを同時に解決する制御プロトコル SnowCrystal の設計を行なった. SnowCrystal を経路制御プロトコルの側面から眺めれば, 細かい操作を行なえるように CBT を拡張したものであるといえる. また, 資源予約プロトコルとして見ると, RSVP の共有木への現実的な適応となっている. 今後は, 経路制御と資源予約に独立性が必要であるかを再検討した上で, 実装を行なっていく予定である.

第 3 章

NACK を用いた信頼性のあるマルチキャスト

3.1 信頼性の要求

近年，マルチキャストが広域ネットワークでの音声や画像を使った会議システムのアプリケーションに広く利用されるようになった．マルチキャストはグループ通信の枠組であり，例えば，会議のような用途に利用することができる．実際に，MBONE[18] では世界規模の音声会議の中継を行なっており，大きな成果を挙げている [19]．会議システムに代表されるアプリケーションは実時間性が重要な要素となる [1]．しかしながら，グループ通信の性質を持つものには，実時間性よりも信頼性が大きく影響するアプリケーションが存在する．例えば，ソフトウェアの配布や電子掲示版，メーリングリストなどである．現在のアプリケーションの状況を見ると，前者の実時間性が必要なアプリケーションは数多く作成されているのに対して，後者の信頼性が必要なアプリケーションはほとんど見られない．この原因は，現在のマルチキャスト機構にあると考えられる．現在，マルチキャストは IP Multicast[20] を主体にしており，信頼性を保証しない．そのため，信頼性が必要なアプリケーションを作成する場合は，アプリケーションレベルで信頼性の確保を行なう必要がある．このことが信頼性の必要なアプリケーションを実現する障害になっている．本章では，信頼性の必要なグループ通信アプリケーションを広域ネットワークで実現するためのトランスポート層マルチキャストプロトコルを提案する．

3.2 トランスポート層マルチキャストプロトコルの目標

トランスポート層マルチキャストプロトコルの目標を明確にするために，既存のプロトコルを 3 つ調べ，問題点を指摘する．

3.2.1 既存のプロトコル

- RM (A reliable internet multicasting scheme)

  RM[21] では送信者がマルチキャストを行うにあたって，自分を根としたマルチキャスト木を作る．木の構築が行なわれた後に，その木に対してマルチキャストを行う．木の途中の計算機は親から受け取ったパケットを自分の子に転送する．最終的にマ

ルチキャストパケットは受信者である葉にたどり着き，葉になっている受信者はその親に ACK を返す．途中の計算機は子からの ACK を待ち，まとめて親に返す．

再送パケットの情報は ACK を返すときに同時に送られる．送信者はパケットを送る際にパケットに通し番号を付ける．受信者はパケットに対する ACK を返す場合に，それまでに受信したパケットの通し番号を調べ，連続した最大の番号を ACK パケットに含めて返信する．途中の計算機は ACK パケットに含まれる受信パケットの番号を調べ，その最小値を親に返す．最終的に送信者は，全員が受信に成功した最大のパケットの通し番号を得る．次のマルチキャスト時には，その通し番号がつけられているパケットから先のパケットが送信される．

- MTP (Multicast Transport Protocol)

  MTP[22] は受信に失敗した受信者が，送信者に NACK (Negative Acknowredgement) を返すことで紛失したパケットの再送を行なう．マルチキャスト木の構成とパケットの転送機構はネットワーク層に任せている．

  MTP では送信者を決定するためにトークンを用いる．トークンはマスターが管理し，マルチキャストを行いたい場合は，マスターに送信要求をだし，マスターからトークンをもらわなければならない．MTP は NACK を用いたプロトコルであるため，送信したパケットを廃棄する時間が判断できない．そこで，マスターがグループに参加した時点でパケットを廃棄する時間を静的に決定する．

- RMP (Reliable Multicast Protocol)

  RMP[23] はトークンリングを使ったプロトコルである．また，ACK と NACK を併用して紛失パケットの検出を行なう．MTP と同様，RMP も木の構成とパケット転送機構はネットワーク層に任せている．RMP では，はじめにグループに参加しているメンバのリストを作る．送信者はグループに対してマルチキャストを行い，その時点でトークンを保持しているメンバがマルチキャストで ACK を返す．この時使われるマルチキャストは信頼性を保証していないもので良い．ACK パケットには順序づけのための番号がつけられ，受信者はその番号を使ってパケットの順序づけを行う．

  トークンを渡す際には，直前に受信したパケットに対する ACK パケットにつけられた番号と同じ番号をトークンにつけて次のメンバに渡す．また，メンバはトークンを受け取る前に，トークンにつけられた番号より若い番号のついているパケットを確実に受信しなければならない．受信に失敗している場合は再送を要求する．これにより，一旦トークンを手放して次にトークンを受けとった時には，(トークンについている番号) − (メンバの数) 個前までの再送用に保持していたパケットを廃棄することができる．

  再送は NACK を使って行われる．受信者は ACK についてくる番号を調べ，それまでに受信したパケットの番号との間に空きがあった場合は，パケットの受信に失

敗したと見なす．NACK はその時点でトークンを持っているメンバにユニキャストで送られ，ユニキャストで再送される．

3.2.2　既存のプロトコルの問題

通信の信頼性を確保するには，紛失したパケットを検出し再送することが必要である．紛失したパケットの検出は ACK や NACK を用いて行なわれる．前節で述べたプロトコルも ACK か NACK もしくはその両方を使って信頼性のある通信を実現しようとしている．

マルチキャストで ACK を使おうとした場合，すべての受信者が ACK を返送する可能性があるため，ACK パケットが氾濫してしまう恐れがある．また，送信者はどの受信者が受信に失敗したかを知るために，すべての受信者を把握しておかねばならない．メンバが動的に入れ替わる広大なインターネットでこの条件を満たすのは難しい．そのため多くの ACK を用いたプロトコルでは，受信者からの ACK を送信者に返送する際に，途中計算機が同じパケットに対する ACK を集約して返送する．実際，RM ではそのようにして ACK パケットの氾濫を防いでいる．しかしながら，この方法では経路上のルータが状態を保持しておかねばならなくなる．そのため，状態を保持している計算機に障害が生じた場合の回復が困難になる．特に広域ネットワークでは途中の経路が常に安定しているとは考えられない．

次に NACK を用いる場合を考える．NACK を用いることの最大の利点は，ACK の時に問題になった，パケットの氾濫とメンバ管理の必要性がなくなることである．反面，NACK では一旦送信したパケットの保持時間が大きな問題となる．前節の MTP では保持時間を静的に決定し，通信が終了するまでその値が変化しない．このため，MTP はパケットの伝送遅延が大きい場所との通信が困難になっている．

また，MTP と RMP はメンバ管理を行なわなければならない．ACK のところで問題にしたように，メンバ管理を行なわなければならないことは大きな制約である．

表 3.2.2 にそれぞれのプロトコルの簡単な評価を示す．表中の○は要求される機能を満たしていることを示し，×は満たしていないことを示している．メンバ管理と再送パケットの保持時間はプロトコルの広域性に影響を与える．そして，途中計算機の状態保持は，下層の経路の変化に対する強さに影響する．ネットワークが広域になれば下層の安定度も低くなると考えられる．よって下層に対する強さは広域性にも影響を及ぼす．この結果，既存のプロトコルは局所的なネットワークでは動作すると考えられるけれども，広域ネットワークにおけるトランスポート層マルチキャストプロトコルとしては不十分であるといえる．

3.2.3　プロトコルの目標

前節で既存の信頼性のあるマルチキャストプロトコルの問題点が朗かになった．広域ネットワークで信頼性のある通信を行なうためには，前節で述べた問題点を解決したプ

表 3.1: 既存のプロトコルの評価

	RM	MTP	RMP
メンバ管理の必要	なし ○	あり ×	あり ×
再送パケット 保持時間	明確 ○	不明確 ×	明確 ○
途中計算機が 状態保持	行なう ×	行なわない ○	行なわない ○

○：要求を満たす
×：要求を満たさない

ロトコルが必要である．

広域ネットワークで動作するためのトランスポート層マルチキャストプロトコルが満たすべき特長は、次のようになる．

- メンバを把握しないでよい．
- 遅延の大きい受信者を考慮する．
- 下層の状態に依存しない．

このような特長を備えるためには，次のような条件を満たす必要がある．

1. NACK を用いる．

   ACK を用いると ACK パケットの氾濫を防ぐために，どうしても途中の計算機が ACK を集約してしまう．前項の条件を満たすためには NACK を用いる必要がある．

2. 送信者毎の再送パケット保持時間設定．

   遅延の大きな場所に存在する受信者と高い信頼性を持って通信するために送信者毎にパケット保持時間を設定する．

3. End to end の通信．

   マルチキャスト経路の途中の計算機が状態を保持しない．

3.3 信頼性の定義

本節ではマルチキャストにおける信頼性を定義する．最初にセッションという用語を新たに定義する．セッションとはアプリケーション層から見て意味のあるデータの単位である．例えば一通のメールはセッションの例である．そして，マルチキャストにおける

信頼性とはセッションの完全性であると定義する．つまり，トランスポート層は送信者が送信したセッションと完全に同じセッションを受信者側のアプリケーションに渡さなければならない．無論，途中経路の障害により，必ずしも完全なセッションをアプリケーションに提供できるとは限らない．その場合は完全なセッションを提供しない代わりに受信に失敗したことを知らせる．これをアプリケーション層での再送要求に利用する．

3.4　プロトコルの概要

本節では考案したトランスポート層マルチキャストプロトコルの動作を述べる．3.4.2節で述べるが，本プロトコルはリソースリザベーションと送信者のマルチキャストグループへの参加を仮定している．そこで，ネットワーク層プロトコルに 2章で述べた SnowCristal [24] を使うものとして議論を進める．無論，SnowCristal と同様の機能を提供するプロトコルであれば，SnowCristal の代用も可能である．

3.4.1　用語の定義

最初に本章で使う用語と, その意味を定義する．

送信者 – マルチキャストグループに対してデータを送信する計算機．同時に受信者でもある．

受信者 – マルチキャストグループに参加して送信者が発したデータを受信する計算機．

プレゼンテーション – 第一送信者がグループに参加してからグループを脱退するまでの期間．

プレゼンテーションマネージャ – プロセスに世界中で一意なマルチキャストアドレスと計算機毎に一意なポート番号を割り当てるサーバ．

コネクション – マルチキャストグループと送信者の間の接続．N 人の送信者が存在するマルチキャストグループには N 本のコネクションが存在する．

セッション – アプリケーションプログラムが送信するデータの一単位．例えば, 一通のメールや一枚の画像に相当する．

セッション番号 – セッションにつけられる通し番号．

セグメント – トランスポート層が取り扱う最小のデータの単位．セッションはセグメントに分割されて, 相手側のトランスポートに転送される．

セッション内番号 – セッション毎にセグメントにつけられる通し番号．

コアルータ – SnowCristal[24] で定義される, マルチキャストアドレスの経路情報とマルチキャストグループ木内の最小 MTU(Maximum Transmission Unit) を流すルータ．

3.4.2 前提事項

本論文で考えるマルチキャストプロトコルは, 現在の Internet で実現されていないいくつかの機構を仮定している．本節ではこれから述べるマルチキャストプロトコルが前提としている事項を述べる．

1. **ネットワーク層における資源予約．**

   本プロトコルは NACK ベースのマルチキャストプロトコルである．3.2.2 節で述べたように, NACK を使う場合には再送パケットの保持時間の決定が難しい．本プロトコルでは再送パケットの保持時間を送信者から最も遠い受信者までのパケットの往復時間 (ラウンドトリップタイム, RTT) を基準にして決定する．そのためには RTT があまり変動せずに, 安定している必要がある．ネットワーク層で資源予約を行なうことによりパケットの遅延がほぼ一定に保たれ, RTT の変動が抑えられる．

2. **送信者のマルチキャストグループへの参加．**

   既存の多くのマルチキャストプロトコルでは, マルチキャストを行なう送信者は必ずしもマルチキャストグループに参加している必要はなかった．しかしながら, この機能は有効な活用法がない．むしろ, 送信者も必ずマルチキャストグループに参加しているほうが都合がよい．本プロトコルでは送信者は必ずマルチキャストグループに参加しなければならない．これによってネットワーク層から制御パケットを得ることが可能になるからである．

   例えば，ネットワーク層からマルチキャスト木の最小 MTU の情報が受信できる．マルチキャストに限らず, 異なる通信媒体を相互接続する場合は MTU が問題になる．もっとも，ユニキャストの場合はたとえフラグメントが生じたとしても，パケットの再組み立てを行なうのは受信計算機 1 台だけですむ．しかしながら，マルチキャストでパケットのフラグメントが起きると, 末端のすべての受信者がフラグメントパケットを再組み立てしなければならない．そのような計算機資源の浪費を回避するために, 送信者は現在のマルチキャスト木のリンクの中で, 最も小さい MTU にパケットの大きさを調整して送信するのが望ましい．MTU の値はネットワーク層が調べ, 定期的にネットワーク層のコアルータからマルチキャストグループにマルチキャストされる．マルチキャストグループのメンバは時々刻々と入れ替わるため, MTU もそれにつれて変化する．送信者は, 自分がパケットを送信しようとする時点での最適な MTU を知るためにマルチキャストグループに参加してコアルータからの MTU 情報を受信しなければならない．

3.4.3 プレゼンテーションの開始

アプリケーションプログラムはプレゼンテーションマネージャに世界中で一意なマルチキャストアドレスと計算機毎に一意なポート番号を割り当ててもらう．もしこのアプリケーションプログラムがマルチキャストグループの最初のメンバであった場合は，ネッ

トワーク層プロトコルが経路情報を流し始める．これがプレゼンテーションの開始となる．同時に，アプリケーションプログラムとマルチキャストアドレスの間にコネクションができた状態になる．

3.4.4　セッションの開始と終了

セッションとはアプリケーションプログラムにとって意味のある最小のデータの単位である．アプリケーションプログラムはマルチキャストグループにデータの送信をはじめる前に, セッションを開始する．そして, アプリケーションプログラムはひとつのセッションの転送が終了したら, トランスポート層にセッションの終了を通知する．セッションの終了が行われると, トランスポート層はそれ以降のアプリケーションプログラムからのデータを, 新たなセッションのデータであるとして処理する．セッションにはセッション番号と呼ぶ通し番号がつけられる．プレゼンテーションが確立した直後のセッション番号は 0 で, アプリケーションプログラムがセッションを終了するたびにセッション番号が 1 加算される．

3.4.5　ソケットのオープンとクローズ

セッションの開始が完了した後, アプリケーションプログラムは, 割り当てられたマルチキャストアドレスとポート番号に対して, ソケットを作成する．実際のデータは, このとき得られたソケット記述子に対する, read,write システムコールによって行われる．ソケットに書き込まれたデータ (セッション) はトランスポート層によってセグメントに分割される．トランスポート層はセグメントにセッション毎の通し番号をつける．この通し番号をセッション内番号と呼ぶ．セッション内番号は 0 から始まり，1 ずつ加算されていく．また, セグメントにはそのセグメントが送信された時点での RTT の値が同時に記録される．これは受信者が NACK を返送するときに，NACK パケットが紛失したことを検出する際のタイムアウト時間の目安に使う．ひとつのセッションの送信が終了したら, アプリケーションプログラムはソケットを終了する．最後に, アプリケーションプログラムがすべてのデータの送信を終了したら，セッション内番号を -1 にしたコネクションの終了を示すダミーセグメントを定数個送信する．

なお, セグメントの大きさは, 参加しているマルチキャストグループ内で最も小さな MTU の値に収まるように設定される．MTU 情報はネットワーク層から得ることができる．

3.4.6　RTT 計測

送信者はマルチキャストグループに参加したとき, データの送信をはじめる前にまず RTT の計測を行う．送信者は RTT の値を使って再送パケットの保持時間を決定する．送信者から最も遠い受信者までの RTT を計測する方法は 4 つ考えられる．

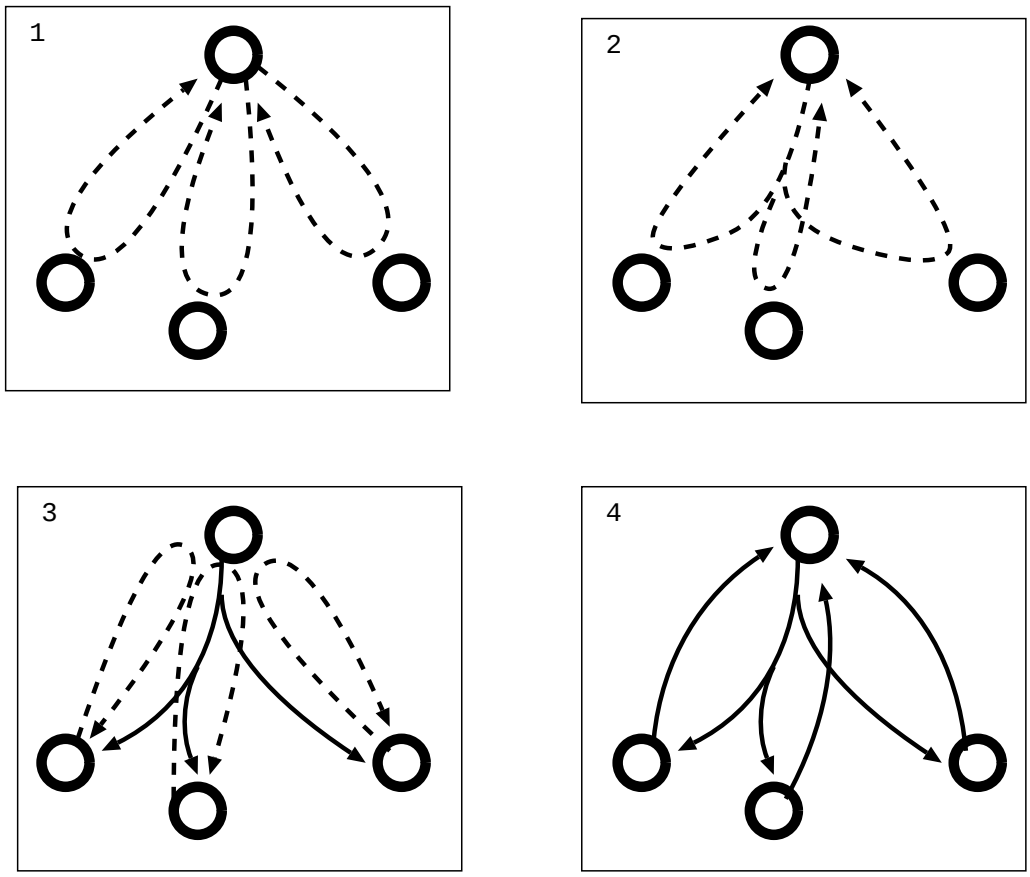

図 3.1: RTT 計測

1. **送信者が個々の受信者に対し RTT を計測する方法 (図 3.1の 1)**

   送信者はすべての受信者に対して時刻情報を記録したパケットを送信する．受信者はパケットを受信したら即座に送信者に対して返送する．パケット送信時刻と受信者からの応答の到着時刻との差から RTT を求め, その最大値を新しい RTT とする．

   この方法には問題点が 3 つある．

   (a) 全受信者にパケットを送信するために, 送信者はすべての受信者を把握しておかなければならない．

   (b) 送信者がすべての受信者の RTT を計測するため，送信者に重い負担がかかる．

   (c) 全受信者からの返事が送信者に対して返ってくるので, 送信者が属するネットワークが輻輳を起こしてしまう可能性がある．

2. **送信者がマルチキャストグループに RTT 計測パケットを送信する方法．(図 3.1の 2)**

この方法は, 送信者がそれぞれの受信者にパケットを送信するのではなく, 送信者が参加しているマルチキャストグループに対してパケットを送信する方法である．後の動作は 1. と変わらない．

この方法では 1. の時に問題になった, 送信者がすべての受信者を把握しておかなければならないという欠点がなくなる．しかしながら, その他の欠点は残ったままである．

3. **受信者が RTT を計測する方法．(図 3.1の 3)**

   これは RTT の計算の負担を, 送信者から受信者に移すことができる方法である．送信者は参加しているマルチキャストグループに RTT 計測を要求するパケットを送信する．受信者は送信者からのパケットを引きがねにして, 送信者に対して RTT を計測する．その結果を送信者に返す．

   送信者は受信者を把握しておく必要がないし, 受信者からの RTT 計測に対して返送するだけでよいので, 負担も軽い．ただ, 輻輳の問題はまだ残ったままである．

4. **送信者との時間のずれを利用する方法．(図 3.1の 4)**

   まず, この方法は全計算機で時間が同期していることを仮定している．送信者は参加しているマルチキャストグループに時刻情報を記録したパケットを送信する．パケットを受信した受信者は, 記録されている時刻と現在時刻との差を取り, その 2 倍の値を RTT として送信者に返送する．現時点では NTP(Network Time Protocol)[25] を用いて時間の同期をとることができる．がある．また，将来的に通信衛星を使って時間を同期させる方法などが普及すれば，すべての計算機での時間の同期も可能であろう．

   この方法を使えば, 送信者の負担はさらに軽くなる．しかしながら, 輻輳の問題はこの方法でも解決しない．

4 つの方法に共通していえることは, どれも送信者が属するネットワークに輻輳を起こしてしまう危険性を持っているということである．これは 3 か 4 の方法に改良を加えることによって解決可能である．3 および 4 で, 送信者は RTT 計測 (開始) パケットに最大遅延時間を記録しておく．最大遅延時間とは, 送信者に対して, RTT 計測の返事をしない範囲を示すパケット遅延時間である．3 と 4 の方法では, RTT を計算するのは受信者側である．そこで, 計算の結果が最大遅延時間より短かった場合は, 送信者に返事をしないようにする．最大遅延時間は送信者が保持している, 現時点での RTT に設定する．こうすることで, 余分な返事が返ってくることを防ぎ, 輻輳を回避する．ただし, この方法では RTT が増加する一方なので, 例えば，定期的に最大遅延時間を RTT の半分に設定するなどして RTT の増加を抑制する必要がある．

3.4.7 再送の機構

本プロトコルは NACK ベースのマルチキャストプロトコルである．一旦送信したセグメントは, 受信者からの NACK に答えるためにある程度保持しておかなければならない．この保持時間は送信者から最も遠い受信者までの RTT の定数倍となる．

受信側のトランスポート層は, ネットワーク層からセグメント単位でデータを受け取る．セグメントのヘッダには送信者の IP アドレス, ポート番号, セッション番号, セッション内番号が記入されている．セグメントの欠落は同一 IP アドレス, ポート番号, セッション番号のセグメント群の中で, セッション内番号を調べることによって検出される．NACK ベースのプロトコルではデータの終わりの欠落の検出が問題になる．本プロトコルでは, セッションの最後のセグメントが欠落したことの検出を, セッションクローズ時に送信される, セッション内番号が -1 となっているダミーセグメントを利用して行う．

受信者はセグメントの欠落に気付いた時点で, セグメントのヘッダに記録されている送信者に対して NACK を送信する．欠落したセグメントは (IP アドレス, ポート番号, セッション番号, セッション内番号) の組で一意に決まる．受信者は送信されてきたセグメントに記録されている RTT の値を参照して NACK のタイムアウト時間を決め, NACK 送信と同時にタイマを起動する．タイマが切れる前に再送が行われなかった場合, 受信者は NACK が紛失したとみなし, NACK を再送する．定数回の NACK 送信に対して期待するセグメントの再送が行われなかった場合, 受信者は送信者が再送できない状態になったと見なし, 対応するセッションを不完全セッションとする．不完全セッションが発生したら, トランスポート層は不完全セッションの送信者の IP アドレスをアプリケーションプログラムに知らせ, アプリケーション層での再送に備える．

送信者は NACK 受信時の再送用にプロセス毎にキューを作り, 送信したセグメントを保持しておく．受信者からの NACK を受信すると, キューの中から NACK パケットに記録されている (IP アドレス, ポート番号, セッション番号, セッション内番号) の中で, ポート番号, セッション番号, セッション内番号の等しいセグメントを見つけ, 二度目の再送に備えてキューの先頭に追加したの等しいセグメントを見つけ, 二度目の再送に備えてキューの先頭に追加した後に再びマルチキャストする．再送にユニキャストを用いない理由は，個々の相手計算機の情報を持たずに再送を行なうためである．ただ，受信した NACK すべてに答えてしまうと不必要な再送を行なってしまう可能性がある．そこで，NACK を受信しても即座に再送を行なうのではなく，時間をおいて再送する必要がある．

3.4.8 セグメント送信間隔

マルチキャストグループのメンバの数は不定である．送信者はひとりかもしれないし, 場合によっては 100 人の送信者が参加しているかもしれない．受信者から見た場合, マルチキャストグループに送信されるセグメントの間隔は一定であるべきである．よって, 送信者はマルチキャストグループ内の送信者の数を考慮して, セグメントの送信間隔を変化させる必要がある．送信者のセグメント送信間隔の決定に関しては, 現在ふたつの候補がある．

ひとつめは, 送信者の数に応じて決定された送信間隔をプレゼンテーションの始まりから終わりまで使う方法である．マルチキャストグループに存在する送信者の数は, 資源予約が行われる際に同時に決定されるとする．そこで, グループへのセグメント送信間隔を送信者の数で割った値を各々の送信者のセグメント送信間隔とするのである．この方法の問題は, 実際の送信者の数を反映した値に設定できないことである．例えば, 100 人のメンバによって構成されるメーリングリストを考える．このマルチキャストグループには, 潜在的に 100 人の送信者が存在することになる．しかしながら, メーリングリストの特性を考慮すると, 同時に 100 人からのメールが送信されるとは考えられない．送信者のセグメント送信間隔を, 送信者の数で割ることによって決定すると, このような状況のときの性能が著しく落ちることになりかねない．

ふたつめは, 現在送信中の送信者が他の送信者からのセッションを監視して, 実際に送信している送信者の数の見積りを取る方法である．送信者は同時に受信者でもあるので, グループ内のすべての送信者からのセッションを受信している．送信間隔は, グループへの送信間隔を, 現在の実際の送信者の数で割ったものに設定する．こちらの方法の問題は, 送信者が結果的にすべての他の送信者のリストを保持しなくてはならなくなることと, 状況によっては, リソースリザベーションで確保した帯域幅よりも多くのデータがネットワークに送出される可能性があることである．帯域幅を越えたデータの送出はパケットの欠落を引き起こし, 再送パケットの数を増加させる．

3.5　考察

本論文では広域性ネットワークで動作する信頼性のあるマルチキャスト通信についての研究と, 新しい信頼性のあるマルチキャストプロトコルについて考えた．信頼性のある通信を行うためには, ACK や NACK を用いた紛失パケットの再送を行う必要がある．ACK を使った通信では, 受信者からの確認応答を使って送信者が送達確認を行うことにより, 確実な転送の機能と明確な再送用の保持パケットの廃棄時間を得ることが可能である．しかしながら, 送信者が全受信者を把握し, すべての受信者からの ACK を待たなければならない．NACK は送達確認を受信者側に任せることにより, 送信者がすべての受信者を把握する必要性をなくしている．その反面, NACK 受信時の再送のために, どのくらいの時間パケットを保持してよいかが判断できなくなったり, また, 通し番号によって紛失パケットの検出を行っているために, 末尾のデータの欠落が検出できないといった欠点が存在する．

3.2.3節で述べたように, マルチキャストの性格を考えた場合 ACK の持つ利点よりも, NACK の持つ利点の方が適している．そこで, 本プロトコルでは信頼性のある通信を行うために, 紛失したパケットの検出に NACK を用いた．NACK を使う場合の問題である, 末尾のパケットの紛失の検知はセッション番号という概念とダミーセグメントの送信を使うことによって行う．また, 再送用にパケットを保持しておく時間の問題は, 送信者毎の RTT の計測によって回避した．

NACK の利点を活かし, メンバの把握の必要性を排除した本プロトコルは, Internet の

ような広域ネットワーク上で充分動作すると期待される．

また, 多数の受信者が存在する通信での信頼性はこれまで議論されることはなかった．本論文で述べたセッションという概念は, この多人数での通信におけるデータの区切りと, 保証すべき通信内容を明確にする．セッションの導入により, アプリケーション層は意味のある単位でのデータ送信が可能となった．セッションはマルチキャストにおける信頼性のあるデータ通信の基本的な概念になると思われる．

3.6 今後の課題

本プロトコルはいくつかの仮定の下に設計されている．まず, 今回仮定した事項を解決しなければならない．特に実現が難しいと思われるのは, RTT 安定化のためのリソースリザベーションによる帯域確保である．今後は本プロトコルの実装を行ない，再送パケット保持時間やダミーパケットの数などの最適値を，実験を通して導出する予定である．

第 4 章

経路制御技術

4.1 はじめに

ここでは，広域ネットワークにおける 1 対多型・放送型の通信媒体を効率よく利用するための通信アーキテクチャを，既存のものと調和して実行できるよう設計した WMA (WIDE Multicast communication Architecture) について述べる．

この通信アーキテクチャは，既存のインターネットで行なわれているユニキャスト通信におけるアドレス体系や経路制御の体系を乱すことなく，マルチキャスト型の通信を広域ネットワークで利用することを目的として設計している．これは，既存のインターネットで用いられている 1 対 1 型の通信媒体に，広域のブロードキャスト型の通信媒体を加えたネットワークアーキテクチャであり，アドレスの割り当て方法，経路制御の方式、トランスポートプロトコル，アプリケーションプロトコルなどを総合的に扱っている．

広域分散システムにおいては，電子掲示板システムや一般に公開されたファイルの転送という機能を介して，ローカルエリアネットワークの枠を越えて，多くの情報を共有している．また，実時間的な会話型応用ソフトウェア，メーリングリストによる電子メールのグループへの配送といった情報共有や情報配送が非常に活発に行なわれており，インターネットのような広域分散システムにおいて非常に重要な機能となっている．このような情報の共有の環境では，マルチキャスト通信，つまり 1 対多型のグループ通信を支援することが重要なことである．既存のネットワークでは，OSI モデルにおける第 3 層，つまりネットワーク層における 1 対 1 型の通信基盤の上に，第 4 層以上の機能として 1 対多型の通信を構築している．

しかしながら，効率の良い 1 対多型の通信を上位層で行なうためには，ネットワーク層が 1 対多型の通信を支援することが必要である．本研究においては，ネットワーク層での 1 対多型の通信方式を利用した広域ネットワークでの通信アーキテクチャについて提案した．この方式では，広域マルチキャストバックボーンと呼ぶ仮想的なマルチキャスト通信網を定義し，それに基づき，広域で適用できる新たな経路制御プロトコル HDVMRP (Hierarchical Distance Vector Multicast Routing Protocol) を開発し実験した．さらに広域ネットワークでの通信アーキテクチャに有用なブロードキャスト型の通信媒体を利用するための通信アーキテクチャを提案した．既存のデータリンク層と整合性をもつようにインタフェースを拡張する方式が有効であることを示し，実例として通信衛星を用い

た広域のブロードキャスト型の通信方式の実験，実証を行なった．

HDVMRP は，現在 The Internet で実験的に使われている DVMRP(Distance Vector Multicast Routing Protocol)[5],[7] の方式に階層化を導入し，拡張性の問題を解決し，効率良いマルチキャスト通信を行なえるようにしたものである．ここでは，通信衛星のような広域の放送型通信媒体の利用を可能とするようなアーキテクチャを取り入れている．このプロトコル体系では，広域ネットワークを地域 (Region) と呼ばれる領域に分割し，地域内の経路制御は既存の DVMRP に準じた方法で行ない，地域間は各地域を管理する地域マスタと呼ばれるホスト同士がマルチキャストデータグラムの中継を行ないながら，それらの間で経路情報交換して行なう．このとき、広域マルチキャストバックボーンを通して各地域のマスタに対してマルチキャストデータグラムが送られる．広域マルチキャストバックボーンは，既存の地上系のネットワークにおいて IP データグラムのカプセル化等により行なわれるトンネリングを用いて構築される仮想的なマルチキャスト網や，広域の放送型通信媒体等から構成され，有効な経路が自動的に選択され使われる．図 4.1に HDVMRP と広域マルチキャストバックボーンの概念を示す．

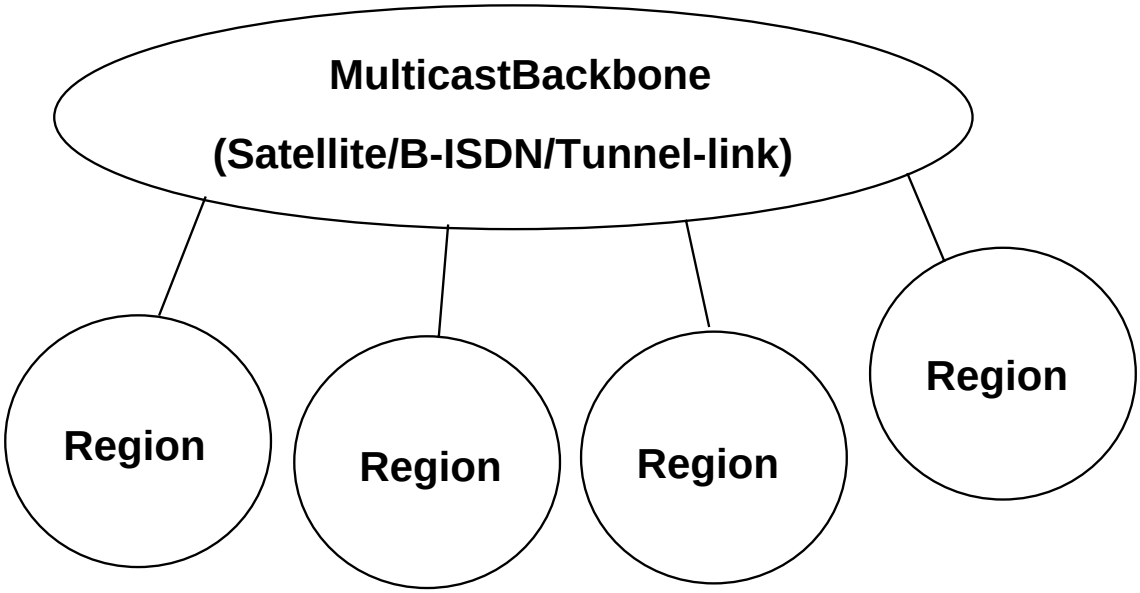

図 4.1: HDVMRP の概念図

また，広域の放送型通信媒体を，既存の汎用のインターネットアーキテクチャで利用するために，既存の放送型データリンク層と整合をとったインタフェース装置を開発した．図 4.2(a) は，従来のプロトコル階層を示しており，図 4.2(b) が本アーキテクチャでのプロトコル階層を示しいる．従来用いられている放送型通信媒体の機能を持つネットワークとしてイーサネットがあり，広域の放送型通信媒体をイーサネットのインタフェースに合わせるために，両者の間を整合性よく接続するインタフェースを定義した．このインタフェースは，広域の放送型媒体を一つのイーサネットセグメントとして見せている．

このインタフェースの実現により，イーサネットに接続された計算機が，従来のイーサネットアドレスとイーサネットフレームを用いて，イーサネットと同様に広域の放送型通信媒体を用いた通信が可能となった．

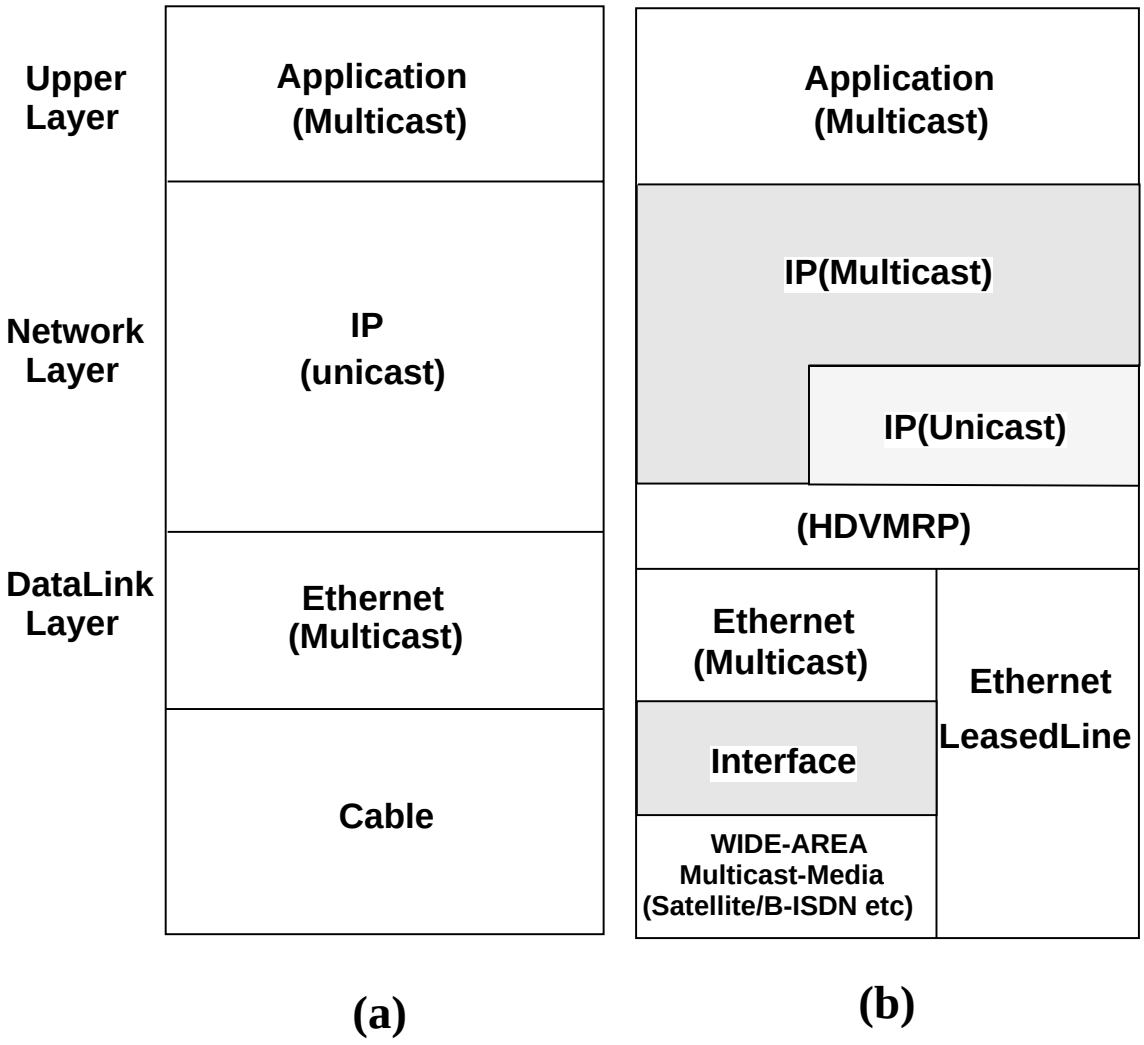

図 4.2: WMA 階層モデル: (a) 従来のアーキテクチャの階層モデルと，(b)WMA における階層モデル

既存の放送型データリンク層と整合したりインタフェースを実現することにより，遅延，信頼性，単方向性といった問題点を隠蔽することが可能となり，1 対 1 型通信や 1 対多型通信のの経路制御に関して，既存の経路制御プロトコルをそのまま適用することが可能となった．本研究では，通信衛星として CS 衛星 (スーパバード B) を広域の放送型通信媒体として使用した．

4.2 ユニキャストデータグラムの経路制御

本研究における衛星通信媒体は単方向の通信媒体であったり，双方向の通信媒体であったりする．単方向の通信媒体の場合には，従来のユニキャストのデータグラムの経路制御にも工夫が必要であり，ここでは，図 4.3におけるユニキャストデータグラムの経路制御について述べる．

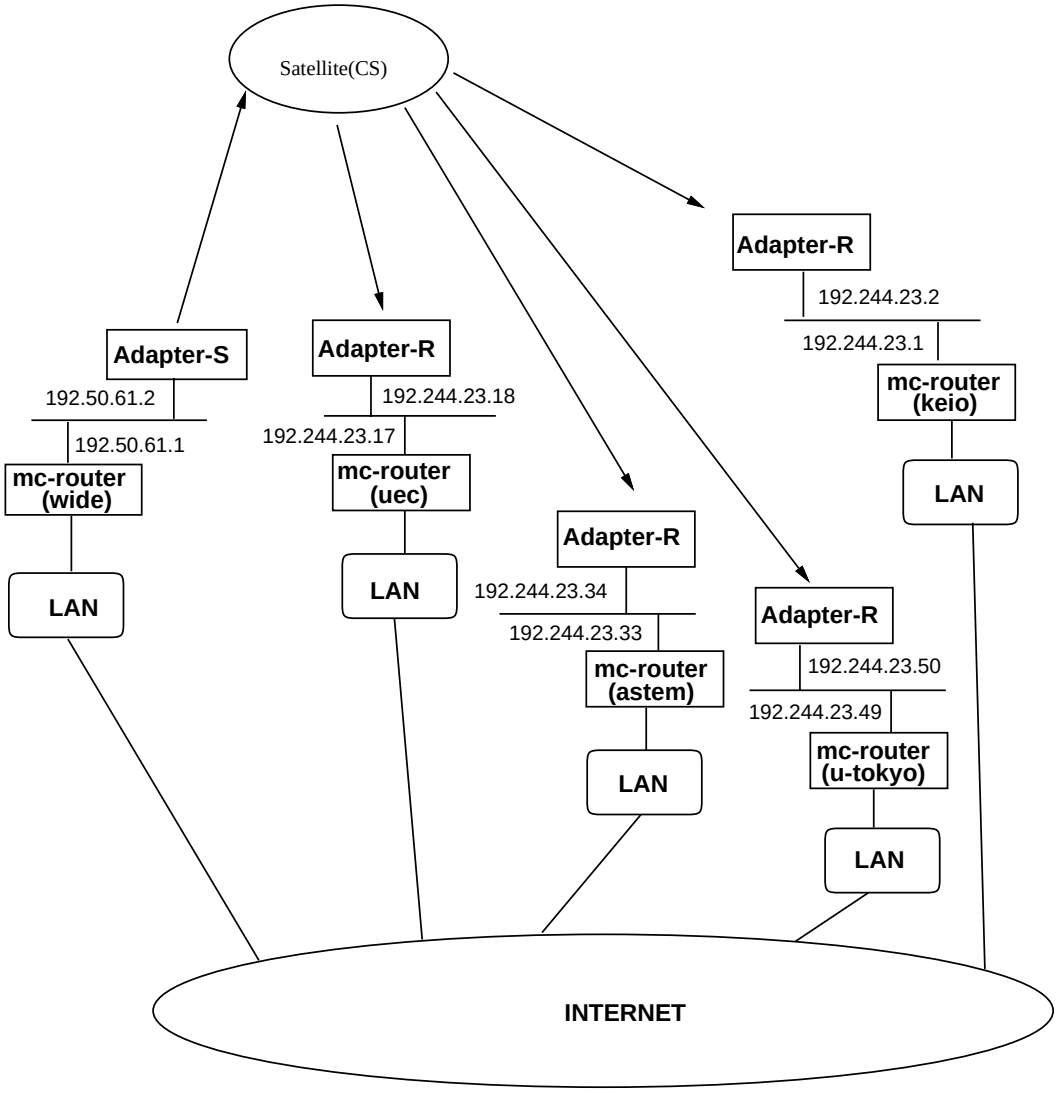

図 4.3: ユニキャストデータグラムの経路制御

図 4.3では，mc-router(wide) が衛星通信を行なうための衛星への打ち上げ局にあるルータであり，その他の mc-router は，受信装置が設置していある 4 組織，慶應義塾大学 (keio)，電気通信大学 (uec)，東京大学 (u-tokyo)，京都高度技術研究所 (astem) にあるルータである．受信側は 4 つの組織でクラス C のインターネットアドレス (192.244.23.0)

を共有しており，サブネットとして用いている．mc-rouer(wide) から，インターネット (Internet) 全体に対して，192.244.23.0 および 192.50.61.0 のネットワークアドレスの経路を広告する．

この仕組みにより，ネットワーク 192.244.23.0 が広域にまたがる一つのイーサネットであるかのように他からは見え，Internet 上のホストは，このアドレスを指定することにより衛星通信回線を用いて，192.244.23.0 上に位置するホストに IP データグラムを送ることができる．192.244.23.0 上のホストからほかのホストへのデータグラムは，従来どおりの経路，つまり専用回線，公衆網等から構成されているネットワークを介して通信する．

問題点は，ある組織内，例えば慶應大学の LAN(Local Area Network) からイーサネット等の高速なネットワークで接続されている 192.244.23.0 のサブネットのホストに到達するためにも通信衛星経由になってしまうという問題点がある．これは，その組織内の mc-router への遠隔ログインしてから必要な操作を行なうことによりある程度回避できる．さらに，192.244.23.0 上に衛星通信アダプタとルータの二つのホストしか存在しない場合には，実際上の問題はない．

ここに示した経路制御方法は，既存の経路制御プロトコルと完全に互換であり，ソフトウェアの変更を必要とせずに，衛星通信を利用することができるのが大きな特徴であり，その優位性を示している．

4.3　マルチキャストデータグラムの経路制御

マルチキャスト通信を行なうための経路制御プロトコルには DVMRP (Distance Vector Multicast Routing Protocol) [7] があり，マルチキャストアドレスのグループのメンバシップを制御するためには RFC1112 に IGMP (Internet Group Membership Protocol) [10] が規定されている．

通信衛星を用いて広域でマルチキャスト通信を実現することを考えた場合，DVMRP はすべてのネットワークアドレスを広域ネットワーク全体で交換するので，経路制御情報の量が発散してしまい，スケーラビリティがない．そこで，次のことを目標に新たな経路制御プロトコルの設計を行なった．この新しい経路制御プロトコルを階層化 DVMRP(HDVMRP) と呼ぶ．このプロトコルの設計目標はつぎのとおりである．

(1) 通信衛星チャネルの有効利用をはかり，地上のリンクのトラフィックをできるだけ減らす．

(2) 単方向の通信チャネルである通信衛星が経路に含まれていても動作する．

(3) グループが広域に広がっていても経路制御情報の交換のための通信量が発散しない．

(4) 通信衛星，衛星通信への発信局および受信局の故障に対処できる．

(5) 一般のホストのソフトウェアをできるだけ変更せず，既存の IP マルチキャストの実装で動作するようにする．

次の節から，HDVMRP によるパケットの伝播がどのように行なわれるかを述べる．

4.4 HDVMRP による平常時のパケットの伝播

4.4.1 ネットワークの地域分割

HDVMRP では，ネットワーク全体をいくつかの地域に分割する．図 4.4 では，ルート地域 (Root Region) と 地域 A から地域 D までの合計 5 つの地域にネットワークが分割されている．これにより，比較的高速の回線で接続されたネットワーク以外は通信衛星からデータを供給して回線速度の遅いリンクをマルチキャストデータグラムができるだけ通らないようにできる．

各地域には地域マスタ (Region Master(RM)) と呼ばれるルータが存在し，地域内の経路制御情報を制御する．衛星通信のような広域のマルチキャスト通信媒体がある場合には，その受信局が存在する．その場合には，地域マスタが受信局を兼ねる場合と，受信局から地域マスタにトンネリング技術により通信衛星のような広域のマルチキャスト通信媒体から受け取ったデータを送る場合とがある．どちらの場合も論理的には，地域マスタが広域のマルチキャスト媒体から直接データを受け取ることになる．地域と別の地域の境界に位置するルータを地域境界ルータ (Region Boarder Router(RBR)) と呼び，RBR 同士は直接相互接続されている．送信局が位置する地域内の地域マスタをルート地域マスタ (Root Region Master) と呼び，この地域をルート地域と呼ぶ．

4.4.2 デフォルト経路

ルート地域マスタは， HDVMRP の経路制御情報としてデフォルト経路を通信衛星を通して定期的に広告する．各受信局と地域マスタは，この経路情報を運ぶデータグラムが来ることにより，送信局および通信衛星が動作していることを知る．

4.4.3 地域内での経路制御

図 4.5に示すように，地域内にはマルチキャストデータグラムの伝播に関与する複数のマルチキャストルータ (Multicast Router(MR)) が存在し，地域内のマルチキャストルータ間では DVMRP により経路制御情報を交換する．地域マスタ (MR) もマルチキャストルータの 1 つであり，この DVMRP による経路制御情報の交換に参加する．地域内には地域マスタがデフォルト経路を広告しており，各マルチキャストルータは，このデフォルト経路及び交換している経路情報に従ってマルチキャストデータグラムの経路制御を行なう．この DVMRP の経路制御情報の伝播は，地域境界ルータ (RBR) のところで止まり，この経路制御情報は地域の境界を越えて伝播されることはない．

地域内のホストから発生したデータグラムは，そのホストをルートとして地域内に形成されたリバースパスブロードキャストツリーにしたがって伝播される．これは通常の

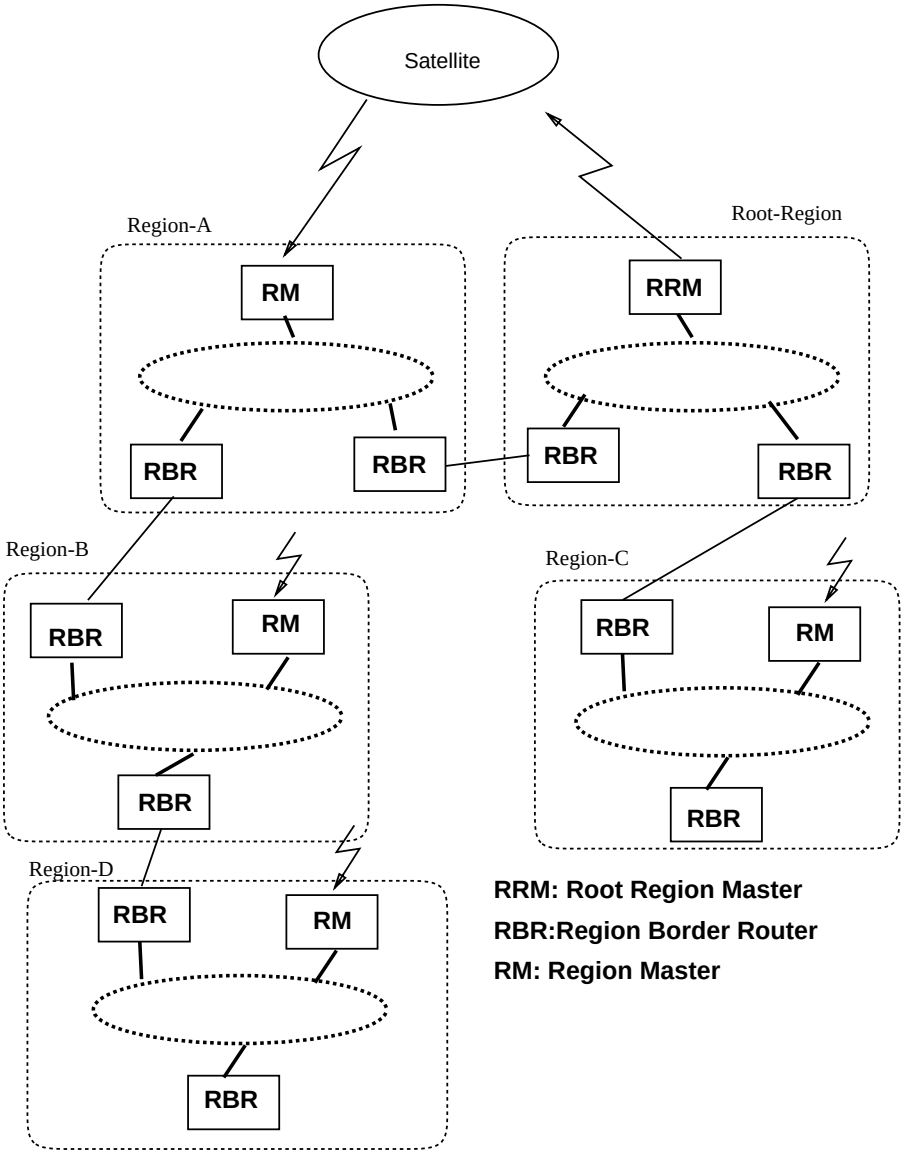

図 4.4: HDVMRP による経路制御

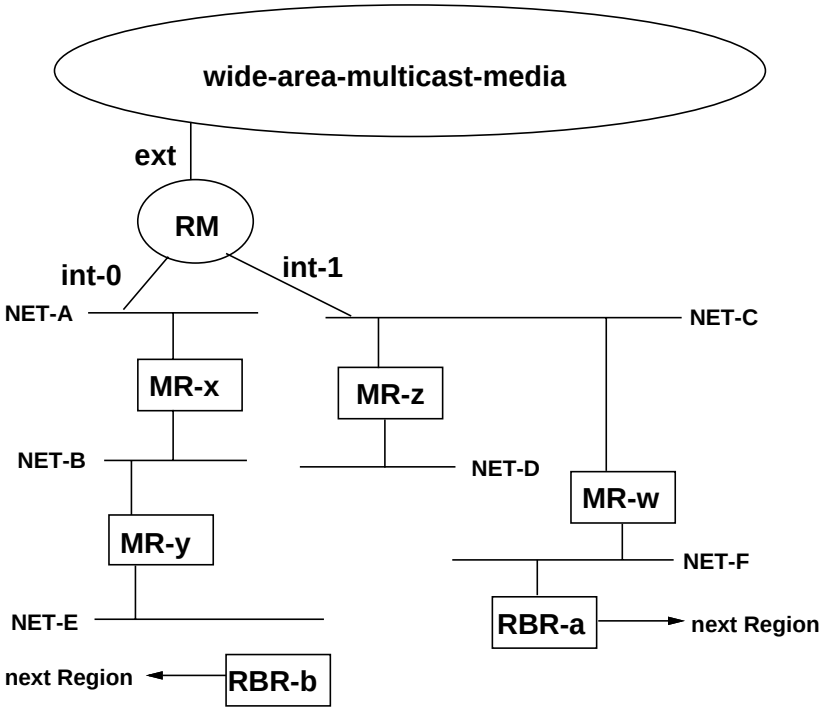

図 4.5: 地域内の経路制御

DVMRP による経路制御と同じであり，地域内のマルチキャストルータのソフトウェアは既存のソフトウェアを変更することなく用いることが可能である．

地域外で発生したデータグラムは，広域のマルチキャスト通信媒体 (図 4.5の **wide-area-multicast-media**) から到達する．この通信媒体は，物理的に広域のマルチキャストやブロードキャストを提供している場合もあれば，仮想的な媒体の場合もある．

地域内のマルチキャストルータは地域内のネットワークについてのみ経路の交換を行なっているので，リバースパスブロードキャストツリーを形成する際，地域外のネットワークを始点とするデータグラムに関してはデフォルト経路に従う．つまりデフォルト経路を広告している地域マスタをルートとするリバースパスブロードキャストツリーが形成されることになり，地域内にくまなく伝播することが可能となる．これは通常の DVMRP による経路制御と同じである．

図 4.5の地域マスタ **RM** の経路制御テーブルを表 4.1に示す．例えば，外部からきたデータグラムは始点ネットが **EXT** であるので，`ext` インタフェースからデータグラムが入ってきた場合だけデータグラムを中継する．この場合，子リンクのフラグが立っている `NET-A` と `NET-C` にデータグラムを中継すればリバースパスツリーを形成できること

がわかる.

表 4.1: 地域マスタ **RM** の経路制御テーブル

始点ネット	距離	次ホップルータ	インタフェース	子リンク		
				NET-A	NET-C	EXT
NET-A	0	RM	int-0	0	1	1
NET-B	1	MR-x	int-0	0	1	1
NET-C	0	RM	int-1	1	0	1
NET-D	1	MR-z	int-1	1	0	1
NET-E	2	MR-x	int-0	0	1	1
NET-F	1	MR-w	int-1	1	0	1
...						
...						
EXT	0	RM	ext	1	1	0

地域内，地域外，どちらのホストから発生したデータグラムでも地域内で伝播し拡散するので，最終的に地域マスタ (RM) 及び地域境界ルータ (RBR) に到達するので，地域の境界を越えてデータグラムを中継しないように地域境界ルータ (RBR) は動作する.

地域内で発生したデータグラムを地域マスタ (RM) が受けとると，広域のマルチキャスト通信媒体を用いてルート地域マスタに送ろうとする．前述したように，この広域のマルチキャスト通信媒体は，物理的に広域のマルチキャストやブロードキャストを提供している場合もあれば，仮想的な媒体の場合もある．単方向の衛星通信と地上網を組み合わせた場合には，受信局側から送信局であるルート地域マスタの地域へは，必ず仮想的な媒体として地上網からなるマルチキャストバックボーンを通ることになる.

この場合には，データグラムを IP in IP によるカプセル化，または厳密でない始点経路制御 (Loose Source Route Routing(LSRR)) によるトンネリング [7] により，直接に送信局、つまりルート地域マスタに送る．送信局はカプセル化されて送られてきたマルチキャストデータグラムを取り出して衛星通信チャネルを通して送信する.

各地域マスタがルート地域マスタの位置を知ることができるようにするために，ルート地域マスタは近隣の地域マスタにその存在を広告している．これはルート地域内の地域境界ルータとルート地域マスタとの間の通信による.

地域内では，地域マスタ及び地域境界ルータは相互に情報を交換するために別のプロトコルで通信を行なう．これは一つの特別なグループを割り当てることにより簡単に実現できる．交換される情報は,「近隣地域の地域マスタのアドレス」,「ルート地域 への地域カウント数 (地域を一つのノードと考えたときのホップカウント)」,「ルート地域マスタの IP アドレス」の 3 つである．つまり，ルート地域内の地域境界ルータは隣の 地域境界ルータに,「ルート地域マスタの IP アドレス」,「ルート地域へのホップカウントが 0

である」,「地域マスタの IP アドレス (この場合，ルート地域マスタと同じ)」という 3 つの情報をわたす.

4.4.4 地域マスタの動作アルゴリズム

地域マスタがマルチキャストデータグラムを受け取った時の処理は，表 4.2のようなアルゴリズムで行なわれる．このアルゴリズムは，後述の広域のマルチキャスト通信媒体が故障等の理由で使用できない場合も包含している.

表 4.2: 地域マスタがマルチキャストデータグラムを受け取った時の処理

```
IF データグラムのソースアドレスが地域内である THEN
        /* 地域外への転送を行なう */
        データグラムの複製を地域外に転送する;
        IF 直接広域媒体に接続している THEN
                広域媒体を通して転送する;
        ELSE IF ルート地域マスタへのトンネルがある
                ルート地域マスタへトンネルで転送する;
        ELSE IF 隣接地域マスタへのトンネルがある
                隣接地域マスタへトンネルで転送する;
        ELSE
                エラー;
        /* 地域内の残りの部分への転送を行なう */
        FOR (各インタフェースについて繰り返し)
                IF 各インタフェースの子リンクフラグが立っている THEN
                        そのインタフェースに複製を転送;
ELSE /* その他 (地域外からのデータグラムの場合) */
        FOR (各インタフェースについて繰り返し)
                そのインタフェースに複製を転送;
        IF 近隣の地域マスタとのトンネルがある THEN
                IF データグラムの始点アドレスが近隣の地域マスタではない.
                        AND トンネルの先よりルート地域に近い THEN
                        データグラムを複製し，近隣地域マスタに転送;
```

表 4.3に地域マスタの経路制御情報の交換の処理のアルゴリズムを示す.

表 4.3: 地域マスタの経路制御情報交換

SWITCH 経路制御情報の発信元
CASE 近隣の地域マスタ:
　　IF トンネル設定の要求 **THEN**
　　近隣の地域マスタへのトンネルを設定;
　　ELSE
　　エラー;
CASE ルート地域マスタ:
　　IF デフォルト経路 **THEN**
　　IF 広域マルチキャストインタフェースに到着 **THEN**
　　経路制御テーブルのデフォルト経路を更新;
　　ELSE
　　エラー;
　　ELSE
　　エラー;
CASE 地域境界ルータ:
　　IF 近接地域マスタ情報 **THEN**
　　近接地域マスタリストを更新;
　　(近隣地域マスタのアドレス)
　　(ルート地域へのホップ数)
　　(ルート地域マスタのアドレス)
　　ELSE IF リバースパス経路制御情報 **THEN**
　　経路制御テーブルを更新;
　　ELSE
　　エラー;
CASE その他 (一般のマルチキャストルータ):
　　IF リバースパス経路制御情報 **THEN**
　　経路制御テーブルを更新;
　　ELSE
　　エラー;
IF デフォルト経路が広域マルチキャストインタフェースから
　　周期的に到着している **THEN**
　　continue;
ELSE
　　ルート地域に近い近隣地域マスタにトンネル要求;

4.4.5 地域境界ルータの動作アルゴリズム

地域境界ルータがマルチキャストデータグラムを受け取った時の処理は，表 4.4のようなアルゴリズムで行なわれる．これは，地域外へのインタフェースを用いないことを除けば，通常のリバースパスツリーを構築して転送する RPB アルゴリズムと基本的に同じである．このアルゴリズムも，後述の広域のマルチキャスト通信媒体が故障等の理由で使用できない場合も包含している．

表 4.4: 地域境界ルータがマルチキャストデータグラムを受け取った時の処理

```
FOR (地域外へのインタフェース以外について繰り返し)
        IF 各インタフェースの子リンクフラグが立っている THEN
                そのインタフェースに複製を転送;
```

表 4.5に地域境界ルータの経路制御情報の交換の処理のアルゴリズムを示す.

表 4.5: 地域境界ルータの経路制御情報交換

```
SWITCH 経路制御情報の発信元
CASE 隣合っている地域境界ルータ
        近隣の地域マスタのアドレスを自分の地域マスタに転送;
        近隣の地域マスタからルート地域までのホップ数を自分の地域マスタに転送;
        ルート地域マスタのアドレスを自分の地域マスタに転送;
CASE 自分の属する地域の地域マスタ:
        隣合っている地域境界ルータに経路制御情報を転送;
                (地域マスタのアドレス);
                (ルート地域までのホップ数);
                (ルート地域マスタのアドレス);
CASE その他 (一般のマルチキャストルータ):
        IF リバースパス経路制御情報 THEN
                経路制御テーブルを更新;
        ELSE
                エラー;
```

4.5　広域の通信媒体が故障している場合のパケットの伝播

ここで考えられ故障には，送信装置の故障と，ある受信局の受信装置の故障の場合がある．現在のハードウェアでは送信装置の故障自体を，ソフトウェアで直接的に検出できない．したがって，送信装置の故障は，受信装置が独立してすべて故障した場合として考え対処する．

4.5.1　1 つの地域における広域通信媒体への接続の故障

ある地域において，広域マルチキャスト通信媒体に接続している受信装置が故障した場合，表 4.3に示したように，地域マスタは送信局からデフォルト経路が一定時間来ないことを検出して，アンテナ，チューナ等の受信装置が故障したものとみなす．さらに，地域境界ルータからの情報に基づき，自分の地域よりもルート地域に近い地域マスタに，衛星通信チャネルからのデータを複製して受信装置が故障した地域マスタに送ることを要求する．要求を受けた近隣の地域マスタは，送信局へ送る場合と同じように，トンネリングか IP を IP にカプセル化することにより，要求を行なった地域マスタに通信衛星チャネルからのデータグラムを送る．例えば，図 4.4において，地域 B の地域マスタの受信装置が故障した場合には，ルート地域に近い地域の地域マスタ，つまり地域 A の地域マスタとの間でトンネリングを行なうようにし，衛星経由のデータグラムを受けとる．受信装置の故障の回復は，送信局からのデフォルト経路の検出により検知し，その場合には，複製の停止を要求する．

4.5.2　複数の地域における広域通信媒体への接続の故障

近隣の地域マスタも故障した場合には，さらに上位，つまりさらに ルート地域に近い地域マスタに複製を要求する．送信装置が故障している場合には，最終的に，ルート地域マスタをルートとして地域マスタによるブロードキャストツリーを衛星を用いないネットワークとして構成されることになる．

受信装置が故障していて，トンネリングを要求している地域に対して，近隣のルート地域に近い方の地域に位置する地域マスタは，通信衛星から受信したデータグラムをトンネリングして，故障している地域に送る．この場合，受信したすべてのデータグラムを故障している地域に送ると，故障している地域を始点とするマルチキャストデータグラムを重複して送ってしまうことになるので，故障している近隣の地域に存在するネットワークのアドレスのリストを保持してフィルタリングする．例えば図 4.4において，地域 B の地域マスタの受信装置が故障していると，この 地域マスタは衛星からのデータグラムを地域 A の地域マスタから受けとる．この場合，地域 B が始点であるようなデータグラムが地域 A から送られてくると無駄であるので，地域 A の地域マスタは始点アドレスを検査してトンネリングしている先から来ている場合には転送を抑制する．

4.6 グループのフィルタリング

地域間をまたがる広域のマルチキャスト通信に関しては，そのトラフィックは必ず地域マスタを通過しなければならない．したがって，地域マスタがマルチキャストグループのリストを保持管理していれば，地域内に関係のないマルチキャストのトラフィックの流出を防ぐことができる．

DVMRP の RPM では，NMR(Non Membership Reprot) を送ることにより，無駄なトラフィックが不必要に部分木の先に到達することを防いでいる．この方法では，1 回めのパケットは TRPB で送信し，2 回め以降は NMR をリーフ側のマルチキャストルータからルートに近いマルチキャストルータへ順番に送っていき枝を苅るということを行なっている．

しかしながら，この方法の欠点として，NMR を用いて枝を苅った後に，苅った枝の先でマルチキャストグループに参加するホストがあった場合にいつまでたってもマルチキャストパケットが到達しなくなることを防ぐために，NMR に有効期限をもうけてタイムアウトさせなくてはいけないことがあげられる．これにより何回かに 1 回は TRPB によりパケットがリーフまで到達するが，無駄なトラフィックが増えてしまう．

また，ショーテストリバースパスを基本として経路制御を行なっているために，ソースネットワークごとに NMR を管理してフィルタリングを行なう必要があり，経路制御表が複雑なものとなる．

これらの欠点を解消するために，HDVMRP では，ホストがマルチキャストグループに参加した場合，隣接ルータがショーテストリバースパスをたどって地域マスタにメンバシップ情報を登録する．これにより地域マスタは，自分が管理する地域のグループメンバシップのすべてを把握することができ，効果的なパケットのフィルタリングをグループアドレスに基づいて行なうことができる．これは，ソースネットワークごとにショーテストリバースパスの形が変化する DVMRP に比べて，マルチキャストパケットの流れる経路がほぼ一定である HDVMRP が優位であるということである．

4.7 まとめ

本章では，広域のブロードキャスト型の通信媒体と，既存のインターネットの広域接続であるポイント間接続の両方を効率良く利用し，広域のマルチキャスト通信を効率良く提供するための経路制御アルゴリズムについて述べた．

まず，ユニキャストデータグラムの経路制御について述べ，従来の経路制御の枠組の中で，単方向性の通信媒体である衛星通信もうまく扱えることを示した．

次に，階層化 DVMRP(HDVMRP) という新たなマルチキャスト経路制御アルゴリズムを提案し，そのアルゴリズムと動作についてまとめた．

HDVMRP では，広域ネットワークをいくつかの「地域」に分割し，その間を「広域マルチキャストバックボーン」で接続する．この広域マルチキャストバックボーンは，衛星通信などを利用した，物理的にも広域接続できるマルチキャストバックボーンネット

ワークであったり，トンネリング技術による仮想的なバックボーン接続であることが可能である．また複数の通信媒体からなるネットワークから構成することもできる．この構成により，広域で交換される経路制御情報の量を減らし，また地域内で交換される経路情報も地域内のネットワークの数程度に抑えるが可能となった．

広域にまたがるマルチキャストトラフィックがある地域内に伝播する場合，すべて地域マスタを通過し，地域マスタをルートとするショーテストパスツリーに沿って伝わるように経路制御が行なわれるため，地域マスタにグループのリストを保持管理させることにより，マルチキャストグループに対する効果的なトラフィックのフィルタリングが可能となった．従来の経路制御方式では，グループごとにショーテストパスツリーの構造が変化し，すべてのマルチキャストルータがグループのフィルタリングに参加しなければ効果的なフィルタリングが行なうことが不可能であったが，本アルゴリズムでは，地域マスタだけが参加すればよいので，グループ情報の交換，各ルータの負荷という面でも優位性をもっている．

第 5 章

JP MBone

5.1 経緯と現状

現在Internetでは、マルチキャストデータグラムを用いた実験のために、Internet上に仮想的に構築されたネットワークであるMBoneと呼ばれるマルチキャストバックボーンの運用と実験が行なわれている。このMBoneは、当初1992年3月にSan Diegoで行なわれたIETFにおいてその会議の模様を実時間でInternet上に放送するため[19]に、音声データや画像データをIPマルチキャストの技術を用いて送信するための実験基盤として始められた。その後、現在にいたるまで継続的なIPマルチキャストの実験基盤としての仮想的なバックボーンネットワークとして実験運営されており、その後のIETF会議などの中継を重ねるごとに参加組織数、参加国数とも増加している。そのため、MBoneは同年7月よりメーリングリスト `mbone@isi.edu` を通して世界中で協調的に運営されており、ここではそのほか情報交換や新しいソフトウェアのリリース案内など活発なやりとりがある。現在、このメーリングリストは表5.1のように、各国・地域ごとに再配布されている。

表 5.1: MBone メーリングリストの各国・地域別再配布

mbone-jp@wide.ad.jp	Japan
mbone-eu@sics.se	Europe
mbone-na@isi.edu	North America
mbone-oz@internode.com.au	Australia
mbone-nz@waikato.ac.nz	New Zealand
mbone-sg@lincoln.technet.sg	Singapore
mbone-korea@mani.kaist.ac.kr	Korea

一方、日本においても、1992年7月にBostonで行なわれたIETF会議の中継の時よりMBoneへの実験参加が始まり、11月にWashington DCで行なわれた3度目のIETF会議の中継を機会に、WIDEバックボーンを用いて東京から九州までの国内のマルチキャストバックボーンが構築された。そして、翌年1月、このJP MBoneの運用管理のために

メーリングリスト mbone-jp@wide.ad.jp が作られ、現在、この参加者による JP MBone 運用グループによって以下のような活動が行なわれている。

- JP MBone の円滑な運用に必要な「ガイドライン等の作成」
- 適切な JP MBone の構成のために必要な「トンネル設定の調整」
- TTL と Threshold によるパケットの「伝播範囲の調整」
- JP MBone を利用した「マルチキャスト実験の相互調整」
- JP MBone での活動の「インターネットへの広報」

この JP MBone メーリングリストは、表 5.1のように MBone メーリングリストの再配布先にもなっており、特に国内においての情報交換や協調的な運用のために使われている。参加希望のメールの宛先は mbone-jp-request@wide.ad.jp となっている。

現在、JP MBone には 30 を越える組織が参加しており、図 5.1 のようにいくつかの JP MBone NOC を設けてそこに各組織を接続することによって、実際の IP ネットワークに沿った効率的な運営を計っている。また、海外へは WIDE の海外線を用いて藤沢 NOC と NASA を結んでいる。

IETF 会議の中継などと同様に、日本国内においてもいくつかの会議やイベントに対して JP MBone を用いて音声や映像などを中継する実験がいくつも行なわれてきており、JP MBone を通して会場外からの会議参加などが行なわれている。これらのリストを 表 5.2 に示す。

5.2　MBone への接続設定

マルチキャスト IP データグラムはクラス D アドレスを用いるが、クラス D のアドレスを経路制御できるゲートウェイは限られている。そこで、クラス D のアドレスを経路制御できないゲートウェイを飛び越してマルチキャスト IP データグラムを伝播させるための技術として RFC1075 に規定されるトンネリングが用いられる。また経路制御情報自体の交換で現在使われているものは同じく RFC1075 に規定されている DVMRP(Distance Vector Multicast Routing Protocol)[7] であり、mrouted として実装されている。RFC1075 に規定されるトンネリングは IP オプションの LSRR(loose source and record route) を用いて実装されているが、効率の点の問題のため現在の MBone においてはマルチキャスト IP データグラムをユニキャストの IP データグラムにカプセル化する方式によってトンネリングが行なわれている。

自分と同じ Ether 上に mrouted が動いている MBone へ接続されたホストがない場合、MBone に接続するためには上記のトンネリングの設定を行なう必要がある。これには、/etc/mrouted.conf という設定ファイルを記述して mrouted を動作させる。設定ファ

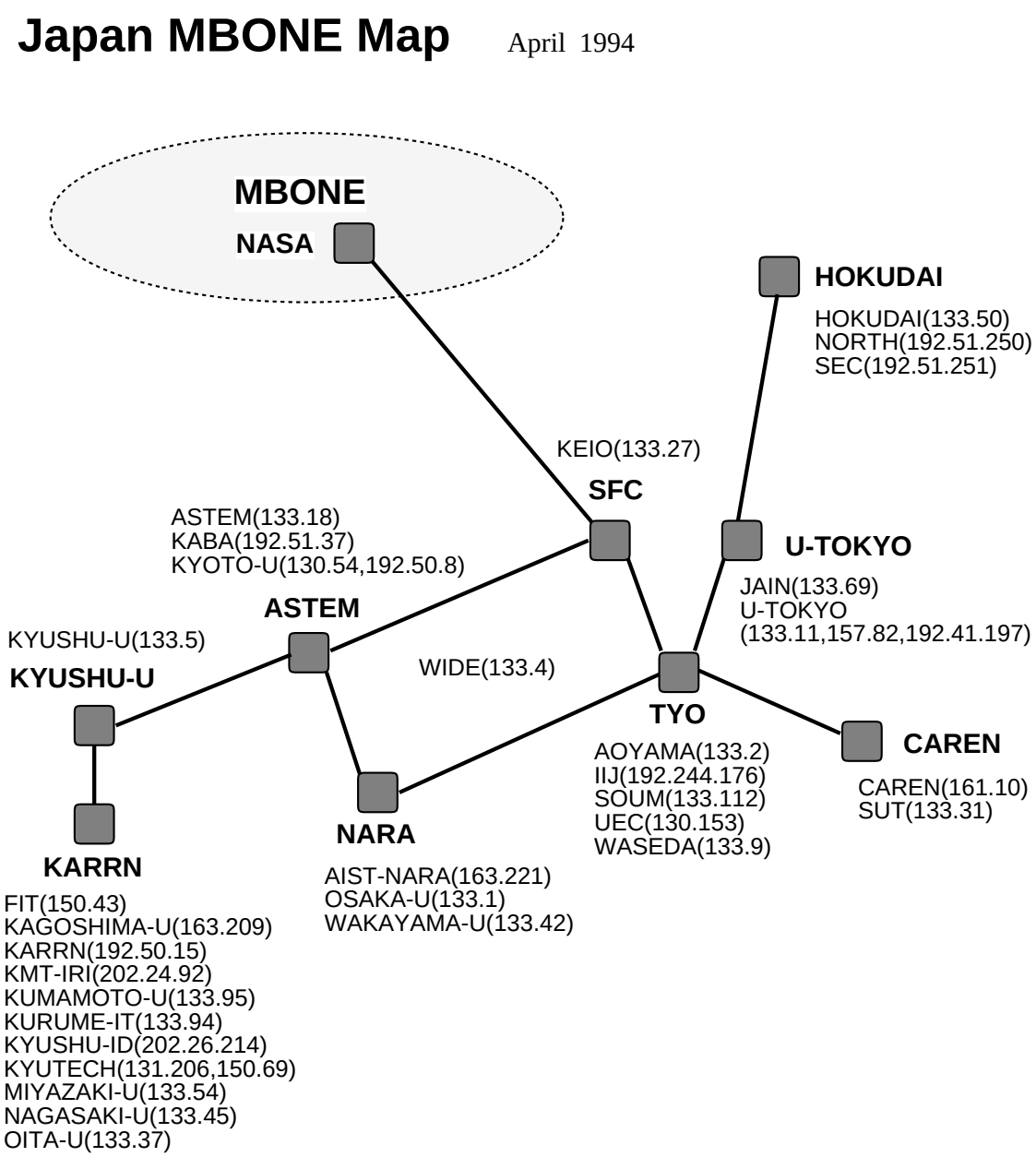

図 5.1: 国内の MBone 接続トポロジー図

```
phyint <local-addr> [disable] [metric <m>] [threshold <t>] [# Comment]
tunnel <local-addr> <remote-addr> [metric <m>] [threshold <t>] [# Comment]
```

図 5.2: /etc/mrouted.conf の設定

表 5.2: JP MBone 利用で行なわれた会議・イベント一覧

日付	内容
93 年 7 月 23 日	KARRN 協会設立総会
	九州大学大型計算機センター (vat nv)
93 年 8 月 6 日	ハイパーネットワーク社会研究所ワークショップ
	大分会場より公衆回線で大分大学 (vat)
93 年 9 月 30 日	和歌山 情報ネットワークセミナー
	加太国民休暇村研修センターより ISDN で和歌山大学 (vat)
93 年 10 月 18 日〜20 日	Jain Consortium Workshop
	熱海会場より ISDN で東京大学 (vat nv wb)
93 年 10 月 25 日	東京大学キャンパスネットワーク UTnet 完成披露式
	東京大学山上会館 (vat nv)
93 年 11 月 11 日	NORTH Symposium
	札幌市エレクトロニクスセンター (vat nv)
93 年 11 月 25 日	1993 年度情報処理学会九州支部若手の会
	熊本県工業技術センター (vat nv wb)
93 年 12 月 8 日	JP MBone meeting
	早稲田大学情報科学研究教育センター (vat nv)
93 年 12 月 9 日	IP meeting '93
	NEC 本社ビルより ISDN で東京大学 (vat)
94 年 1 月 25 日	医療とネットワーク講演会
	札幌医科大学 (vat)
94 年 1 月 28 日	JAIN Consortium Symposium '94
	工学院大学より ISDN で東京大学 (vat nv wb)
94 年 2 月 18 日〜19 日	札幌 インターネットセミナー
	札幌市エレクトロニクスセンター (vat nv)
94 年 3 月 8 日	映像作家 David Blair 氏訪問
	東京大学人工物研究センター (vat nv)
94 年 3 月 14 日	KARRN 協会講演会
	長崎厚生年金会館より ISDN で九州大学 (vat nv wb)
94 年 3 月 17 日	第 28 回高度情報システム研究会
	久留米リサーチセンターより ISDN で九州大学 (vat nv wb)
94 年 4 月 22 日	第 1 回京都大学高度情報化フォーラム
	京都大学医学部附属病院 (vat nv)

イルは各行が 1 つのトンネルまたは物理インタフェースに対応しており、図 5.2のように記述する。

tunnel で始まる行がトンネル設定のための記述である。local-addr は mrouted を動作させるマシンの IP アドレスであり、remote-addr はトンネルの反対側のマシンの IP アドレスである。ここで注意しなければいけないのは、local-addr に記述するインターフェースは、remote-addr にユニキャストの IP データグラムが送られるときのソースアドレスと一致していなければならないことである。metric はそのトンネルに対するメトリック値を記述できる。threshold はこのトンネルに対する通過のための閾値であり、マルチキャストデータグラムが伝播される場合に、IP データグラムのヘッダ中の TTL がこの値以下になるとこのトンネルを通っては伝わらない。

phyint で始まる行は、ある物理インタフェースに対して、特にメトリックや閾値を指定したい場合や、マルチキャストの経路制御を禁止したい場合に用いることができる。

```
tunnel 133.18.64.3 133.4.19.2 metric 1 threshold 64     # nakasu
tunnel 133.18.64.3 133.4.23.2 metric 1 threshold 32     # wnoc-nara-ss2
tunnel 133.18.64.3 133.4.27.27 metric 3 threshold 64    # fasthand
tunnel 133.18.64.3 192.50.8.1 metric 1 threshold 64     # handshake
tunnel 133.18.64.3 192.51.37.16 metric 1 threshold 96   # chotto
tunnel 133.18.212.19 133.18.96.4 metric 1 threshold 1   # wawona
```

図 5.3: /etc/mrouted.conf の記述例

図 5.3に国内で使われているトンネルの記述例を示す。このホストは 133.18.64.3 と 133.18.212.19 という二つの物理インターフェースを持ち、133.18.64.3 側では 5 つのトンネル、133.18.212.19 側では 1 つのトンネルを持っている。threshold や metric の設定については、後述の運用ガイドラインなどを参照して、適切に設定する必要がある。

```
% mrinfo localhost
127.0.0.1 (localhost) [version 2.0]:
  133.18.64.3 -> 133.18.64.2 (wnoc-kyo-ss2.wide.ad.jp) [1/1]
  133.18.212.19 -> 0.0.0.0 (?) [1/1/querier]
  133.18.64.3 -> 133.4.19.2 (nakasu.wide.ad.jp) [1/64/tunnel]
  133.18.64.3 -> 133.4.23.2 (wnoc-nara-ss2.wide.ad.jp) [1/32/tunnel]
  133.18.64.3 -> 133.4.27.27 (fasthand.sfc.wide.ad.jp) [3/64/tunnel]
  133.18.64.3 -> 192.50.8.1 (handshake.gw.kyoto-u.ac.jp) [1/64/tunnel]
  133.18.64.3 -> 192.51.37.16 (chotto.kaba.or.jp) [1/96/tunnel/down]
  133.18.212.19 -> 133.18.96.4 (wawona.rcac.astem.or.jp) [1/1/tunnel]
```

図 5.4: mrinfo の実行例

双方でのトンネリングの設定がうまくいって mrouted を再起動すると、マルチキャス

トの経路制御交換が行なわれるようになり、マルチキャストカーネルに対応した netstat を用いて netstat -M などで確認できる。また、トンネリングなどの設定の状況は mrinfo 用いて確認することができ、例えば図 5.3で設定されたホストにおいて自分の設定状況は図 5.4のように確認することができる

5.3　実験運用のガイドライン

5.3.1　MBone の運用に関する調整

現在の IP マルチキャストの技術は発展途上にあり、またその実験環境である MBone は、ユニキャストのインターネットと物理的な回線を共有しており、不用意なマルチキャストパケットの送出は、他のインターネット利用者に多大な影響を与えることがある。多くのマルチキャストアプリケーションは、音声や画像といった多量の情報を広い範囲に流すため、その影響はユニキャストのファイル転送などとは比較できない程である。また、UDP を用いた現在のブロードキャストに近いマルチキャストパケットの配送技術は、より慎重な運用と調整を必要とする。マルチキャスト技術やマルチメディアアプリケーションの円滑な発展のために、この実験に係わる者は、連携しお互いに調整しつつ活動を行ない、同じようにインターネットを利用する他の人達と協調して、MBone の運用を行なっていかなければならない。

日本においては IP マルチキャスト通信の実験、運用を円滑に進めるため、JP MBone 運用グループが構成されている。無秩序な MBone の利用は過大なトラフィックを生み出し、一般のユニキャストの通信に障害をもたらすことなどがあるので、各組織が協調して運用する必要がある。そのため、JP MBone に接続された組織の担当者 (最低 1 人以上) は、この運用グループ（メーリングリスト）に参加することが期待されており、その組織における JP MBone に接続されているすべてのホストについてのマルチキャスト的アクティビティに関して責任を持てる人が望ましい。

現在 JP MBone では IP マルチキャストのトラフィックに関する統計をきちんと取ることを計画している。JP MBone を利用した IP マルチキャスト通信の実験を行なう場合には、それに先だって (できるだけ前に)、メーリングリストに対して必ずアナウンスを行なう。これは、実験のスケジュールが重なった場合など、メーリングリスト上にて調整を行なう必要からである。

5.3.2　MBone の設定とパケットの送出について

一般に threshold の値はネットワークの管理者の指示の下で設定するものである。一方、初期 TTL はアプリケーションの利用者が設定するものである。これらネットワークの管理者と利用者の設定の間に同一の基準を設け、マルチキャストパケットが通過するリンクの適切な利用を図る。

本来 threshold は組織内外や region などの範囲を設けるために使われるべき概念であ

り、その範囲を指定するために 初期 TTL が使われる。しかし、現在のマルチキャストカーネルの実装では multicast といえども事実上の broadcast になっていること、また国内で MBone に参加しているところのリンクの回線速度にばらつきがありすぎることのため、当面の処置として、threshold の設定によって作られる region の概念を、そこで最大利用できるバンド幅によって決めることとし、それによって、細いリンクにその利用バンド幅を越えた大きな traffic をかけないようにするとともに、太いリンクの region 内では大きな traffic を使えるようにする。

- 原則として、そのリンクの利用バンド幅に応じて threshold を設定する。
- 例外として、組織内外間のリンクでは、threshold を 32 以上にする。
- 海外リンクについては threshold=128 として扱う (現状は事情により 160)。

表 5.3: 国内での利用バンド幅別 threshold 設定

使用される上限目安	→	threshold
制限なし	→	1 (自由)
384Kbps	→	32
192Kbps	→	48
96Kbps	→	64
64Kbps	→	80
32Kbps	→	96
16Kbps	→	112
海外	→	128

最大どれだけのバンド幅を MBone で使用されてよいかは、トンネルが使用する物理的なリンクの管理者、あるいはネットワークプロジェクトの担当者と相談の上決定し、それに基づいて threshold の値を設定する。例えば、「うちのリンクは 192Kbps だが、MBone に使用される上限目安としては 64Kbps くらいにして欲しい」という場合は、表 5.3より、threshold=80 とする。一般的には、物理的回線速度の半分を MBone での使用の上限目安として設定する場合が多い。

ユーザがアプリケーションなどで MBone を使う場合、表 5.3に沿って表 5.4のような目安で 初期 TTL を決定して動かすとよい。(ただし、使用する上限の目安は、ユーザが独占して使用する場合)

例えば、国内 conference のために vat と nv をそれぞれ 32Kbps くらいで動かしたい場合、合計使用量 64Kbps なので、両方とも 初期 TTL=95 で放送することができる。しかし、vat による音声放送だけをより優先して広い範囲に流したい場合、vat の 初期 TTL を 111 にし、nv の intial TTL を 95 にすることも上記の表より可能である。しかし、現実には自分一人で独占して利用するわけではないので、調整が必要となる。

表 5.4: 国内での利用バンド幅別初期 TTL 設定

使用される上限目安	→	初期 TTL
組織内	→	31 以下
384Kbps まで	→	47 以下
192Kbps まで	→	63 以下
96Kbps まで	→	79 以下
64Kbps まで	→	95 以下
32Kbps まで	→	111 以下
国内全体	→	127 以下

5.3.3　MBone 利用のための運用規定

JP MBone を利用したいひとは、コンタクト先 (名前, 電話, e-mail address)、内容、メディア、バンド幅、期間 (時間)、範囲 (初期 TTL)、および、sd を利用する場合は session 名、使用できない場合は address、port などの情報を、以下の例のように JP MBone メーリングリストへなるべく早めに通知することとし、必要な場合はメーリングリスト上にてそれぞれの調整することになる。

また、その結果、traffic 的、時間的に影響範囲が多いと判断されたものについては、適宜、ip-connection メーリングリストなどへのアナウンスを行なうことがある。将来は Gopher や WWW などを用いて Internet 上からそれらの情報が引けるようにすることも計画されている。

例:

To: mbone-jp@wide.ad.jp
Subject: 2nd JP MBone meeting

1. 内容: 第 2 回 JP MBone meeting の中継
2. session 名: 2nd JP MBone meeting
3. 使用メディア: nv, vat
4. 使用バンド幅: 76Kbps (nv 32Kbps, vat dvi 46Kbps)
5. 開始日時: 1994/3/10 19:00
6. 終了日時: 1994/3/10 20:30
7. 初期 TTL: 79
8. コンタクト先: 村井 純, 0466-49-XXXX
9. メールアドレス: junsec@wide.ad.jp

また、実験その他の理由により、長期間にわたって session を保持する必要がある場合があるが、その場合についても同様にメーリングリストへ通知を行なう。

初期 TTL が 128 （現在は 160）を越えて、海外へパケットが流れていく利用の際には、mbone@isi.edu へ通知を行なうなどして、国際的な調整を行なう。

5.3.4 常時開かれているセッション

JP MBone の運用および調整を円滑に進めるために、次の 3 つのセッションを常時開いている。

- JP MBone Audio
- JP MBone Video
- JP MBone Whiteboard

上記のセッションに関するアドレス、ポートおよび TTL 等の情報は sd を使用して常時アナウンスされている。これらのセッションは誰でも参加でき, JP MBone のメンバとの迅速な情報交換を可能にしている。また MBone 全体のために同様に次のセッションが常に開かれている。

- MBone Audio

5.4 ソフトウェア環境

ここでは、MBone へ参加するために用意すべき IP マルチキャストカーネル及び各アプリケーションの現状について、それぞれ概要を説明する。なお、これらのソフトウェアは国内においては、biscuit.mmws.astem.or.jp などの anonymous ftp から取ってくることができる。以下のそれぞれの説明において、例として biscuit.mmws.astem.or.jp: ftp/multicast のしたのディレクトリ (kernel, mrouted, vat, nv, wb, imm, sd など) からの相対パスで、参考になるファイルなどを挙げている。ただし、MBone の実験基盤としての性格のため、常にバグやバージョンアップなどの情報に注意することが望ましい。

5.4.1 カーネル

- SunOS 4.1.∗
  パッチを当てる事によって利用可能 (ipmulti-sunos41x.tar.Z)。また、枝狩りを行なうベータリリース版も利用可能 (ipmulti-3.1beta.tar.Z 及び ipmulti-3.1beta.patch1)。

- Solaris 2.∗
  2.2 はパッチを当てる事によって利用可能 (Solaris2.x/Kernel)。2.3 では標準で実装されている。

- BSD/386 1.0/1.1
  1.0 はパッチを当てる事によって利用可能 (bsd386-ipmcast.tar.Z)。1.1 では標準で実装されている。

- DEC Ultrix 4.* (DECStation5000)
  パッチを当てる事によって利用可能。ただし packetfilter 関係のカーネルのオブジェクト /sys/MIPS/BINARY/pfilt.o を入れ換える必要がある (dec-pfilt/pfilt.o.Z)。音声を利用するためには AudioFile を Kernel に組み込まなくてはならない。

- DEC OSF/1 1.*/2.0 (DEC alpha 3000)
  1.*はパッチを当てる事により利用可能 (ipmulti-decosf1.tar.Z)。2.0 では標準で実装されている。音声を利用するためには AudioFile を Kernel に組み込まなくてはならない。

- SGI IRIS 4.*/5.*
  4.*はパッチを当てる事により利用可能 (sgi/IRIX4)。5.1 以降では標準で実装されている。

- SONY NEWS-OS 4.0
  パッチを当てる事により利用可能 (ipmulticast+vlsm.newsos4.tar.Z)。

- HP-UX 9.01
  パッチを当てる事により利用可能 (hp-ipmulti.tar.Z)。

- AIX 3.2.5 (RISC SYSTEM/6000)
  パッチを当てる事により利用可能 (aix325-ipmcast.tar.Z)。

5.4.2　ツール類

- mrouted
  上記のマルチキャストカーネルに加え、マルチキャストルーターとして用いる時は mrouted を走らせる必要がある。現在最新のものは、動かすカーネルのバージョンにそれぞれ応じて mrouted2.2 もしくは mrouted3.2b である。

- mtest
  マルチキャストの簡単なテストとして、各物理インターフェースに対して任意の class D アドレスで join や leave を試すことができる。

- ping
  マルチキャスト対応のものがあり、通常の ping とは違って TTL の指定などができる。

- netstat
  マルチキャスト対応のものがあり、netstat -M によってマルチキャストルーティングテーブルの状態を見ることができる。

- mrinfo
  引数でマルチキャストルーターを指定し、その設定と状態を見ることができる。

- map-mbone
  mrinfo と同様の機能のほか、MBone 全体の様子を見ることもできる。

5.4.3 アプリケーション

- vat
  vat(Visual Audio Tool) は、X Window ベースの音声会議ツールであり、バイナリ形式で配布されている。現在の最新バージョンは 3.2。

- nv
  nv(Network Video) は、X Window ベースのビデオ会議ツールであり、ソースも公開されている。現在の最新バージョンは 3.2 であるが、機能強化された 3.3 が実験用として配布されている。

- wb
  wb(White Board) は、X Window ベースの共有ホワイトボードツールであり、バイナリ形式で配布されている。現在の最新バージョンは 1.57。

- imm
  imm(IMage Multicaster) は、JPEG 形式のイメージ情報をマルチキャストを用いて配布するツールであり、ソースも公開されている。現在の最新バージョンは 3.3。

  イメージを送り出すサーバ (immserv) と受け取るクライアント (imm) とから構成されている。

- sd
  sd(Session Directory) は、 X Window ベースのセッション管理ツールであり、バイナリ形式で配布されている。現在の最新バージョンは 1.14。

  通常、vat や nv といったマルチキャストアプリケーションの多くは、この sd のパネルを操作し希望するセッションを選択することによって、sd によって適切な引数が与えられて起動することができる。それらのセッションのアナウンスも sd によって行なう。